# Table of Contents

Welcome ........................................................................................................................................... 1

Resources .......................................................................................................................................... 2

Channels .................................................................................................................................. 2
  URI .................................................................................................................................. 2
  HTTP Methods ..................................................................................................................... 2
  Schemas ........................................................................................................................ 3
  Properties .................................................................................................................. 27

Channels channelId ................................................................. 126
  URI ............................................................................................................................... 126
  HTTP Methods ............................................................................................................... 126
  Schemas ........................................................................................................................ 127
  Properties .................................................................................................................. 138

Channels channelId Start .................................................. 235
  URI ............................................................................................................................... 235
  HTTP Methods ............................................................................................................... 235
  Schemas ........................................................................................................................ 236
  Properties .................................................................................................................. 247

Channels channelId Stop ................................................... 344
  URI ............................................................................................................................... 344
  HTTP Methods ............................................................................................................... 344
  Schemas ........................................................................................................................ 345
  Properties .................................................................................................................. 356

InputSecurityGroups .......................................................... 453
  URI .................................................................................................................................. 453
  HTTP Methods ............................................................................................................... 453
  Schemas ........................................................................................................................ 454
  Properties .................................................................................................................. 455

InputSecurityGroups inputSecurityGroupId ................. 457
  URI ............................................................................................................................... 457
  HTTP Methods ............................................................................................................... 457
  Schemas ........................................................................................................................ 459
  Properties .................................................................................................................. 460

Inputs .................................................................................. 461
  URI ............................................................................................................................... 461
  HTTP Methods ............................................................................................................... 461
  Schemas ........................................................................................................................ 462
  Properties .................................................................................................................. 464

Inputs inputId ................................................................. 469
  URI ............................................................................................................................... 469
  HTTP Methods ............................................................................................................... 469
  Schemas ........................................................................................................................ 471
  Properties .................................................................................................................. 472
Welcome

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

We assume that your IAM user credentials have the permissions needed to use the 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.
Resources

The AWS Elemental MediaLive REST API includes the following resources.

Topics
- Channels (p. 2)
- Channels channelId (p. 126)
- Channels channelId Start (p. 235)
- Channels channelId Stop (p. 344)
- InputSecurityGroups (p. 453)
- InputSecurityGroups inputSecurityGroupId (p. 457)
- Inputs (p. 461)
- Inputs inputId (p. 469)

Channels

URI
/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 (p. 14)</td>
<td>Array of channels</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 26)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 26)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>None</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 26)</td>
<td>You do not have permission to list channels.</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. 15)</td>
<td>Creation of channel is started.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 26)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>422</td>
<td>ChannelConfigurationValidationError (p. 26)</td>
<td>Channel failed validation and could not be created.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 26)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>None</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>None</td>
<td>Gateway Timeout Error</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 26)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 26)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 26)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
</tbody>
</table>

Schemas

Request Bodies

Example POST

```json
{
  "inputAttachments (p. 51)": [
    {
      "inputId (p. 89)": "string",
      "inputSettings (p. 90)": {
        "sourceEndBehavior (p. 93)": "enum",
        "audioSelectors (p. 93)": [
          {
            "name (p. 38)": "string",
            "selectorSettings (p. 38)": {
              "audioLanguageSelection (p. 38)": {
```
"languageSelectionPolicy (p. 35)" : enum,
  "languageCode (p. 35)" : "string",
  "audioPidSelection (p. 38)" : {
    "pid (p. 37)" : integer
  },
  "deblockFilter (p. 93)" : enum,
  "inputFilter (p. 93)" : enum,
  "networkInputSettings (p. 93)" : {
    "hlsInputSettings (p. 112)" : {
      "retries (p. 84)" : integer,
      "bandwidth (p. 85)" : integer,
      "retryInterval (p. 85)" : integer,
      "bufferSegments (p. 85)" : integer
    },
    "serverValidation (p. 112)" : enum
  },
  "videoSelector (p. 93)" : {
    "colorSpace (p. 124)" : enum,
    "selectorSettings (p. 124)" : {
      "videoSelectorPid (p. 125)" : {
        "pid (p. 125)" : integer
      },
      "videoSelectorProgramId (p. 126)" : {
        "programId (p. 125)" : integer
      }
    },
    "colorSpaceUsage (p. 124)" : enum
  },
  "filterStrength (p. 93)" : integer,
  "captionSelectors (p. 94)" : [
    {
      "name (p. 47)" : "string",
      "languageCode (p. 47)" : "string",
      "selectorSettings (p. 47)" : {
        "embeddedSourceSettings (p. 47)" : {
          "scte20Detection (p. 63)" : enum,
          "source608ChannelNumber (p. 64)" : integer,
          "convert608To708 (p. 64)" : enum,
          "source608TrackNumber (p. 64)" : integer
        },
        "scte20SourceSettings (p. 47)" : {
          "source608ChannelNumber (p. 116)" : integer,
          "convert608To708 (p. 116)" : integer
        },
        "dvbSubSourceSettings (p. 47)" : {
          "pid (p. 58)" : integer
        },
        "teletextSourceSettings (p. 47)" : {
          "pageNumber (p. 120)" : "string"
        },
        "aribSourceSettings (p. 48)" : {
        },
        "scte27SourceSettings (p. 48)" : {
          "pid (p. 116)" : integer
        }
      }
    }
  ],
  "denoiseFilter (p. 94)" : enum
}
AWS Elemental MediaLive API Reference
Schemas

"reserved (p. 51)" : "string",
"roleArn (p. 51)" : "string",
"requestId (p. 51)" : "string",
"destinations (p. 51)" : [  
  
  "settings (p. 113)" : [  
  
  "passwordParam (p. 113)" : "string",
  "url (p. 113)" : "string",
  "username (p. 113)" : "string"
  
  ],
  "id (p. 113)" : "string"
  
  ],

"name (p. 52)" : "string",
"encoderSettings (p. 52)" : {
  "timecodeConfig (p. 64)" : {
    "syncThreshold (p. 120)" : integer,
    "source (p. 120)" : enum
  },
  "outputGroups (p. 64)" : [  
  
  "outputs (p. 113)" : [  
  
  "videoDescriptionName (p. 112)" : "string",
  "outputName (p. 112)" : "string",
  "captionDescriptionNames (p. 112)" : [  
    "string"
  ],
  "outputSettings (p. 112)" : [  
  
  "archiveOutputSettings (p. 114)" : {  
  
  "extension (p. 32)" : "string",
  "containerSettings (p. 32)" : {  
  
  "m2tsSettings (p. 31)" : {  
  
  "audioStreamType (p. 98)" : enum,
  "ecmPid (p. 98)" : "string",
  "dvbTeletextPid (p. 98)" : "string",
  "aribCaptionsPidControl (p. 98)" : enum,
  "bitrate (p. 98)" : integer,
  "segmentationTime (p. 98)" : number,
  "rateMode (p. 98)" : enum,
  "audioPids (p. 99)" : "string",
  "ebpLookaheadMs (p. 99)" : integer,
  "ebpAudioInterval (p. 99)" : enum,
  "audioFramesPerPes (p. 99)" : integer,
  "scte35Pid (p. 99)" : "string",
  "programNum (p. 100)" : integer,
  "pcrPeriod (p. 100)" : integer,
  "segmentationStyle (p. 100)" : enum,
  "pmtInterval (p. 100)" : integer,
  "absentInputAudioBehavior (p. 101)" : enum,
  "audioBufferModel (p. 101)" : enum,
  "dvbNitSettings (p. 101)" : {  
  
  "networkName (p. 52)" : "string",
  "networkId (p. 52)" : integer,
  "repInterval (p. 52)" : integer
  },
  "absentInputAudioBehavior (p. 101)" : enum,
  "timedMetadataBehavior (p. 101)" : enum,
  "scte35Control (p. 101)" : enum,
  "bufferModel (p. 102)" : enum,
  "scte35Control (p. 102)" : enum,
"ebpPlacement (p. 102)": enum,
"arib (p. 102)": enum,
"dvbSdtSettings (p. 102)": {
  "serviceName (p. 53)": "string",
  "serviceProviderName (p. 53)": "string",
  "repInterval (p. 53)": integer,
  "outputSdt (p. 53)": enum
},
"nullPacketBitrate (p. 102)": number,
"pcrPid (p. 102)": "string",
"transportStreamId (p. 103)": integer,
"videoPid (p. 103)": "string",
"pcrControl (p. 103)": enum,
"esRateInPes (p. 103)": enum,
"segmentationMarkers (p. 103)": enum,
"klv (p. 103)": enum,
"dvbTdtSettings (p. 103)": {
  "repInterval (p. 58)": integer
},
"ccDescriptor (p. 104)": enum,
"patInterval (p. 104)": integer,
"etvPlatformPid (p. 104)": "string",
"dvbSubPids (p. 104)": "string",
"aribCaptionsPid (p. 104)": "string",
"scte27Pids (p. 104)": "string",
"klvDataPids (p. 105)": "string"
},
"nameModifier (p. 32)": "string"
},
"msSmoothOutputSettings (p. 114)": {
  "nameModifier (p. 111)": "string"
},
"udpOutputSettings (p. 115)": {
  "destination (p. 122)": {
    "destinationRefId (p. 114)": "string"
  },
  "bufferMsec (p. 122)": integer,
  "containerSettings (p. 122)": {
    "m2tsSettings (p. 121)": {
      "audioStreamType (p. 98)": enum,
      "ecmPid (p. 98)": "string",
      "dvbTeletextPId (p. 98)": "string",
      "aribCaptionsPidControl (p. 98)": enum,
      "bitrate (p. 98)": integer,
      "segmentationTime (p. 98)": number,
      "rateMode (p. 98)": enum,
      "audioPids (p. 99)": "string",
      "ebpLookaheadMs (p. 99)": integer,
      "ebpAudioInterval (p. 99)": enum,
      "audioFramesPerFes (p. 99)": integer,
      "fragmentTime (p. 99)": number,
      "scte35PId (p. 99)": "string",
      "programNum (p. 100)": integer,
      "pcrPeriod (p. 100)": integer,
      "pmtInterval (p. 100)": integer,
      "segmentationStyle (p. 100)": enum,
      "ebif (p. 100)": enum,
      "audioBufferModel (p. 101)": enum,
      "dvbNitSettings (p. 101)": {
        "networkName (p. 52)": "string",
        "networkId (p. 52)": integer,
        "repInterval (p. 52)": integer
      },
      "absentInputAudioBehavior (p. 101)": enum,
      "timedMetadataBehavior (p. 101)": enum,
"timedMetadataPid (p. 101)" : "string",
"pmtPid (p. 101)" : "string",
"etvSignalPid (p. 101)" : "string",
"bufferModel (p. 102)" : enum,
"scte35Control (p. 102)" : enum,
"ebpPlacement (p. 102)" : enum,
"arib (p. 102)" : enum,
"dvbSdtSettings (p. 102)" : {
  "serviceName (p. 53)" : "string",
  "serviceProviderName (p. 53)" : "string",
  "repInterval (p. 53)" : integer,
  "outputSdt (p. 53)" : enum
},

"nullPacketBitrate (p. 102)" : number,
"pcrPid (p. 102)" : "string",
"transportStreamId (p. 103)" : integer,
"videoPid (p. 103)" : "string",
"pcrControl (p. 103)" : enum,
"esRateInPes (p. 103)" : enum,
"segmentationMarkers (p. 103)" : enum,
"klv (p. 103)" : enum,
"dvbTdtSettings (p. 103)" : {
  "repInterval (p. 58)" : integer
},
"ccDescriptor (p. 104)" : enum,
"patInterval (p. 104)" : integer,
"etvPlatformPid (p. 104)" : "string",
"dvbSubPids (p. 104)" : "string",
"aribCaptionsPid (p. 104)" : "string",
"scte27Pids (p. 104)" : "string",
"klvDataPids (p. 105)" : "string"
}
"
"fecOutputSettings (p. 122)" : {
  "rowLength (p. 65)" : integer,
  "columnDepth (p. 66)" : integer,
  "includeFec (p. 66)" : enum
}
"
"hlsOutputSettings (p. 115)" : {
  "segmentModifier (p. 87)" : "string",
  "hlsSettings (p. 87)" : {
    "standardHlsSettings (p. 88)" : {
      "m3u8Settings (p. 119)" : {
        "pmtPid (p. 105)" : "string",
        "ecmPid (p. 105)" : "string",
        "scte35Behavior (p. 105)" : enum,
        "pcrPid (p. 106)" : "string",
        "audioPids (p. 106)" : "string",
        "audioFramesPerPes (p. 106)" : integer,
        "scte35Pid (p. 106)" : "string",
        "transportStreamId (p. 106)" : integer,
        "videoPid (p. 106)" : "string",
        "pcrControl (p. 107)" : enum,
        "pcrPeriod (p. 107)" : integer,
        "programNum (p. 107)" : integer,
        "patInterval (p. 107)" : integer,
        "timeMetadataBehavior (p. 107)" : enum
      },
      "audioRenditionSets (p. 119)" : "string"
    },
    "audioOnlyHlsSettings (p. 88)" : {
      "audioTrackType (p. 36)" : enum,
      "audioGroupId (p. 37)" : "string",
      "audioOnlyImage (p. 37)" : "string",
      "audioOnlyImage (p. 37)" : "string"
"passwordParam (p. 91)": "string",
"uri (p. 91)": "string",
"username (p. 91)": "string"
},

"nameModifier (p. 87)": "string"
},

"audioDescriptionNames (p. 112)": [
  "string"
]
],

"outputGroupSettings (p. 113)": {
  "archiveGroupSettings (p. 114)": {
    "destination (p. 31)": {
      "destinationRefId (p. 114)": "string"
    },
    "rolloverInterval (p. 32)": integer
  },
  "udpGroupSettings (p. 114)": {
    "inputLossAction (p. 121)": enum,
    "timedMetadataId3Frame (p. 121)": enum,
    "timedMetadataId3Period (p. 121)": integer
  },
  "msSmoothGroupSettings (p. 114)": {
    "eventId (p. 108)": "string",
    "fragmentLength (p. 108)": integer,
    "timestampOffset (p. 109)": "string",
    "segmentationMode (p. 109)": enum,
    "numRetries (p. 109)": integer,
    "acquisitionPointId (p. 109)": "string",
    "eventStopBehavior (p. 109)": enum,
    "sparseTrackType (p. 109)": enum,
    "destination (p. 109)": {
      "destinationRefId (p. 114)": "string"
    },
    "timestampOffsetMode (p. 110)": enum,
    "audioOnlyTimecodeControl (p. 110)": enum,
    "connectionRetryInterval (p. 110)": integer,
    "filecacheDuration (p. 110)": integer,
    "certificateMode (p. 110)": enum,
    "inputLossAction (p. 110)": enum,
    "sendDelayMs (p. 110)": integer,
    "eventIdMode (p. 111)": enum,
    "restartDelay (p. 111)": integer,
    "streamManifestBehavior (p. 111)": enum
  },
  "hlsGroupSettings (p. 114)": {
    "segmentsPerSubdirectory (p. 79)": integer,
    "ivInManifest (p. 79)": enum,
    "outputSelection (p. 80)": enum,
    "encryptionType (p. 80)": enum,
    "destination (p. 80)": {
      "destinationRefId (p. 114)": "string"
    },
    "indexNSegments (p. 80)": integer,
    "timedMetadataId3Frame (p. 80)": enum,
    "constantIv (p. 80)": "string",
    "baseUrlManifest (p. 80)": "string",
    "captionLanguageSetting (p. 81)": enum,
    "minSegmentLength (p. 81)": integer,
    "mode (p. 81)": enum,
    "keyProviderSettings (p. 81)": {
      "staticKeySettings (p. 94)": {
        "staticKeyId (p. 94)": "string",
        "provisioningType (p. 94)": enum,
        "keyProviderRefId (p. 94)": "string"
      }
    }
  }
}
"staticKeyValue (p. 119)": "string",
"keyProviderServer (p. 120)": {
  "passwordParam (p. 91)": "string",
  "uri (p. 91)": "string",
  "username (p. 91)": "string"
}
},
"manifestCompression (p. 81)": enum,
"ivSource (p. 81)": enum,
"tsFileMode (p. 82)": enum,
"manifestDurationFormat (p. 82)": enum,
"keyFormatVersions (p. 82)": "string",
"streamInfResolution (p. 82)": enum,
"manifestDurationFormat (p. 82)": enum,
"timestampDeltaMilliseconds (p. 82)": integer,
"baseUrlContent (p. 82)": "string",
"segmentationMode (p. 82)": enum,
"captionLanguageMappings (p. 83)": [
  {
    "languageDescription (p. 46)": "string",
    "captionChannel (p. 46)": integer,
    "languageCode (p. 47)": "string"
  }
],
"clientCache (p. 83)": enum,
"codecSpecification (p. 83)": enum,
"keepSegments (p. 83)": integer,
"timedMetadataId3Period (p. 83)": integer,
"programDateTime (p. 83)": enum,
"directoryStructure (p. 83)": enum,
"keyFormat (p. 84)": "string",
"inputLossAction (p. 84)": enum,
"adMarkers (p. 84)": [
  enum
],
"programDateTimePeriod (p. 84)": integer,
"segmentLength (p. 84)": integer,
"hlsCdnSettings (p. 84)": {
  "hlsAkamaSettings (p. 78)": {
    "httpTransferMode (p. 76)": enum,
    "salt (p. 76)": "string",
    "numRetries (p. 76)": integer,
    "restartDelay (p. 77)": integer,
    "connectionRetryInterval (p. 77)": integer,
    "filecacheDuration (p. 77)": integer,
    "token (p. 77)": "string"
  },
  "hlsWebdavSettings (p. 78)": {
    "httpTransferMode (p. 89)": enum,
    "numRetries (p. 89)": integer,
    "restartDelay (p. 89)": integer,
    "connectionRetryInterval (p. 89)": integer,
    "filecacheDuration (p. 89)": integer
  },
  "hlsBasicPutSettings (p. 78)": {
    "numRetries (p. 77)": integer,
    "restartDelay (p. 77)": integer,
    "connectionRetryInterval (p. 78)": integer,
    "filecacheDuration (p. 78)": integer
  },
  "hlsMediaStoreSettings (p. 79)": {
    "mediaStoreStorageClass (p. 86)": enum,
    "numRetries (p. 86)": integer,
    "restartDelay (p. 86)": integer,
    "connectionRetryInterval (p. 86)": integer,
    "filecacheDuration (p. 86)": integer
  }
},
"hlsMediaStoreSettings (p. 79)": {
"mediaStoreStorageClass (p. 86)": enum,
"numRetries (p. 86)": integer,
"restartDelay (p. 86)": integer,
"connectionRetryInterval (p. 86)": integer,
"filecacheDuration (p. 86)": integer
}
audioDescriptions (p. 64)

- languageCodeControl (p. 33)
- audioTypeControl (p. 33)
- remixSettings (p. 34)
  - channelMappings (p. 115)
    - outputChannel (p. 32)
    - inputChannelLevels (p. 32)
      - inputChannel (p. 90)
      - gain (p. 90)
  - channelsOut (p. 115)
  - channelsIn (p. 115)
- audioType (p. 34)
- name (p. 34)
- codecSettings (p. 34)
  - ac3Settings (p. 33)
    - drcProfile (p. 30)
    - dialnorm (p. 30)
    - codingMode (p. 30)
    - metadataControl (p. 30)
    - bitrate (p. 31)
    - lfeFilter (p. 31)
    - bitstreamMode (p. 31)
  - aacSettings (p. 33)
    - vbrQuality (p. 27)
    - codingMode (p. 28)
    - profile (p. 28)
    - inputType (p. 28)
    - bitrate (p. 28)
    - rawFormat (p. 28)
    - rateControlMode (p. 28)
    - sampleRate (p. 28)
    - spec (p. 29)
  - eac3Settings (p. 33)
    - dialnorm (p. 60)
    - passthroughControl (p. 60)
    - drcLine (p. 60)
    - metadataControl (p. 60)
    - bitrate (p. 60)
    - ltrtSurroundMixLevel (p. 61)
    - surroundExMode (p. 61)
    - lfeControl (p. 61)
    - codingMode (p. 61)
    - surroundMode (p. 61)
    - attenuationControl (p. 61)
    - lfeFilter (p. 61)
    - dceFilter (p. 61)
    - phaseControl (p. 62)
    - ltrtCenterMixLevel (p. 62)
    - stereoDownmix (p. 62)
"bitstreamMode (p. 62)": enum,
"loRoSurroundMixLevel (p. 62)": number,
"drcRf (p. 62)": enum,
"loRoCenterMixLevel (p. 62)": number
},
"passThroughSettings (p. 33)": {
},
"mp2Settings (p. 33)": {
"codingMode (p. 108)": enum,
"bitrate (p. 108)": number,
"sampleRate (p. 108)": number
},
"languageCode (p. 34)": "string",
"streamName (p. 34)": "string",
"audioNormalizationSettings (p. 34)": {
"targetLkfs (p. 36)": number,
"algorithmControl (p. 36)": enum,
"algorithm (p. 36)": enum
},
"audioSelectorName (p. 35)": "string"
}
],
"captionDescriptions (p. 64)": [
{
"captionSelectorName (p. 44)": "string",
"languageDescription (p. 45)": "string",
"name (p. 45)": "string",
"DestinationSettings (p. 45)": {
"scte27DestinationSettings (p. 45)": {
},
"burnInDestinationSettings (p. 45)": {
"xPosition (p. 41)": integer,
"backgroundColor (p. 41)": enum,
"yPosition (p. 41)": integer,
"teletextGridControl (p. 41)": enum,
"backgroundOpacity (p. 41)": integer,
"fontOpacity (p. 42)": integer,
"shadowOpacity (p. 42)": integer,
"fontResolution (p. 42)": integer,
"shadowYOffset (p. 42)": integer,
"outlineSize (p. 42)": integer,
"outlineColor (p. 42)": enum,
"fontSize (p. 43)": "string",
"shadowXOffset (p. 43)": integer,
"alignment (p. 43)": enum,
"shadowColor (p. 43)": enum,
"fontColor (p. 43)": enum,
"font (p. 43)": {
"passwordParam (p. 91)": "string",
"uri (p. 91)": "string",
"username (p. 91)": "string"

},
"teletextDestinationSettings (p. 45)": {
},
"webvttDestinationSettings (p. 45)": {
},
"ttMLDestinationSettings (p. 46)": {
"styleControl (p. 121)": enum
},
"smpteTtDestinationSettings (p. 46)": {
},
"embeddedPlusScte20DestinationSettings (p. 46)": {
},
"dvbSubDestinationSettings (p. 46)": {
}
"xPosition (p. 54)": integer,
"backgroundColor (p. 54)": enum,
"yPosition (p. 55)": integer,
"teletextGridControl (p. 55)": enum,
"backgroundOpacity (p. 55)": integer,
"fontOpacity (p. 55)": integer,
"shadowOpacity (p. 55)": integer,
"fontResolution (p. 56)": integer,
"shadowYOffset (p. 56)": integer,
"outlineSize (p. 56)": integer,
"outlineColor (p. 56)": enum,
"fontSize (p. 56)": "string",
"shadowXOffset (p. 56)": integer,
"alignment (p. 57)": enum,
"shadowColor (p. 57)": enum,
"fontColor (p. 57)": enum,
"font (p. 57)": {
  "passwordParam (p. 91)": "string",
  "uri (p. 91)": "string",
  "username (p. 91)": "string"
},
"embeddedDestinationSettings (p. 46)": {
},
"aribDestinationSettings (p. 46)": {
},
"scte20PlusEmbeddedDestinationSettings (p. 46)": {
},
"languageCode (p. 45)": "string"}]
],
"availConfiguration (p. 65)": {
  "availSettings (p. 39)": {
    "scte35TimeSignalApos (p. 39)": {
      "adAvailOffset (p. 117)": integer,
      "webDeliveryAllowedFlag (p. 118)": enum,
      "noRegionalBlackoutFlag (p. 118)": enum
    },
    "scte35SpliceInsert (p. 39)": {
      "adAvailOffset (p. 117)": integer,
      "webDeliveryAllowedFlag (p. 117)": enum,
      "noRegionalBlackoutFlag (p. 117)": enum
    }
  }
},
"globalConfiguration (p. 65)": {
  "inputLossBehavior (p. 66)": {
    "inputLossImageType (p. 92)": enum,
    "inputLossImageColor (p. 92)": "string",
    "inputLossImageSlate (p. 92)": {
      "passwordParam (p. 91)": "string",
      "uri (p. 91)": "string",
      "username (p. 91)": "string"
    },
    "repeatFrameMsec (p. 92)": integer,
    "blackFrameMsec (p. 92)": integer
  },
  "supportLowFramerateInputs (p. 66)": enum,
  "initialAudioGain (p. 67)": integer,
  "inputEndAction (p. 67)": enum,
  "outputTimingSource (p. 67)": enum
},
"videoDescriptions (p. 65)": [
  {
    "respondToAfd (p. 123)": enum,
"scalingBehavior (p. 123)": enum,
"name (p. 123)": "string",
"width (p. 123)": integer,
"sharpness (p. 123)": integer,
"codecSettings (p. 123)": {
  "h264Settings (p. 123)": {
    "minIInterval (p. 70)": integer,
    "slices (p. 70)": integer,
    "parNumerator (p. 70)": integer,
    "gopSizeUnits (p. 71)": enum,
    "maxBitrate (p. 71)": integer,
    "bitrate (p. 71)": integer,
    "bufFillPct (p. 71)": integer,
    "temporalAq (p. 71)": enum,
    "afdsignaling (p. 71)": enum,
    "timecodeInsertion (p. 72)": enum,
    "bufSize (p. 72)": integer,
    "softness (p. 72)": integer,
    "framerateControl (p. 72)": enum,
    "fixedAfds (p. 72)": enum,
    "level (p. 72)": enum,
    "lookAheadRateControl (p. 72)": enum,
    "profile (p. 73)": enum,
    "framerateNumerator (p. 73)": integer,
    "framerateDenominator (p. 73)": integer,
    "gopClosedCadence (p. 73)": integer,
    "framerateControl (p. 73)": enum,
    "parControl (p. 73)": enum,
    "parDenominator (p. 73)": integer,
    "sceneChangeDetect (p. 73)": enum,
    "scanType (p. 73)": enum,
    "gopNumBFrames (p. 73)": integer,
    "flickerAq (p. 73)": enum,
    "rateControlMode (p. 73)": enum
  }
},
"height (p. 124)": integer
},
"availBlanking (p. 65)": {
  "state (p. 38)": enum,
  "availBlankingImage (p. 38)": {
    "passwordParam (p. 91)": "string",
    "uri (p. 91)": "string",
    "username (p. 91)": "string"
  }
},
"blackoutSlate (p. 65)": {
  "networkEndBlackoutImage (p. 39)": {
    "passwordParam (p. 91)": "string",
    "uri (p. 91)": "string",
    "username (p. 91)": "string"
  },
  "networkEndBlackout (p. 39)": enum,
  "networkId (p. 39)": "string",
  "blackoutSlateImage (p. 40)": {
    "passwordParam (p. 91)": "string",
    "uri (p. 91)": "string",
    "username (p. 91)": "string"
Response Bodies

Example ListChannelsResultModel

```json
{
    "channels (p. 95)": [
        {
            "inputAttachments (p. 50)": [
                {
                    "inputId (p. 89)": "string",
                    "inputSettings (p. 90)": {
                        "sourceEndBehavior (p. 93)": enum,
                        "audioSelectors (p. 93)": [
                            {
                                "name (p. 38)": "string",
                                "selectorSettings (p. 38)": {
                                    "audioLanguageSelection (p. 38)": {
                                        "languageSelectionPolicy (p. 35)": enum,
                                        "languageCode (p. 35)": "string"
                                    },
                                    "audioPidSelection (p. 38)": {
                                        "pid (p. 37)": integer
                                    }
                                }
                            }
                        ]
                    }
                }
            ],
            "deblockFilter (p. 93)": enum,
            "inputFilter (p. 93)": enum,
            "networkInputSettings (p. 93)": {
                "hlsInputSettings (p. 112)": {
                    "retries (p. 84)": integer,
                    "bandwidth (p. 85)": integer,
                    "retryInterval (p. 85)": integer,
                    "bufferSegments (p. 85)": integer
                },
                "serverValidation (p. 112)": enum
            },
            "videoSelector (p. 93)": {
                "colorSpace (p. 124)": enum,
                "selectorSettings (p. 124)": {
                    "videoSelectorPid (p. 125)": {
                        "pid (p. 125)": integer
                    },
                    "videoSelectorProgramId (p. 126)": {
                        "programId (p. 125)": integer
                    }
                }
            },
            "colorSpaceUsage (p. 124)": enum
        }
    ],
    "filterStrength (p. 93)": integer,
    "captionSelectors (p. 94)": [
        {
            "name (p. 47)": "string",
            "languageCode (p. 47)": "string",
            "selectorSettings (p. 47)": {
                "embeddedSourceSettings (p. 47)": {
                    "scte20Detection (p. 63)": enum,
                    "source608ChannelNumber (p. 64)": integer
                }
            }
        }
    ]
}
```
Example CreateChannelResultModel

```json
{
  "channel (p. 52)": {
    "inputAttachments (p. 48)": [
      {
        "inputId (p. 89)": "string",
        "inputSettings (p. 90)": {
          "sourceEndBehavior (p. 93)": "string",
          "audioSelectors (p. 93)": [
            {
              "source608To708 (p. 64)": enum,
              "source608TrackNumber (p. 64)": integer
            },
            "scte20SourceSettings (p. 47)": {
              "source608ChannelNumber (p. 116)": integer,
              "convert608To708 (p. 116)": enum
            },
            "dvbSubSourceSettings (p. 47)": {
              "pid (p. 58)": integer
            },
            "teletextSourceSettings (p. 47)": {
              "pageNumber (p. 120)": "string"
            },
            "aribSourceSettings (p. 48)": {},
            "scte27SourceSettings (p. 48)": {
              "pid (p. 116)": integer
            }
          }
        }
      }
    ],
    "roleArn (p. 50)": "string",
    "destinations (p. 50)": [
      {
        "settings (p. 113)": [
          {
            "passwordParam (p. 113)": "string",
            "url (p. 113)": "string",
            "username (p. 113)": "string"
          }
        ],
        "id (p. 113)": "string"
      }
    ],
    "name (p. 50)": "string",
    "id (p. 50)": "string",
    "pipelinesRunningCount (p. 50)": integer,
    "state (p. 51)": "string",
    "arn (p. 51)": "string",
    "egressEndpoints (p. 51)": [
      {
        "sourceIp (p. 49)": "string"
      }
    ],
    "nextToken (p. 95)": "string"
  }
}
```
"name (p. 38)": "string",
"selectorSettings (p. 38)": {
  "audioLanguageSelection (p. 38)": {
    "languageSelectionPolicy (p. 35)": enum,
    "languageCode (p. 35)": "string"
  },
  "audioPidSelection (p. 38)": {
    "pid (p. 37)": integer
  }
},
"deblockFilter (p. 93)": enum,
"inputFilter (p. 93)": enum,
"networkInputSettings (p. 93)": {
  "hlsInputSettings (p. 112)": {
    "retries (p. 84)": integer,
    "bandwidth (p. 85)": integer,
    "retryInterval (p. 85)": integer,
    "bufferSegments (p. 85)": integer
  },
  "serverValidation (p. 112)": enum
},
"videoSelector (p. 93)": {
  "colorSpace (p. 124)": enum,
  "selectorSettings (p. 124)": {
    "videoSelectorPid (p. 125)": {
      "pid (p. 125)": integer
    },
    "videoSelectorProgramId (p. 126)": {
      "programId (p. 125)": integer
    }
  },
  "colorSpaceUsage (p. 124)": enum
},
"filterStrength (p. 93)": integer,
"captionSelectors (p. 94)": [
  {
    "name (p. 47)": "string",
    "languageCode (p. 47)": "string",
    "selectorSettings (p. 47)": {
      "embeddedSourceSettings (p. 47)": {
        "scte20Detection (p. 63)": enum,
        "source608ChannelNumber (p. 64)": integer,
        "convert608To708 (p. 64)": enum,
        "source608TrackNumber (p. 64)": integer
      },
      "scte20SourceSettings (p. 47)": {
        "source608ChannelNumber (p. 116)": integer,
        "convert608To708 (p. 116)": integer
      },
      "dvbSubSourceSettings (p. 47)": {
        "pid (p. 58)": integer
      },
      "teletextSourceSettings (p. 47)": {
        "pageNumber (p. 120)": "string"
      },
      "aribSourceSettings (p. 48)": {
      },
      "scte27SourceSettings (p. 48)": {
        "pid (p. 116)": integer
      }
    }
  }
]
"denoiseFilter (p. 94)": enum
`roleArn (p. 48)`: "string",
`destinations (p. 48)`: [
  {
    "settings (p. 113)": {
      "passwordParam (p. 113)": "string",
      "url (p. 113)": "string",
      "username (p. 113)": "string"
    },
    "id (p. 113)": "string"
  },
  "name (p. 48)": "string",
  "encoderSettings (p. 48)": {
    "timecodeConfig (p. 64)": {
      "syncThreshold (p. 120)": integer,
      "source (p. 120)": enum
    },
    "outputGroups (p. 64)": [
      {
        "outputs (p. 113)": {
          "videoDescriptionName (p. 112)": "string",
          "outputName (p. 112)": "string",
          "captionDescriptionNames (p. 112)": [
            "string"
          ],
          "outputSettings (p. 112)": {
            "archiveOutputSettings (p. 114)": {
              "extension (p. 32)": "string",
              "containerSettings (p. 32)": {
                "m2tsSettings (p. 31)": {
                  "audioStreamType (p. 98)": enum,
                  "ecmPid (p. 98)": "string",
                  "dvbTeletextPid (p. 98)": "string",
                  "aribCaptionsPidControl (p. 98)": enum,
                  "bitrate (p. 98)": integer,
                  "segmentationTime (p. 98)": number,
                  "rateMode (p. 98)": enum,
                  "audioPids (p. 99)": "string",
                  "ebpLookaheadMs (p. 99)": integer,
                  "ebpAudioInterval (p. 99)": enum,
                  "audioFramesPerPes (p. 99)": integer,
                  "fragmentTime (p. 99)": number,
                  "scte35Pid (p. 99)": "string",
                  "programNum (p. 100)": integer,
                  "pcrPeriod (p. 100)": integer,
                  "pmtInterval (p. 100)": integer,
                  "segmentationStyle (p. 100)": enum,
                  "ebif (p. 100)": enum,
                  "audioBufferModel (p. 101)": enum,
                  "dvbNitSettings (p. 101)": {
                    "networkName (p. 52)": "string",
                    "networkId (p. 52)": integer,
                    "repInterval (p. 52)": integer
                  }
                },
                "absentInputAudioBehavior (p. 101)": enum,
                "timedMetadataBehavior (p. 101)": enum,
                "timedMetadataPid (p. 101)": "string",
                "pmtPid (p. 101)": "string",
                "etvSignalPid (p. 101)": "string",
                "bufferModel (p. 102)": enum,
"scte35Control (p. 102)": enum,
"ebpPlacement (p. 102)": enum,
"arib (p. 102)": enum,
"dvbSdtSettings (p. 102)": {
  "serviceName (p. 53)": "string",
  "serviceProviderName (p. 53)": "string",
  "repInterval (p. 53)": integer,
  "outputSdt (p. 53)": enum
},
"nullPacketBitrate (p. 102)": number,
"pcrPid (p. 102)": "string",
"transportStreamId (p. 103)": integer,
"videoPid (p. 103)": "string",
"pcrControl (p. 103)": enum,
"esRateInPES (p. 103)": enum,
"segmentationMarkers (p. 103)": enum,
"klv (p. 103)": enum,
"dvbTdtSettings (p. 103)": {
  "repInterval (p. 58)": integer
},
"ccDescriptor (p. 104)": enum,
"patInterval (p. 104)": integer,
"etvPlatformPid (p. 104)": "string",
"dvbSubPids (p. 104)": "string",
"aribCaptionsPid (p. 104)": "string",
"scte27Pids (p. 104)": "string",
"klvDataPids (p. 105)": "string"
},
"nameModifier (p. 32)": "string"
},
"msSmoothOutputSettings (p. 114)": {
  "nameModifier (p. 111)": "string"
},
"udpOutputSettings (p. 115)": {
  "destination (p. 122)": {
    "destinationRefId (p. 114)": "string"
  },
  "bufferMsec (p. 122)": integer,
  "containerSettings (p. 122)": {
    "m2tsSettings (p. 121)": {
      "audioStreamType (p. 98)": enum,
      "ecmPid (p. 98)": "string",
      "dvbTeletextPid (p. 98)": "string",
      "aribCaptionsPidControl (p. 98)": enum,
      "bitrate (p. 98)": integer,
      "segmentationTime (p. 98)": number,
      "rateMode (p. 98)": enum,
      "audioPids (p. 99)": "string",
      "ebpLookaheadMs (p. 99)": integer,
      "ebpAudioInterval (p. 99)": enum,
      "audioFramesPerPES (p. 99)": integer,
      "fragmentTime (p. 99)": number,
      "scte35Pid (p. 99)": "string",
      "programNum (p. 100)": integer,
      "pcrPeriod (p. 100)": integer,
      "pmtInterval (p. 100)": integer,
      "segmentationStyle (p. 100)": enum,
      "ebif (p. 100)": enum,
      "audioBufferSize (p. 101)": enum,
      "dvbNitSettings (p. 101)": {
        "networkName (p. 52)": "string",
        "networkId (p. 52)": integer,
        "repInterval (p. 52)": integer
      },
      "absentInputAudioBehavior (p. 101)": enum,


"timedMetadataBehavior (p. 101)": enum,
"timedMetadataPid (p. 101)": "string",
"pmtPid (p. 101)": "string",
"etvSignalPid (p. 101)": "string",
"bufferModel (p. 102)": enum,
"scte35Control (p. 102)": enum,
"ebpPlacement (p. 102)": enum,
"arib (p. 102)": enum,
"dvbSdtSettings (p. 102)": {
  "serviceName (p. 53)": "string",
  "serviceProviderName (p. 53)": "string",
  "repInterval (p. 53)": integer,
  "outputSdt (p. 53)": enum
},
"nullPacketBitrate (p. 102)": number,
"pcrPid (p. 102)": "string",
"transportStreamId (p. 103)": integer,
"videoPid (p. 103)": "string",
"pcrControl (p. 103)": enum,
"esRateInPes (p. 103)": enum,
"segmentationMarkers (p. 103)": enum,
"etvSignalPid (p. 103)": "string",
"dvbTdtSettings (p. 103)": {
  "repInterval (p. 58)": integer
},
"ccDescriptor (p. 104)": enum,
"patInterval (p. 104)": integer,
"etvPlatformPid (p. 104)": "string",
"dvbSubPids (p. 104)": "string",
"aribCaptionsPid (p. 104)": "string",
"scte27Pids (p. 104)": "string",
"dvbDataDts (p. 104)": "string",
"nullPacketBitrate (p. 105)": number,
"audioPid (p. 105)": "string",
"audioPids (p. 105)": "string",
"audioFramesPerPes (p. 105)": integer,
"scte35Pid (p. 105)": "string",
"pcrPid (p. 106)": "string",
"audioFramesPerPes (p. 106)": "string",
"audioPids (p. 106)": "string",
"audioFramesPerPes (p. 106)": integer,
"scte35Behavior (p. 106)": enum,
"transportStreamId (p. 106)": "string",
"videoPid (p. 106)": "string",
"pcrControl (p. 107)": enum,
"pcrPeriod (p. 107)": integer,
"programNum (p. 107)": integer,
"scte35Interval (p. 107)": integer,
"patInterval (p. 107)": integer,
"timedMetadataBehavior (p. 107)": enum
},
"audioRenditionSets (p. 119)": "string"
}
"audioOnlyImage (p. 37)": {
    "passwordParam (p. 91)": "string",
    "uri (p. 91)": "string",
    "username (p. 91)": "string"
  }
},
"nameModifier (p. 87)": "string"
},
"audioDescriptionNames (p. 112)": [
  "string"
],
"outputGroupSettings (p. 113)": {
  "archiveGroupSettings (p. 114)": {
    "destination (p. 31)": {
      "destinationRefId (p. 114)": "string"
    },
    "rolloverInterval (p. 32)": integer
  },
  "udpGroupSettings (p. 114)": {
    "inputLossAction (p. 121)": enum,
    "timedMetadataId3Frame (p. 121)": enum,
    "timedMetadataId3Period (p. 121)": integer
  },
  "msSmoothGroupSettings (p. 114)": {
    "eventId (p. 108)": "string",
    "fragmentLength (p. 108)": integer,
    "timestampOffset (p. 109)": "string",
    "segmentationMode (p. 109)": enum,
    "numRetries (p. 109)": integer,
    "acquisitionPointId (p. 109)": "string",
    "eventStopBehavior (p. 109)": enum,
    "sparseTrackType (p. 109)": enum,
    "destination (p. 109)": {
      "destinationRefId (p. 114)": "string"
    },
    "timestampOffsetMode (p. 110)": enum,
    "audioOnlyTimecodeControl (p. 110)": enum,
    "connectionRetryInterval (p. 110)": integer,
    "filecacheDuration (p. 110)": integer,
    "certificateMode (p. 110)": enum,
    "inputLossAction (p. 110)": enum,
    "sendDelayMs (p. 110)": integer,
    "eventIdMode (p. 111)": enum,
    "restartDelay (p. 111)": integer,
    "streamManifestBehavior (p. 111)": enum
  },
  "hlsGroupSettings (p. 114)": {
    "segmentsPerSubdirectory (p. 79)": integer,
    "ivInManifest (p. 79)": enum,
    "outputSelection (p. 80)": enum,
    "encryptionType (p. 80)": enum,
    "destination (p. 80)": {
      "destinationRefId (p. 114)": "string"
    },
    "indexNSegments (p. 80)": integer,
    "timedMetadataId3Frame (p. 80)": enum,
    "constantIv (p. 80)": "string",
    "baseUrlManifest (p. 80)": "string",
    "captionLanguageSetting (p. 81)": enum,
    "minSegmentLength (p. 81)": integer,
    "mode (p. 81)": enum,
    "keyProviderSettings (p. 81)": {
  }
"staticKeySettings (p. 94)": {
  "staticKeyValue (p. 119)": "string",
  "keyProviderServer (p. 120)": {
    "passwordParam (p. 91)": "string",
    "uri (p. 91)": "string",
    "username (p. 91)": "string"
  }
},
"manifestCompression (p. 81)": enum,
"ivSource (p. 81)": enum,
"tsFileMode (p. 82)": enum,
"manifestDurationFormat (p. 82)": enum,
"keyFormatVersions (p. 82)": "string",
"streamInfResolution (p. 82)": enum,
"timestampDeltaMilliseconds (p. 82)": integer,
"baseUrlContent (p. 82)": "string",
"segmentationMode (p. 82)": enum,
"captionLanguageMappings (p. 83)": [
  {
    "languageDescription (p. 46)": "string",
    "captionChannel (p. 46)": integer,
    "languageCode (p. 47)": "string"
  }
],
"clientCache (p. 83)": enum,
"codecSpecification (p. 83)": enum,
"keepSegments (p. 83)": integer,
"timedMetadataId3Period (p. 83)": integer,
"programDateTime (p. 83)": enum,
"directoryStructure (p. 83)": enum,
"keyFormat (p. 84)": "string",
"inputLossAction (p. 84)": "string",
"adMarkers (p. 84)": [
  enum
],
"programDateTimePeriod (p. 84)": integer,
"segmentLength (p. 84)": integer,
"hlsCdnSettings (p. 84)": {
  "hlsAkamaiSettings (p. 78)": {
    "httpTransferMode (p. 76)": enum,
    "salt (p. 76)": "string",
    "numRetries (p. 76)": integer,
    "restartDelay (p. 77)": integer,
    "connectionRetryInterval (p. 77)": integer,
    "filecacheDuration (p. 77)": integer,
    "token (p. 77)": "string"
  },
  "hlsWebdavSettings (p. 78)": {
    "httpTransferMode (p. 89)": enum,
    "numRetries (p. 89)": integer,
    "restartDelay (p. 89)": integer,
    "connectionRetryInterval (p. 89)": integer,
    "filecacheDuration (p. 89)": integer
  },
  "hlsBasicPutSettings (p. 78)": {
    "numRetries (p. 77)": integer,
    "restartDelay (p. 77)": integer,
    "connectionRetryInterval (p. 78)": integer,
    "filecacheDuration (p. 78)": integer
  },
  "hlsMediaStoreSettings (p. 79)": {
    "mediaStoreStorageClass (p. 86)": enum,
    "numRetries (p. 86)": integer,
    "restartDelay (p. 86)": integer,
    "connectionRetryInterval (p. 86)": integer,
"filecacheDuration (p. 86)": integer
}
"
"
"
"
"name (p. 114)": "string"
",
"languageCodeControl (p. 33)": enum,
"audioTypeControl (p. 33)": enum,
"remixSettings (p. 34)": {
  "channelMappings (p. 115)": [
    "outputChannel (p. 32)": integer,
    "inputChannelLevels (p. 32)": [
      "inputChannel (p. 90)": integer,
      "gain (p. 90)": integer
    ]
  ]
},
"channelsOut (p. 115)": integer,
"channelsIn (p. 115)": integer
",
"audioType (p. 34)": enum,
"name (p. 34)": "string",
"codecSettings (p. 34)": {
  "ac3Settings (p. 33)": {
    "drcProfile (p. 30)": enum,
    "dialnorm (p. 30)": integer,
    "codingMode (p. 30)": enum,
    "metadataControl (p. 30)": enum,
    "bitrate (p. 31)": number,
    "lfeFilter (p. 31)": enum,
    "bitstreamMode (p. 31)": enum
  },
  "aacSettings (p. 33)": {
    "vbrQuality (p. 27)": enum,
    "codingMode (p. 28)": enum,
    "profile (p. 28)": enum,
    "inputType (p. 28)": enum,
    "bitrate (p. 28)": number,
    "rawFormat (p. 28)": enum,
    "rateControlMode (p. 28)": enum,
    "sampleRate (p. 28)": number,
    "spec (p. 29)": enum
  },
  "eac3Settings (p. 33)": {
    "dialnorm (p. 60)": integer,
    "passthroughControl (p. 60)": enum,
    "drcLine (p. 60)": enum,
    "metadataControl (p. 60)": enum,
    "bitrate (p. 60)": number,
    "ltRtSurroundMixLevel (p. 61)": number,
    "surroundExMode (p. 61)": enum,
    "lfeControl (p. 61)": enum,
    "codingMode (p. 61)": enum,
    "surroundMode (p. 61)": enum,
    "attenuationControl (p. 61)": enum,
    "lfeFilter (p. 61)": enum,
    "dcFilter (p. 61)": enum,
    "phaseControl (p. 62)": enum,
    "ltRtCenterMixLevel (p. 62)": number,
"stereoDownmix (p. 62)": enum,
"bitstreamMode (p. 62)": enum,
"loRoSurroundMixLevel (p. 62)": number,
"drcRf (p. 62)": enum,
"loRoCenterMixLevel (p. 62)": number
},
"passThroughSettings (p. 33)": {
},
"mp2Settings (p. 33)": {
  "codingMode (p. 108)": enum,
  "bitrate (p. 108)": number,
  "sampleRate (p. 108)": number
},
"languageCode (p. 34)": "string",
"streamName (p. 34)": "string",
"audioNormalizationSettings (p. 34)": {
  "targetLkfs (p. 36)": number,
  "algorithmControl (p. 36)": enum,
  "algorithm (p. 36)": enum
},
"audioSelectorName (p. 35)": "string"
],
"captionDescriptions (p. 64)": [
{
  "captionSelectorName (p. 44)": "string",
  "languageDescription (p. 45)": "string",
  "name (p. 45)": "string",
  "destinationSettings (p. 45)": {
    "scte27DestinationSettings (p. 45)": {
    },
    "burnInDestinationSettings (p. 45)": {
      "xPosition (p. 41)": integer,
      "backgroundColor (p. 41)": enum,
      "yPosition (p. 41)": integer,
      "teletextGridControl (p. 41)": enum,
      "backgroundOpacity (p. 41)": integer,
      "fontOpacity (p. 42)": integer,
      "shadowOpacity (p. 42)": integer,
      "fontResolution (p. 42)": integer,
      "shadowYOffset (p. 42)": integer,
      "outlineSize (p. 42)": integer,
      "outlineColor (p. 42)": enum,
      "fontsize (p. 43)": "string",
      "shadowXOffset (p. 43)": integer,
      "alignment (p. 43)": enum,
      "shadowColor (p. 43)": enum,
      "fontColor (p. 43)": enum,
      "font (p. 43)": {
        "passwordParam (p. 91)": "string",
        "uri (p. 91)": "string",
        "username (p. 91)": "string"
      }
    },
    "teletextDestinationSettings (p. 45)": {
    },
    "webvttDestinationSettings (p. 45)": {
    },
    "ttmlDestinationSettings (p. 46)": {
      "styleControl (p. 121)": enum
    },
    "smpteTtDestinationSettings (p. 46)": {
    },
    "embeddedPlusScte20DestinationSettings (p. 46)": {
    }
  }
}];

23
"dvbSubDestinationSettings (p. 46)": {
  "xPosition (p. 54)": integer,
  "backgroundColor (p. 54)": enum,
  "yPosition (p. 55)": integer,
  "teletextGridControl (p. 55)": enum,
  "backgroundOpacity (p. 55)": integer,
  "fontOpacity (p. 55)": integer,
  "shadowOpacity (p. 55)": integer,
  "fontResolution (p. 56)": integer,
  "shadowYOffset (p. 56)": integer,
  "outlineSize (p. 56)": integer,
  "outlineColor (p. 56)": enum,
  "fontSize (p. 56)": "string",
  "shadowXOffset (p. 56)": integer,
  "alignment (p. 57)": enum,
  "shadowColor (p. 57)": enum,
  "fontColor (p. 57)": enum,
  "font (p. 57)": {
    "passwordParam (p. 91)": "string",
    "uri (p. 91)": "string",
    "username (p. 91)": "string"
  }
},
"embeddedDestinationSettings (p. 46)": {
},
"aribDestinationSettings (p. 46)": {
},
"scte20PlusEmbeddedDestinationSettings (p. 46)": {
},
"languageCode (p. 45)": "string"
},
"availConfiguration (p. 65)": {
  "availSettings (p. 39)": {
    "scte35TimeSignalApos (p. 39)": {
      "adAvailOffset (p. 117)": integer,
      "webDeliveryAllowedFlag (p. 118)": enum,
      "noRegionalBlackoutFlag (p. 118)": enum
    },
    "scte35SpliceInsert (p. 39)": {
      "adAvailOffset (p. 117)": integer,
      "webDeliveryAllowedFlag (p. 117)": enum,
      "noRegionalBlackoutFlag (p. 117)": enum
    }
  },
  "globalConfiguration (p. 65)": {
    "inputLossBehavior (p. 66)": {
      "inputLossImageType (p. 92)": enum,
      "inputLossImageColor (p. 92)": "string",
      "inputLossImageSlate (p. 92)": {
        "passwordParam (p. 91)": "string",
        "uri (p. 91)": "string",
        "username (p. 91)": "string"
      },
      "repeatFrameMsec (p. 92)": integer,
      "blackFrameMsec (p. 92)": integer
    },
    "supportLowFramerateInputs (p. 66)": enum,
    "initialAudioGain (p. 67)": integer,
    "inputEndAction (p. 67)": enum,
    "outputTimingSource (p. 67)": enum
  },
  "videoDescriptions (p. 65)": [
    {
      "input (p. 39)": integer,
      "startNumber (p. 39)": integer,
      "duration (p. 39)": integer,
      "uri (p. 39)": "string",
      "username (p. 39)": "string",
      "passwordParam (p. 39)": "string"
    }
  ]
}
"respondToAfd (p. 123)": enum,
"scalingBehavior (p. 123)": enum,
"name (p. 123)": "string",
"width (p. 123)": integer,
"sharpness (p. 123)": integer,
"codecSettings (p. 123)": {
  "h264Settings (p. 123)": {
    "minIInterval (p. 70)": integer,
    "slices (p. 70)": integer,
    "parNumerator (p. 70)": integer,
    "gopSizeUnits (p. 71)": enum,
    "maxBitrate (p. 71)": integer,
    "bitrate (p. 71)": integer,
    "bufFillPct (p. 71)": integer,
    "temporalAq (p. 71)": enum,
    "afdSignaling (p. 71)": enum,
    "timecodeInsertion (p. 72)": enum,
    "bufSize (p. 72)": integer,
    "softness (p. 72)": integer,
    "framerateControl (p. 72)": enum,
    "fixedAfd (p. 72)": enum,
    "level (p. 72)": enum,
    "lookAheadRateControl (p. 72)": enum,
    "profile (p. 73)": enum,
    "framerateNumerator (p. 73)": integer,
    "gopClosedCadence (p. 73)": integer,
    "framerateDenominator (p. 73)": integer,
    "spatialAq (p. 73)": enum,
    "entropyEncoding (p. 73)": enum,
    "adaptiveQuantization (p. 73)": enum,
    "colorMetadata (p. 74)": enum,
    "gopSize (p. 74)": number,
    "numRefFrames (p. 74)": integer,
    "gopBReference (p. 74)": enum,
    "syntax (p. 74)": enum,
    "parControl (p. 74)": enum,
    "parDenominator (p. 74)": integer,
    "sceneChangeDetect (p. 75)": enum,
    "gopNumBFrames (p. 75)": integer,
    "flickerAq (p. 75)": enum,
    "rateControlMode (p. 75)": enum
  }
},
"height (p. 124)": integer
},
"availBlanking (p. 65)": {
  "state (p. 38)": enum,
  "availBlankingImage (p. 38)": {
    "passwordParam (p. 91)": "string",
    "uri (p. 91)": "string",
    "username (p. 91)": "string"
  }
},
"blackoutSlate (p. 65)": {
  "networkEndBlackoutImage (p. 39)": {
    "passwordParam (p. 91)": "string",
    "uri (p. 91)": "string",
    "username (p. 91)": "string"
  },
  "networkEndBlackout (p. 39)": enum,
  "networkId (p. 39)": "string",
  "blackoutSlateImage (p. 40)": {
    "passwordParam (p. 91)": "string",
    "uri (p. 91)": "string",
    "username (p. 91)": "string"
  }
}
"username (p. 91)": "string",
"state (p. 40)": enum,
"id (p. 48)": "string",
"pipelinesRunningCount (p. 49)": integer,
"state (p. 49)": enum,
"arn (p. 49)": "string",
"egressEndpoints (p. 49)": [
  {
    "sourceIp (p. 49)": "string"
  }
]
}

Example InvalidRequest

{
  "message (p. 94)": "string"
}

Example AccessDenied

{
  "message (p. 31)": "string"
}

Example ResourceConflict

{
  "message (p. 115)": "string"
}

Example ChannelConfigurationValidationError

{
  "validationErrors (p. 49)": [
    {
      "errorMessage (p. 122)": "string",
      "elementPath (p. 122)": "string"
    }
  ],
  "message (p. 49)": "string"
}

Example LimitExceeded

{
  "message (p. 94)": "string"
}

Example InternalServiceError

{
  "message (p. 94)": "string"
}
Properties

AacCodingMode (Enum)

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.

AD_RECEIVER_MIX
CODING_MODE_1_0
CODING_MODE_1_1
CODING_MODE_2_0
CODING_MODE_5_1

AacInputType (Enum)

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.

BROADCASTER_MIXED_AD
NORMAL

AacProfile (Enum)

AAC Profile.

HEV1
HEV2
LC

AacRateControlMode (Enum)

Rate Control Mode.

CBR
VBR

AacRawFormat (Enum)

Sets LATM / LOAS AAC output for raw containers.

LATM_LOAS
NONE

AacSettings

vbrQuality

VBR Quality Level - Only used if rateControlMode is VBR.
Properties

**Type: AacVbrQuality (p. 29)**
**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. 27)**
**Required: False**

profile
AAC Profile.

**Type: AacProfile (p. 27)**
**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. 27)**
**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. 27)**
**Required: False**

rateControlMode
Rate Control Mode.

**Type: AacRateControlMode (p. 27)**
**Required: False**

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

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

- **Type**: AacSpec (p. 29)
- **Required**: False

**AacSpec (Enum)**

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

- MPEG2
- MPEG4

**AacVbrQuality (Enum)**

VBR Quality Level - Only used if rateControlMode is VBR.

- HIGH
- LOW
- MEDIUM_HIGH
- MEDIUM_LOW

**Ac3BitstreamMode (Enum)**

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

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

**Ac3CodingMode (Enum)**

Dolby Digital coding mode. Determines number of channels.

- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_3_2_LFE

**Ac3DrcProfile (Enum)**

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

- FILM_STANDARD
NONE

**Ac3LfeFilter (Enum)**

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

- DISABLED
- ENABLED

**Ac3MetadataControl (Enum)**

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.

- FOLLOW_INPUT
- USE_CONFIGURED

**Ac3Settings**

**drcProfile**

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

- **Type:** Ac3DrcProfile (p. 29)
- **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. 29)
- **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. 30)
- **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. 30)
  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. 29)
  Required: False

AccessDenied

message

  Type: string
  Required: False

AfdSignaling (Enum)

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.

  AUTO
  FIXED
  NONE

ArchiveContainerSettings

m2tsSettings

  Type: M2tsSettings (p. 98)
  Required: False

ArchiveGroupSettings

destination

A directory and base filename where archive files should be written. If the base filename portion of the URI is left blank, the base filename of the first input will be automatically inserted.
Properties

**Type**: OutputLocationRef (p. 114)
**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**

**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. 31)
**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**

**AribSourceSettings**

**AudioChannelMapping**

**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.
**Properties**

**Type**: Array of type `InputChannelLevel (p. 90)`
**Required**: True

**AudioCodecSettings**

- **ac3Settings**
  **Type**: `Ac3Settings (p. 30)`
  **Required**: False

- **aacSettings**
  **Type**: `AacSettings (p. 27)`
  **Required**: False

- **eac3Settings**
  **Type**: `Eac3Settings (p. 60)`
  **Required**: False

- **passThroughSettings**
  **Type**: `PassThroughSettings (p. 115)`
  **Required**: False

- **mp2Settings**
  **Type**: `Mp2Settings (p. 108)`
  **Required**: False

**AudioDescription**

- **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. 35)`
  **Required**: False

- **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. 35)`
  **Required**: False
**remixSettings**

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

- **Type**: RemixSettings (p. 115)
- **Required**: False

**audioType**

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

- **Type**: AudioType (p. 38)
- **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. 33)
- **Required**: True

**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. 36)
- **Required**: False
**audioSelectorName**

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

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

**AudioDescriptionAudioTypeControl (Enum)**

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.

- FOLLOW_INPUT
- USE_CONFIGURED

**AudioDescriptionLanguageCodeControl (Enum)**

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.

- FOLLOW_INPUT
- USE_CONFIGURED

**AudioLanguageSelection**

**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. 35)
- **Required**: False

**languageCode**

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

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

**AudioLanguageSelectionPolicy (Enum)**

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.

- LOOSE
STRICT

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

- ITU_1770_1
- ITU_1770_2

AudioNormalizationAlgorithmControl (Enum)
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.

- CORRECT_AUDIO

AudioNormalizationSettings

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

- Type: number
- Required: False
- 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. 36)
- 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. 36)
- Required: False

AudioOnlyHlsSettings

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. 37)  
Required: False

audioGroupId

Specifies the group to which the audio Rendition belongs.

Type: string  
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. 91)  
Required: False

AudioOnlyHlsTrackType (Enum)

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

ALTERNATE_AUDIO_AUTO_SELECT  
ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT  
ALTERNATE_AUDIO_NOT_AUTO_SELECT  
AUDIO_ONLY_VARIANT_STREAM

AudioPidSelection

pid

Selects a specific PID from within a source.

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

37
AudioSelector

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

selectorSettings
The audio selector settings.

Type: AudioSelectorSettings (p. 38)
Required: False

AudioSelectorSettings

audioLanguageSelection

Type: AudioLanguageSelection (p. 35)
Required: False

audioPidSelection

Type: AudioPidSelection (p. 37)
Required: False

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

CLEAN_EFFECTS
HEARING_IMPAIRED
UNDEFINED
VISUAL_IMPAIRED_COMMENTARY

AvailBlanking

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

Type: AvailBlankingState (p. 39)
Required: False

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

Type: InputLocation (p. 91)
Required: False
AvailBlankingState (Enum)

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

DISABLED
ENABLED

AvailConfiguration

availSettings

Ad avail settings.

Type: AvailSettings (p. 39)
Required: False

AvailSettings

scte35TimeSignalApos

Type: Scte35TimeSignalApos (p. 117)
Required: False

scte35SpliceInsert

Type: Scte35SpliceInsert (p. 117)
Required: False

BlackoutSlate

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. 91)
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. 40)
Required: False

networkId

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

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

**blackoutSlateImage**

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

**Type**: InputLocation (p. 91)  
**Required**: False

**state**

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

**Type**: BlackoutSlateState (p. 40)  
**Required**: False

**BlackoutSlateNetworkEndBlackout (Enum)**

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".

- DISABLED
- ENABLED

**BlackoutSlateState (Enum)**

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

- DISABLED
- ENABLED

**BurnInAlignment (Enum)**

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.

- CENTERED
- LEFT
- SMART

**BurnInBackgroundColor (Enum)**

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

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit 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. 40)
- **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. 44)
- **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

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

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

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. 44)
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

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

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. 40)
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. 44)
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. 44)
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. 91)
**Required:** False

**BurnInFontColor (Enum)**

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

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

**BurnInOutlineColor (Enum)**

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

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

**BurnInShadowColor (Enum)**

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

BLACK
NONE
WHITE

**BurnInTeletextGridControl (Enum)**

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.

FIXED
SCALED

**CaptionDescription**

**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

destinationSettings

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

Type: CaptionDestinationSettings (p. 45)
Required: False

languageCode


Type: string
Required: False

CaptionDestinationSettings

scte27DestinationSettings

Type: Scte27DestinationSettings (p. 116)
Required: False

burnInDestinationSettings

Type: BurnInDestinationSettings (p. 41)
Required: False

teletextDestinationSettings

Type: TeletextDestinationSettings (p. 120)
Required: False

webvttDestinationSettings

Type: WebvttDestinationSettings (p. 126)
Required: False
**Properties**

**ttmlDestinationSettings**
- **Type**: TtmlDestinationSettings (p. 121)
- **Required**: False

**smpteTtDestinationSettings**
- **Type**: SmpteTtDestinationSettings (p. 119)
- **Required**: False

**embeddedPlusScte20DestinationSettings**
- **Type**: EmbeddedPlusScte20DestinationSettings (p. 63)
- **Required**: False

**dvbSubDestinationSettings**
- **Type**: DvbSubDestinationSettings (p. 54)
- **Required**: False

**embeddedDestinationSettings**
- **Type**: EmbeddedDestinationSettings (p. 63)
- **Required**: False

**aribDestinationSettings**
- **Type**: AribDestinationSettings (p. 32)
- **Required**: False

**scte20PlusEmbeddedDestinationSettings**
- **Type**: Scte20PlusEmbeddedDestinationSettings (p. 116)
- **Required**: False

**CaptionLanguageMapping**

**languageDescription**
- Textual description of language
  - **Type**: string
  - **Required**: False

**captionChannel**
- Channel to insert closed captions. Each channel mapping must have a unique channel number (maximum of 4)
  - **Type**: integer
  - **Required**: False
  - **Minimum**: 1
  - **Maximum**: 4
languageCode

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

  Type: string
  Required: False

**CaptionSelector**

**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

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. 47)
  Required: False

**CaptionSelectorSettings**

**embeddedSourceSettings**

  Type: EmbeddedSourceSettings (p. 63)
  Required: False

**scte20SourceSettings**

  Type: Scte20SourceSettings (p. 116)
  Required: False

**dvbSubSourceSettings**

  Type: DvbSubSourceSettings (p. 58)
  Required: False

**teletextSourceSettings**

  Type: TeletextSourceSettings (p. 120)
  Required: False
### aribSourceSettings
- **Type:** [AribSourceSettings](#)
- **Required:** False

### scte27SourceSettings
- **Type:** [Scte27SourceSettings](#)
- **Required:** False

---

### Channel

#### inputAttachments
List of input attachments for channel.
- **Type:** Array of type [InputAttachment](#)
- **Required:** False

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

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

#### encoderSettings
- **Type:** [EncoderSettings](#)
- **Required:** False

#### id
The unique id of the channel.
- **Type:** string
- **Required:** False
pipelinesRunningCount

The number of currently healthy pipelines.

   Type: integer
   Required: False

state

   Type: ChannelState (p. 49)
   Required: False

arn

The unique arn of the channel.

   Type: string
   Required: False

egressEndpoints

The endpoints where outgoing connections initiate from

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

ChannelConfigurationValidationError

validationErrors

A collection of validation error responses from attempting to create a channel with a bouquet of settings.

   Type: Array of type ValidationError (p. 122)
   Required: False

message

   Type: string
   Required: False

ChannelEgressEndpoint

sourceIp

Public IP of where a channel's output comes from

   Type: string
   Required: False

ChannelState (Enum)

CREATING
CREATE_FAILED
IDLE
STARTING
RUNNING
RECOVERING
STOPPING
DELETING
DELETED

ChannelSummary

inputAttachments
List of input attachments for channel.
  Type: Array of type InputAttachment (p. 89)
  Required: False

destination
The Amazon Resource Name (ARN) of the role assumed when running the Channel.
  Type: string
  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. 113)
  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

pipelinesRunningCount
The number of currently healthy pipelines.
  Type: integer
  Required: False
state

  Type: ChannelState
  Required: False

arn

  The unique arn of the channel.

  Type: string
  Required: False

egressEndpoints

  The endpoints where outgoing connections initiate from

  Type: Array of type ChannelEgressEndpoint
  Required: False

CreateChannel

inputAttachments

  List of input attachments for channel.

  Type: Array of type InputAttachment
  Required: False

reserved

  Reserved for future use.

  Type: string
  Required: False

roleArn

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

  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

destinations

  Type: Array of type OutputDestination
  Required: False
name
Name of channel.

  Type: string
  Required: False

encoderSettings

  Type: EncoderSettings (p. 64)
  Required: False

CreateChannelResultModel
channel

  Type: Channel (p. 48)
  Required: False

DvbNitSettings

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

  Type: string
  Required: True

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

  Type: integer
  Required: True
  Minimum: 0
  Maximum: 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 (Enum)
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.
**DvbSdtSettings**

**serviceName**

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

_Type:_ string  
_Required:_ False

**serviceProviderName**

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

_Type:_ string  
_Required:_ False

**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. 52)  
_Required:_ False

**DvbSubDestinationAlignment (Enum)**

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.

CENTERED  
LEFT
Properties

SMART

**DvbSubDestinationBackgroundColor (Enum)**

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

- BLACK
- NONE
- WHITE

**DvbSubDestinationFontColor (Enum)**

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

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

**DvbSubDestinationOutlineColor (Enum)**

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

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

**DvbSubDestinationSettings**

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit 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.
**Properties**

**Type**: DvbSubDestinationBackgroundColor (p. 54)
**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. 57)
**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

**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
**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

**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. 54)
- **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

**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
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. 53)
  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. 57)
  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. 54)
  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. 91)
  Required: False

DvbSubDestinationShadowColor (Enum)

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

  BLACK
  NONE
  WHITE

DvbSubDestinationTeletextGridControl (Enum)

 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.

  FIXED
DvbSubSourceSettings

pid

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

Type: integer
Required: False
Minimum: 1

DvbTdtSettings

repInterval

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

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

Eac3AttenuationControl (Enum)

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

ATTENUATE_3_DB
NONE

Eac3BitstreamMode (Enum)

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

COMMENTARY
COMPLETE_MAIN
EMERGENCY
HEARING_IMPAIRED
VISUALLY_IMPAIRED

Eac3CodingMode (Enum)

Dolby Digital Plus coding mode. Determines number of channels.

CODING_MODE_1_0
CODING_MODE_2_0
CODING_MODE_3_2

Eac3DcFilter (Enum)

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

**Eac3DrcLine (Enum)**

Sets the Dolby dynamic range compression profile.

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

**Eac3DrcRf (Enum)**

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

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

**Eac3LfeControl (Enum)**

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

- LFE
- NO_LFE

**Eac3LfeFilter (Enum)**

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

- DISABLED
- ENABLED

**Eac3MetadataControl (Enum)**

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.

- FOLLOW_INPUT
- USE_CONFIGURED

**Eac3PassthroughControl (Enum)**

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.
NO_PASSTHROUGH
WHEN_POSSIBLE

**Eac3PhaseControl (Enum)**

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

- 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. 59)
- **Required**: False

**drcLine**

Sets the Dolby dynamic range compression profile.

- **Type**: Eac3DrcLine (p. 59)
- **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. 59)
- **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. 63)
  Required: False

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

  Type: Eac3LfeControl (p. 59)
  Required: False

codingMode
Dolby Digital Plus coding mode. Determines number of channels.

  Type: Eac3CodingMode (p. 58)
  Required: False

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

  Type: Eac3SurroundMode (p. 63)
  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. 58)
  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. 59)
  Required: False

dcFilter
When set to enabled, activates a DC highpass filter for all input channels.
**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. 60)
- **Required:** False

**ltRtCenterMixLevel**

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

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

**stereoDownmix**

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

- **Type:** Eac3StereoDownmix (p. 63)
- **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. 58)
- **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. 59)
- **Required:** False

**loRoCenterMixLevel**

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

- **Type:** number
Required: False

Eac3StereoDownmix (Enum)

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

- DPL2
- LO_RO
- LT_RT
- NOT_INDICATED

Eac3SurroundExMode (Enum)

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

- DISABLED
- ENABLED
- NOT_INDICATED

Eac3SurroundMode (Enum)

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

- DISABLED
- ENABLED
- NOT_INDICATED

EmbeddedConvert608To708 (Enum)

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.

- DISABLED
- UPCONVERT

EmbeddedDestinationSettings

EmbeddedPlusScte20DestinationSettings

EmbeddedScte20Detection (Enum)

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

- AUTO
- OFF

EmbeddedSourceSettings

scte20Detection

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

Type: EmbeddedScte20Detection (p. 63)
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. 63)
Required: False

source608TrackNumber
This field is unused and deprecated.

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

EncoderSettings

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

Type: TimecodeConfig (p. 120)
Required: True

outputGroups

Type: Array of type OutputGroup (p. 113)
Required: True

audioDescriptions

Type: Array of type AudioDescription (p. 33)
Required: True

captionDescriptions
Settings for caption decriptions

Type: Array of type CaptionDescription (p. 44)
**Properties**

**Required**: False

**availConfiguration**

Event-wide configuration settings for ad avail insertion.

**Type**: `AvailConfiguration (p. 39)`

**globalConfiguration**

Configuration settings that apply to the event as a whole.

**Type**: `GlobalConfiguration (p. 66)`

**Required**: False

**videoDescriptions**

**Type**: Array of type `VideoDescription (p. 123)`

**Required**: True

**availBlanking**

Settings for ad avail blanking.

**Type**: `AvailBlanking (p. 38)`

**Required**: False

**blackoutSlate**

Settings for blackout slate.

**Type**: `BlackoutSlate (p. 39)`

**Required**: False

**FecOutputIncludeFec (Enum)**

Enables column only or column and row based FEC

- COLUMN
- COLUMN_AND_ROW

**FecOutputSettings**

**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
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. 65)
Required: False

FixedAfd (Enum)

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

AFD_0000
AFD_0010
AFD_0011
AFD_0100
AFD_1000
AFD_1001
AFD_1010
AFD_1011
AFD_1101
AFD_1110
AFD_1111

GlobalConfiguration

inputLossBehavior

Settings for system actions when input is lost.

Type: InputLossBehavior (p. 92)
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. 67)
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 an input completes (e.g. end-of-file.) Options include immediately switching to the next sequential input (via "switchInput"), switching to the next input and looping back to the first input when last input ends (via "switchAndLoopInputs") or not switching inputs and instead transcoding black / color / slate images per the "Input Loss Behavior" configuration until an activateInput REST command is received (via "none").

Type: GlobalConfigurationInputEndAction (p. 67)
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. 67)
Required: False

GlobalConfigurationInputEndAction (Enum)

Indicates the action to take when an input completes (e.g. end-of-file.) Options include immediately switching to the next sequential input (via "switchInput"), switching to the next input and looping back to the first input when last input ends (via "switchAndLoopInputs") or not switching inputs and instead transcoding black / color / slate images per the "Input Loss Behavior" configuration until an activateInput REST command is received (via "none").

NONE
SWITCH_AND_LOOP_INPUTS

GlobalConfigurationLowFramerateInputs (Enum)

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.

DISABLED
ENABLED

GlobalConfigurationOutputTimingSource (Enum)

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.

INPUT_CLOCK
SYSTEM_CLOCK

H264AdaptiveQuantization (Enum)
Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.
  
  HIGH
  HIGHER
  LOW
  MAX
  MEDIUM
  OFF

H264ColorMetadata (Enum)
Includes colorspace metadata in the output.
  
  IGNORE
  INSERT

H264EntropyEncoding (Enum)
Entropy encoding mode. Use cabac (must be in Main or High profile) or cavlc.
  
  CABAC
  CAVLC

H264FlickerAq (Enum)
If set to enabled, adjust quantization within each frame to reduce flicker or 'pop' on I-frames.
  
  DISABLED
  ENABLED

H264FramerateControl (Enum)
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.
  
  INITIALIZE_FROM_SOURCE
  SPECIFIED

H264GopBReference (Enum)
If enabled, use reference B frames for GOP structures that have B frames > 1.
  
  DISABLED
  ENABLED

H264GopSizeUnits (Enum)
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.
**H264Level (Enum)**

H.264 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 (Enum)**

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

- HIGH
- LOW
- MEDIUM

**H264ParControl (Enum)**

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.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**H264Profile (Enum)**

H.264 Profile.

- BASELINE
- HIGH
- HIGH_10BIT
- HIGH_422
- HIGH_422_10BIT
H264RateControlMode (Enum)
Rate control mode.
   CBR
   VBR

H264ScanType (Enum)
Sets the scan type of the output to progressive or top-field-first interlaced.
   INTERLACED
   PROGRESSIVE

H264SceneChangeDetect (Enum)
Scene change detection. Inserts I-frames on scene changes when enabled.
   DISABLED
   ENABLED

H264Settings

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. 68)  
**Required**: False

**maxBitrate**

Maximum bitrate in bits/second (for VBR mode only).

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

**bitrate**

Average bitrate in bits/second. Required for VBR, CBR, and ABR. For MS Smooth outputs, bitrates must be unique when 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. 76)  
**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. 31)  
**Required**: False
Properties

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. 76)
  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. 68)
  Required: False

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. 66)
  Required: False

level
H.264 Level.

  Type: H264Level (p. 69)
  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.
**Properties**

**Type:** H264LookAheadRateControl (p. 69)
**Required:** False

**profile**
H.264 Profile.

**Type:** H264Profile (p. 69)
**Required:** False

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

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

**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

**framerateDenominator**
Framerate denominator.

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

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

**Type:** H264SpatialAq (p. 75)
**Required:** False

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

**Type:** H264EntropyEncoding (p. 68)
**Required:** False

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

**Type:** H264AdaptiveQuantization (p. 68)
**colorMetadata**

Includes colorspace metadata in the output.

- **Type:** H264ColorMetadata (p. 68)
- **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. 68)
- **Required:** False

**syntax**

Produces a bitstream compliant with SMPTE RP-2027.

- **Type:** H264Syntax (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. 69)
- **Required:** False

**parDenominator**

Pixel Aspect Ratio denominator.
Properties

Type: integer  
Required: False  
Minimum: 1

sceneChangeDetect

Scene change detection. Inserts I-frames on scene changes when enabled.

Type: H264SceneChangeDetect (p. 70)  
Required: False

scanType

Sets the scan type of the output to progressive or top-field-first interlaced.

Type: H264ScanType (p. 70)  
Required: False

gopNumBFrames

Number of B-frames between reference frames.

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

flickerAq

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

Type: H264FlickerAq (p. 68)  
Required: False

rateControlMode

Rate control mode.

Type: H264RateControlMode (p. 70)  
Required: False

H264SpatialAq (Enum)

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

DISABLED  
ENABLED

H264Syntax (Enum)

Produces a bitstream compliant with SMPTE RP-2027.

DEFAULT
RP2027

**H264TemporalAq (Enum)**

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

- DISABLED
- ENABLED

**H264TimecodeInsertionBehavior (Enum)**

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

- DISABLED
- PIC_TIMING_SEI

**HlsAdMarkers (Enum)**

- ADOBE
- ELEMENTAL
- ELEMENTAL_SCTE35

**HlsAkamaiHttpTransferMode (Enum)**

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

- CHUNKED
- NON_CHUNKED

**HlsAkamaiSettings**

**httpTransferMode**

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

- Type: HlsAkamaiHttpTransferMode (p. 76)
- Required: False

**salt**

Salt for authenticated Akamai.

- Type: string
- 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**

**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

**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 (Enum)**

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

- **INSERT**
- **NONE**
- **OMIT**

**HlsCdnSettings**

**hlsAkamaiSettings**

- **Type**: [HlsAkamaiSettings](p. 76)
- **Required**: False

**hlsWebdavSettings**

- **Type**: [HlsWebdavSettings](p. 89)
- **Required**: False

**hlsBasicPutSettings**

- **Type**: [HlsBasicPutSettings](p. 77)
- **Required**: False
hlsMediaStoreSettings

Type: HlsMediaStoreSettings (p. 86)
Required: False

HlsClientCache (Enum)

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.

   DISABLED
   ENABLED

HlsCodecSpecification (Enum)

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

   RFC_4281
   RFC_6381

HlsDirectoryStructure (Enum)

Place segments in subdirectories.

   SINGLE_DIRECTORY
   SUBDIRECTORY_PER_STREAM

HlsEncryptionType (Enum)

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

   AES128
   SAMPLE_AES

HlsGroupSettings

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. 85)
properties

Required: False

outputSelection
Generates the .m3u8 playlist file for this HLS output group. The segmentsOnly option will output segments without the .m3u8 file.

Type: HlsOutputSelection (p. 87)
Required: False

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

Type: OutputLocationRef (p. 114)
Required: True

indexNSegments
Number of segments to keep in the playlist (.m3u8) file. mode must be "vod" for this setting to have an effect, and this number should be less than or equal to keepSegments.

Type: integer
Required: False
Minimum: 1

timedMetadataId3Frame
Indicates ID3 frame that has the timecode.

Type: HlsTimedMetadataId3Frame (p. 88)
Required: False

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

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. 78)
Required: False

minSegmentLength

When set, minSegmentLength 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 set to "vod", keeps and indexes all segments starting with the first segment. If set to "live" segments will age out and only the last keepSegments number of segments will be retained.

Type: HlsMode (p. 87)
Required: False

keyProviderSettings

The key provider settings.

Type: KeyProviderSettings (p. 94)
Required: False

manifestCompression

When set to gzip, compresses HLS playlist.

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

**tsFileMode**

When set to "singleFile", emits the program as a single media resource (.ts) file, and uses #EXT-X-BYTERANGE tags to index segment for playback. Playback of VOD mode content during event is not guaranteed due to HTTP server caching.

*Type: HlsTsFileMode (p. 88)*

*Required: False*

**manifestDurationFormat**

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

*Type: HlsManifestDurationFormat (p. 86)*

*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. 88)*

*Required: False*

**timestampDeltaMilliseconds**

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

*Type: integer*

*Required: False*

**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**

When set to useInputSegmentation, the output segment or fragment points are set by the RAI markers from the input streams.

*Type: HlsSegmentationMode (p. 88)*

*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. 46)
- **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. 79)
- **Required:** False

**codecSpecification**

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

- **Type:** HlsCodecSpecification (p. 79)
- **Required:** False

**keepSegments**

Number of segments to retain in the destination directory. mode must be "live" for this setting to have an effect.

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

**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. 87)
- **Required:** False

**directoryStructure**

Place segments in subdirectories.

- **Type:** HlsDirectoryStructure (p. 79)
- **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. 91)
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. 76)
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. 78)
Required: False

HlsInputSettings

retries

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

**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 (Enum)**
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.

- EXCLUDE
- INCLUDE

**HlsIvSource (Enum)**
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.

- EXPLICIT
- FOLLOWSEGMENT NUMBER

**HlsManifestCompression (Enum)**
When set to gzip, compresses HLS playlist.
GZIP
NONE

**HlsManifestDurationFormat (Enum)**
Indicates whether the output manifest should use floating point or integer values for segment duration.

  - FLOATING_POINT
  - INTEGER

**HlsMediaStoreSettings**

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

  - Type: HlsMediaStoreStorageClass (p. 87)
  - 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 (Enum)**

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

- TEMPORAL

**HlsMode (Enum)**

If set to "vod", keeps and indexes all segments starting with the first segment. If set to "live" segments will age out and only the last keepSegments number of segments will be retained.

- LIVE
- VOD

**HlsOutputSelection (Enum)**

Generates the .m3u8 playlist file for this HLS output group. The segmentsOnly option will output segments without the .m3u8 file.

- MANIFESTS_AND_SEGMENTS
- SEGMENTS_ONLY

**HlsOutputSettings**

**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. 88)
- **Required:** True

**nameModifier**

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

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

**HlsProgramDateTime (Enum)**

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

**HlsSegmentationMode (Enum)**

When set to `useInputSegmentation`, the output segment or fragment points are set by the RAI markers from the input streams.

- USE_INPUT_SEGMENTATION
- USE_SEGMENT_DURATION

**HlsSettings**

**standardHlsSettings**

_Type: StandardHlsSettings (p. 119)_

_REQUIRED: False_

**audioOnlyHlsSettings**

_Type: AudioOnlyHlsSettings (p. 36)_

_REQUIRED: False_

**HlsStreamInfResolution (Enum)**

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

- EXCLUDE
- INCLUDE

**HlsTimedMetadataId3Frame (Enum)**

Indicates ID3 frame that has the timecode.

- NONE
- PRIV
- TDRL

**HlsTsFileMode (Enum)**

When set to "singleFile", emits the program as a single media resource (.ts) file, and uses #EXT-X-BYTERANGE tags to index segment for playback. Playback of VOD mode content during event is not guaranteed due to HTTP server caching.

- SEGMENTED_FILES
- SINGLE_FILE

**HlsWebdavHttpTransferMode (Enum)**

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

- CHUNKED
NON_CHUNKED

**HlsWebdavSettings**

**httpTransferMode**

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

- **Type:** HlsWebdavHttpTransferMode (p. 88)
- **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

**InputAttachment**

**inputId**

The ID of the input
Properties

**Type**: string

**Required**: False

**inputSettings**

Settings of an input (caption selector, etc.)

**Type**: InputSettings (p. 93)

**Required**: False

**InputChannelLevel**

**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

**InputDeblockFilter (Enum)**

Enable or disable the deblock filter when filtering.

- DISABLED
- ENABLED

**InputDenoiseFilter (Enum)**

Enable or disable the denoise filter when filtering.

- DISABLED
- ENABLED

**InputFilter (Enum)**

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

- AUTO
- DISABLED
FORCED

**InputLocation**

**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 rtmpEndpoint 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 (Enum)**
Parameter that control output group behavior on input loss.

  - EMIT_OUTPUT  
  - PAUSE_OUTPUT

**InputLossActionForMsSmoothOut (Enum)**
Parameter that control output group behavior on input loss.

  - EMIT_OUTPUT  
  - PAUSE_OUTPUT

**InputLossActionForUdpOut (Enum)**
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.

  - DROP_PROGRAM  
  - DROP_TS
**InputLossBehavior**

**inputLossImageType**
Indicates whether to substitute a solid color or a slate into the output after input loss exceeds blackFrameMsec.

*Type: InputLossImageType (p. 92)*
*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*

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

*Type: InputLocation (p. 91)*
*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: 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 (Enum)**
Indicates whether to substitute a solid color or a slate into the output after input loss exceeds blackFrameMsec.

COLOR
SLATE

**InputSettings**

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

- **Type**: InputSourceEndBehavior (p. 94)
- **Required**: False

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

- **Type**: Array of type AudioSelector (p. 38)
- **Required**: False

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

- **Type**: InputDeblockFilter (p. 90)
- **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. 90)
- **Required**: False

**networkInputSettings**
Input settings.

- **Type**: NetworkInputSettings (p. 112)
- **Required**: True

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

- **Type**: VideoSelector (p. 124)
- **Required**: False

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

- **Type**: integer
Properties

**Required**: False
**Minimum**: 1
**Maximum**: 5

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

- **Type**: Array of type CaptionSelector (p. 47)
- **Required**: False

**denoiseFilter**
Enable or disable the denoise filter when filtering.

- **Type**: InputDenoiseFilter (p. 90)
- **Required**: False

**InputSourceEndBehavior (Enum)**
Loop input if it is a file. This allows a file input to be streamed indefinitely.

- CONTINUE
- LOOP

**InternalServerError**

**message**

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

**InvalidRequest**

**message**

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

**KeyProviderSettings**

**staticKeySettings**

- **Type**: StaticKeySettings (p. 119)
- **Required**: False

**LimitExceeded**

**message**

- **Type**: string
Required: False

**ListChannelsResultModel**

`channels`

- **Type**: Array of type `ChannelSummary (p. 50)`
- **Required**: False

`nextToken`

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

**M2tsAbsentInputAudioBehavior (Enum)**

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.

- DROP
- ENCODE_SILENCE

**M2tsArib (Enum)**

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

- DISABLED
- ENABLED

**M2tsAribCaptionsPidControl (Enum)**

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.

- AUTO
- USE_CONFIGURED

**M2tsAudioBufferModel (Enum)**

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

- ATSC
- DVB

**M2tsAudioInterval (Enum)**

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.
**M2tsAudioStreamType (Enum)**

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

- ATSC
- DVB

**M2tsBufferModel (Enum)**

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.

- MULTIPLEX
- NONE

**M2tsCcDescriptor (Enum)**

When set to enabled, generates captionServiceDescriptor in PMT.

- DISABLED
- ENABLED

**M2tsEbifControl (Enum)**

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

- NONE
- PASSTHROUGH

**M2tsEbpPlacement (Enum)**

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.

- VIDEO_AND_AUDIO_PIDS
- VIDEO_PID

**M2tsEsRateInPes (Enum)**

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

- EXCLUDE
- INCLUDE

**M2tsKlv (Enum)**

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

- NONE
- PASSTHROUGH
**M2tsPcrControl (Enum)**

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.

```java
CONFIGURED_PCR_PERIOD
PCR_EVERY_PES_PACKET
```

**M2tsRateMode (Enum)**

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.

```java
CBR
VBR
```

**M2tsScte35Control (Enum)**

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

```java
NONE
PASSTHROUGH
```

**M2tsSegmentationMarkers (Enum)**

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.

```java
EBP
EBP_LEGACY
NONE
PSI_SEGSTART
RAI_ADAPT
RAI_SEGSTART
```

**M2tsSegmentationStyle (Enum)**

The segmentation style parameter controls how segmentation markers are inserted into the transport stream. With avails, it is possible that segments may be truncated, which can influence where future segmentation markers are inserted. When a segmentation style of "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.

```java
MAINTAIN_CADENCE
RESET_CADENCE
```
**M2tsSettings**

**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. 96)  
*Required:* False

**ecmPid**
Packet Identifier (PID) for ECM in the transport stream. Only enabled when Simulcrypt is enabled. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

*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. 95)  
*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

**segmentationTime**
The length in seconds of each segment. Required unless markers is set to None_.

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

**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

**Type**: M2tsRateMode (p. 97)  
**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. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

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

**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. 95)  
**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

**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).
**Properties**

**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

**pcrPeriod**

Maximum time in milliseconds between Program Clock Reference (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. Valid values are 0, 10..1000.

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

**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. 97)  
**Required**: False

**ebif**

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

**Type**: M2tsEbifControl (p. 96)  
**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. 95)
- **Required:** False

**dvbNitSettings**

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

- **Type:** DvbNitSettings (p. 52)
- **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. 95)
- **Required:** False

**timedMetadataBehavior**

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

- **Type:** M2tsTimedMetadataBehavior (p. 105)
- **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. 96)
Required: False

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

Type: M2tsScte35Control (p. 97)
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. 96)
Required: False

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

Type: M2tsArib (p. 95)
Required: False

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

Type: DvbSdtSettings (p. 53)
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

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

**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

**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. 97)
- **Required:** False

**esRateInPes**

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

- **Type:** M2tsEsRateInPes (p. 96)
- **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. 97)
- **Required:** False

**klv**

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

- **Type:** M2tsKlv (p. 96)
- **Required:** False

**dvbTdtSettings**

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

**Type**: DvbTdtSettings (p. 58)
**Required**: False

**ccDescriptor**

When set to enabled, generates captionServiceDescriptor in PMT.

**Type**: M2tsCcDescriptor (p. 96)
**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 (0x20)..<8182 (0x1ff6).

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

**M2tsTimedMetadataBehavior (Enum)**

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

- NO_PASSTHROUGH  
- PASSTHROUGH

**M3u8PcrControl (Enum)**

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.

- CONFIGURED_PCR_PERIOD  
- PCR_EVERY_PES_PACKET

**M3u8Scte35Behavior (Enum)**

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

- NO_PASSTHROUGH  
- PASSTHROUGH

**M3u8Settings**

**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**

ThePlatform-protected transport streams using 'microsoft' as Target Client include an ECM stream. This ECM stream contains the size, IV, and PTS of every sample in the transport stream. This stream PID is specified here. This PID has no effect on non ThePlatform-protected streams.

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

**scte35Behavior**

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

**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
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. 105)
    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

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. A value of "0" writes out the PMT once per segment file.

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

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

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

    Type: M3u8TimedMetadataBehavior (p. 108)
    Required: False
**M3u8TimedMetadataBehavior (Enum)**

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

- NO_PASSTHROUGH
- PASSTHROUGH

**Mp2CodingMode (Enum)**

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

- CODING_MODE_1_0
- CODING_MODE_2_0

**Mp2Settings**

**codingMode**

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

- Type: Mp2CodingMode (p. 108)
- Required: False

**bitrate**

Average bitrate in bits/second.

- Type: number
- Required: False

**sampleRate**

Sample rate in Hz.

- Type: number
- Required: False

**MsSmoothGroupSettings**

**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

**fragmentLength**

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

- Type: integer
**Properties**

- **timestampOffset**
  Timestamp offset for the event. Only used if timestampOffsetMode is set to useConfiguredOffset.
  - **Type:** string
  - **Required:** False

- **segmentationMode**
  When set to useInputSegmentation, the output segment or fragment points are set by the RAI markers from the input streams.
  - **Type:** SmoothGroupSegmentationMode (p. 118)
  - **Required:** False

- **numRetries**
  Number of retry attempts.
  - **Type:** integer
  - **Required:** False
  - **Minimum:** 0

- **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

- **eventStopBehavior**
  When set to sendEos, send EOS signal to IIS server when stopping the event
  - **Type:** SmoothGroupEventStopBehavior (p. 118)
  - **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. 119)
  - **Required:** False

- **destination**
  Smooth Streaming publish point on an IIS server. Elemental Live acts as a "Push" encoder to IIS.
  - **Type:** OutputLocationRef (p. 114)
  - **Required:** True
Properties

**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. 119)
_Required_: False

**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. 118)
_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 unless those certificates are manually added to the OS trusted keystore.

_Type_: SmoothGroupCertificateMode (p. 118)
_Required_: False

**inputLossAction**

Parameter that control output group behavior on input loss.

_Type_: InputLossActionForMsSmoothOut (p. 91)
_Required_: False

**sendDelayMs**

Outputs that are "output locked" can use this delay. Assign a delay to the output that is "secondary". Do not assign a delay to the "primary" output. The delay means that the primary output will always
reach the downstream system before the secondary, which helps ensure that the downstream system always uses the primary output. (If there were no delay, the downstream system might flip-flop between whichever output happens to arrive first.) If the primary fails, the downstream system will switch to the secondary output. When the primary is restarted, the downstream system will switch back to the primary (because once again it is always arriving first)

**Properties**

- **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. 118)
  - **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. 119)
  - **Required**: False

**MsSmoothOutputSettings**

- **nameModifier**
  String concatenated to the end of the destination filename. Required for multiple outputs of the same type.
  - **Type**: string
  - **Required**: False

**NetworkInputServerValidation (Enum)**

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.

CHECK_CRYPTOGRAPHY_AND_VALIDATE_NAME
CHECK_CRYPTOGRAPHY_ONLY

**NetworkInputSettings**

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

- **Type:** HlsInputSettings (p. 84)
- **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. 111)
- **Required:** False

**Output**

**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

**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. 114)
- **Required:** True

**audioDescriptionNames**
The names of the AudioDescriptions used as audio sources for this output.
OutputDestination

settings

Destination settings for output; one for each redundant encoder.

Type: Array of type OutputDestinationSettings (p. 113)
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

url

A URL specifying a destination

Type: string
Required: False

username

username for destination

Type: string
Required: False

OutputGroup

outputs

Type: Array of type Output (p. 112)
Required: True

outputGroupSettings

Settings associated with the output group.
Properties

Type: OutputGroupSettings (p. 114)
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

OutputGroupSettings

archiveGroupSettings

Type: ArchiveGroupSettings (p. 31)
Required: False

udpGroupSettings

Type: UdpGroupSettings (p. 121)
Required: False

msSmoothGroupSettings

Type: MsSmoothGroupSettings (p. 108)
Required: False

hlsGroupSettings

Type: HlsGroupSettings (p. 79)
Required: False

OutputLocationRef

destinationRefId

Type: string
Required: False

OutputSettings

archiveOutputSettings

Type: ArchiveOutputSettings (p. 32)
Required: False

msSmoothOutputSettings

Type: MsSmoothOutputSettings (p. 111)
Required: False
udpOutputSettings

Type: UdpOutputSettings (p. 122)
Required: False

hlsOutputSettings

Type: HlsOutputSettings (p. 87)
Required: False

PassThroughSettings

RemixSettings

channelMappings

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

Type: Array of type AudioChannelMapping (p. 32)
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

Scte20Convert608To708 (Enum)

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.

DISABLED
UPCONVERT
Scte20PlusEmbeddedDestinationSettings

Scte20SourceSettings

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. 115)
Required: False

Scte27DestinationSettings

Scte27SourceSettings

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 (Enum)

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

FOLLOW
IGNORE

Scte35AposWebDeliveryAllowedBehavior (Enum)

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

FOLLOW
IGNORE
**Scte35SpliceInsert**

**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. 117)
- **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. 117)
- **Required:** False

**Scte35SpliceInsertNoRegionalBlackoutBehavior (Enum)**
When set to ignore, Segment Descriptors with noRegionalBlackoutFlag set to 0 will no longer trigger blackouts or Ad Avail slates

- FOLLOW
- IGNORE

**Scte35SpliceInsertWebDeliveryAllowedBehavior (Enum)**
When set to ignore, Segment Descriptors with webDeliveryAllowedFlag set to 0 will no longer trigger blackouts or Ad Avail slates

- FOLLOW
- IGNORE

**Scte35TimeSignalApos**

**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. 116)
- **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. 116)
- **Required**: False

SmoothGroupAudioOnlyTimecodeControl (Enum)

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.

- PASSTHROUGH
- USE_CONFIGURED_CLOCK

SmoothGroupCertificateMode (Enum)

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 unless those certificates are manually added to the OS trusted keystore.

- SELF_SIGNED
- VERIFY_AUTHENTICITY

SmoothGroupEventIdMode (Enum)

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.

- NO_EVENT_ID
- USE_CONFIGURED
- USE_TIMESTAMP

SmoothGroupEventStopBehavior (Enum)

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

- NONE
- SEND_EOS

SmoothGroupSegmentationMode (Enum)

When set to useInputSegmentation, the output segment or fragment points are set by the RAI markers from the input streams.
Properties

USE_INPUT_SEGMENTATION
USE_SEGMENT_DURATION

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

- NONE
- SCTE_35

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

- DO_NOT_SEND
- SEND

SmoothGroupTimestampOffsetMode (Enum)
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

- USE_CONFIGURED_OFFSET
- USE_EVENT_START_DATE

SmpteTtDestinationSettings

StandardHlsSettings

m3u8Settings
- Type: M3u8Settings (p. 105)
- 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

staticKeyValue
Static key value as a 32 character hexadecimal string.
- Type: string
- Required: True
keyProviderServer
The URL of the license server used for protecting content.
- **Type**: InputLocation (p. 91)
- **Required**: False

TeletextDestinationSettings

TeletextSourceSettings

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

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. 120)
- **Required**: True

TimecodeConfigSource (Enum)
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.
- **EMBEDDED**
- **SYSTEMCLOCK**
**TtmlDestinationSettings**

**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. 121)
- **Required**: False

**TtmlDestinationStyleControl (Enum)**
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.

- PASSTHROUGH
- USE_CONFIGURED

**UdpContainerSettings**

**m2tsSettings**

- **Type**: M2tsSettings (p. 98)
- **Required**: False

**UdpGroupSettings**

**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. 91)
- **Required**: False

**timedMetadataId3Frame**
Indicates ID3 frame that has the timecode.

- **Type**: UdpTimedMetadataId3Frame (p. 122)
- **Required**: False

**timedMetadataId3Period**
Timed Metadata interval in seconds.

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

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. 114)
  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. 121)
  Required: True

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

  Type: FecOutputSettings (p. 65)
  Required: False

UdpTimedMetadataId3Frame (Enum)
Indicates ID3 frame that has the timecode.

  NONE
  PRIV
  TDRL

Validation>Error

errorMessage

  Type: string
  Required: False

elementPath

  Type: string
  Required: False
VideoCodecSettings

h264Settings

Type: H264Settings (p. 70)
Required: False

VideoDescription

respondToAfd

Indicates how to respond to the AFD values in the input stream. Setting to "respond" causes input video to be clipped, depending on AFD value, input display aspect ratio and output display aspect ratio.

Type: VideoDescriptionRespondToAfd (p. 124)
Required: False

scalingBehavior

When set to "stretchToOutput", automatically configures the output position to stretch the video to the specified output resolution. This option will override any position value.

Type: VideoDescriptionScalingBehavior (p. 124)
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). Leave out to use source video width. If left out, height must also be left out. Display aspect ratio is always preserved by letterboxing or pillarboxing when necessary.

Type: integer
Required: False

sharpness

Changes the width of the anti-alias filter kernel used for scaling. Only applies if scaling is being performed and antiAlias is set to true. 0 is the softest setting, 100 the sharpest, and 50 recommended for most content.

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

codecSettings

Video codec settings.
**VideoCodecSettings** (p. 123)  
**Required:** True

**height**
Output video height (in pixels). Leave blank to use source video height. If left blank, width must also be unspecified.

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

**VideoDescriptionRespondToAfd (Enum)**
Indicates how to respond to the AFD values in the input stream. Setting to "respond" causes input video to be clipped, depending on AFD value, input display aspect ratio and output display aspect ratio.

- NONE
- PASSTHROUGH
- RESPOND

**VideoDescriptionScalingBehavior (Enum)**
When set to "stretchToOutput", automatically configures the output position to stretch the video to the specified output resolution. This option will override any position value.

- DEFAULT
- STRETCH_TO_OUTPUT

**VideoSelector**

**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. 125)  
**Required:** False

**selectorSettings**
The video selector settings.

**Type:** VideoSelectorSettings (p. 125)  
**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.
Properties

**VideoSelectorColorSpace (Enum)**

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

- FOLLOW
- REC_601
- REC_709

**VideoSelectorColorSpaceUsage (Enum)**

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.

- FALLBACK
- FORCE

**VideoSelectorPid**

**pid**

Selects a specific PID from within a video source.

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 8191

**VideoSelectorProgramId**

**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**

**videoSelectorPid**

- Type: `VideoSelectorPid (p. 125)`
- Required: False
videoSelectorProgramId
Type: VideoSelectorProgramId (p. 125)
Required: False

WebvttDestinationSettings

Channels channelId

URI
/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. 127)</td>
<td>Channel details</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 138)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 138)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>None</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 138)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 138)</td>
<td>The channel you’re requesting to describe does not exist.</td>
</tr>
<tr>
<td>504</td>
<td>None</td>
<td>Gateway Timeout Error</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 138)</td>
<td>Request limit exceeded on list channel calls to channel service.</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. 127)</td>
<td>Deletion was successfully initiated.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 138)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 117)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>None</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 138)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFoundException (p. 138)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>504</td>
<td>None</td>
<td>Gateway Timeout Error</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 138)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 138)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
</tbody>
</table>

**Schemas**

**Response Bodies**

**Example Channel**

```json
{
    "inputAttachments (p. 160)": [
        {
            "inputId (p. 199)": "string",
            "inputSettings (p. 199)": {
                "sourceEndBehavior (p. 202)": enum,
                "audioSelectors (p. 202)": [
                    {
                        "name (p. 149)": "string",
                        "selectorSettings (p. 150)": {
                            "audioLanguageSelection (p. 150)": enum,
                            "languageSelectionPolicy (p. 147)": enum,
                            "languageCode (p. 147)": "string"
                        },
                        "audioPidSelection (p. 150)": {
```
"pid (p. 149)": integer
},
"deblockFilter (p. 202)": enum,
"inputFilter (p. 202)": enum,
"networkInputSettings (p. 202)": {
  "hlsInputSettings (p. 221)": {
    "retries (p. 194)": integer,
    "bandwidth (p. 194)": integer,
    "retryInterval (p. 194)": integer,
    "bufferSegments (p. 194)": integer
  },
  "serverValidation (p. 221)": enum
},
"videoSelector (p. 203)": {
  "colorSpace (p. 233)": enum,
  "selectorSettings (p. 233)": {
    "videoSelectorPid (p. 234)": {
      "pid (p. 234)": integer
    },
    "videoSelectorProgramId (p. 235)": {
      "programId (p. 234)": integer
    }
  },
  "colorSpaceUsage (p. 233)": enum
},
"filterStrength (p. 203)": integer,
"captionSelectors (p. 203)": [
  {
    "name (p. 159)": "string",
    "languageCode (p. 159)": "string",
    "selectorSettings (p. 159)": {
      "embeddedSourceSettings (p. 159)": {
        "scte20Detection (p. 173)": enum,
        "source608ChannelNumber (p. 173)": integer,
        "convert608To708 (p. 173)": enum,
        "source608TrackNumber (p. 173)": integer
      },
      "scte20SourceSettings (p. 159)": {
        "source608ChannelNumber (p. 225)": integer,
        "convert608To708 (p. 225)": enum
      },
      "dvbsubSourceSettings (p. 159)": {
        "pid (p. 167)": integer
      },
      "teletextSourceSettings (p. 159)": {
        "pageNumber (p. 229)": "string"
      },
      "aribSourceSettings (p. 160)": {
      },
      "scte27SourceSettings (p. 160)": {
        "pid (p. 225)": integer
      }
    }
  }
}]
"denoiseFilter (p. 203)": enum
"roleArn (p. 160)": "string",
"destinations (p. 160)": [
  {
    "settings (p. 222)": [
  128
Schemas

```json
{
  "passwordParam (p. 222)": "string",
  "url (p. 222)": "string",
  "username (p. 222)": "string"
}

"id (p. 222)": "string"

"name (p. 160)": "string",
"encoderSettings (p. 160)": {
  "timecodeConfig (p. 174)": {
    "syncThreshold (p. 229)": integer,
    "source (p. 229)": enum
  },
  "outputGroups (p. 174)": [
    {"outputs (p. 222)": [
      {"videoDescriptionName (p. 221)": "string",
       "outputName (p. 221)": "string",
       "captionDescriptionNames (p. 221)": [
         "string"
      ],
      "outputSettings (p. 221)": {
        "archiveOutputSettings (p. 223)": {
          "extension (p. 144)": "string",
          "containerSettings (p. 144)": {
            "m2tsSettings (p. 143)": {
              "audioStreamType (p. 207)": enum,
              "ecmPid (p. 207)": "string",
              "dvBTeletextPid (p. 207)": "string",
              "aribCaptionsPidControl (p. 207)": enum,
              "bitrate (p. 207)": integer,
              "segmentationTime (p. 207)": number,
              "rateMode (p. 207)": integer,
              "audioPids (p. 208)": "string",
              "ebpLookaheadMs (p. 208)": integer,
              "ebpAudioInterval (p. 208)": enum,
              "audioFramesPerPes (p. 208)": integer,
              "fragmentTime (p. 208)": number,
              "scte35Pid (p. 208)": "string",
              "programNum (p. 209)": integer,
              "pmtInterval (p. 209)": integer,
              "segmentationStyle (p. 209)": enum,
              "ebif (p. 209)": "string",
              "audioBufferModel (p. 210)": enum,
              "dvBNitSettings (p. 210)": {
                "networkName (p. 161)": "string",
                "networkId (p. 162)": integer,
                "repInterval (p. 162)": integer
              },
              "absentInputAudioBehavior (p. 210)": enum,
              "timedMetadataBehavior (p. 210)": enum,
              "timedMetadataPid (p. 210)": "string",
              " egregiousAccept (p. 210)": "string",
              "etvSignalPid (p. 210)": "string",
              "bufferModel (p. 211)": enum,
              "scte35Control (p. 211)": enum,
              "ebpPlacement (p. 211)": enum,
              "arib (p. 211)": enum,
              "dvBsdSettings (p. 211)": {
                "serviceName (p. 162)": "string",
                "serviceProviderName (p. 162)": "string",
                "repInterval (p. 162)": integer,
              }
            },
            "dvBSdtSettings (p. 211)": {
              "service (p. 162)": "string",
              "serviceProviderName (p. 162)": "string",
              "repInterval (p. 162)": integer,
            }
          }
        }
      }
    }
  }
}
```
"outputSdt (p. 163)": enum,
"nullPacketBitrate (p. 211)": number,
"pcrPid (p. 211)": "string",
"transportStreamId (p. 212)": integer,
"videoPid (p. 212)": "string",
"pcrControl (p. 212)": enum,
"esRateInPes (p. 212)": enum,
"segmentationMarkers (p. 212)": enum,
"klv (p. 212)": enum,
"dvbTdtSettings (p. 212)": {
  "repInterval (p. 167)": integer
},
"ccDescriptor (p. 213)": enum,
"patInterval (p. 213)": integer,
"etvPlatformPid (p. 213)": "string",
"dvbSubPids (p. 213)": "string",
"aribCaptionsPid (p. 213)": "string",
"scte27Pids (p. 213)": "string",
"klvDataPids (p. 214)": "string"
},
"nameModifier (p. 144)": "string"
},
"msSmoothOutputSettings (p. 223)": {
  "nameModifier (p. 220)": "string"
},
"udpOutputSettings (p. 224)": {
  "destination (p. 231)": {
    "destinationRefId (p. 223)": "string"
  },
  "bufferMsec (p. 231)": integer,
  "containerSettings (p. 231)": {
    "m2tsSettings (p. 230)": {
      "audioStreamType (p. 207)": enum,
      "ecmPid (p. 207)": "string",
      "dvbTeletextPid (p. 207)": "string",
      "aribCaptionsPidControl (p. 207)": enum,
      "bitrate (p. 207)": integer,
      "segmentationTime (p. 207)": number,
      "rateMode (p. 207)": enum,
      "audioPids (p. 208)": "string",
      "ebpLookaheadMs (p. 208)": integer,
      "ebpAudioInterval (p. 208)": integer,
      "audioFramesPerPes (p. 208)": integer,
      "fragmentTime (p. 208)": number,
      "scte35Pid (p. 208)": "string",
      "programNum (p. 209)": integer,
      "pcrPeriod (p. 209)": integer,
      "pmtInterval (p. 209)": integer,
      "segmentationStyle (p. 209)": enum,
      "ebif (p. 209)": enum,
      "audioBufferModel (p. 210)": enum,
      "dvbNitSettings (p. 210)": {
        "networkName (p. 161)": "string",
        "networkId (p. 162)": integer,
        "repInterval (p. 162)": integer
      },
      "absentInputAudioBehavior (p. 210)": enum,
      "timedMetadataBehavior (p. 210)": enum,
      "scte35Control (p. 210)": enum,
      "bufferModel (p. 211)": enum,
      "scte35Control (p. 211)": enum,
      "ebpPlacement (p. 211)": enum,
"arib (p. 211)" : enum,
"dvbsdtSettings (p. 211)" : {
  "serviceName (p. 162)" : "string",
  "serviceProviderName (p. 162)" : "string",
  "repInterval (p. 162)" : integer,
  "outputSdt (p. 163)" : enum
},
"nullPacketBitrate (p. 211)" : number,
"pcrPid (p. 211)" : "string",
"transportStreamId (p. 212)" : integer,
"videoPid (p. 212)" : "string",
"pcrControl (p. 212)" : enum,
"esRateInPes (p. 212)" : enum,
"segmentationMarkers (p. 212)" : enum,
"klv (p. 212)" : enum,
"dvbtstSettings (p. 212)" : {
  "repInterval (p. 167)" : integer
},
"ccDescriptor (p. 213)" : enum,
"patInterval (p. 213)" : integer,
"etvPlatformPid (p. 213)" : "string",
"dvbsubPids (p. 213)" : "string",
"aribCaptionsPid (p. 213)" : "string",
"scte27Pids (p. 213)" : "string",
"klvDataPids (p. 214)" : "string"
}
},
"fecOutputSettings (p. 231)" : {
  "rowLength (p. 175)" : integer,
  "columnDepth (p. 175)" : integer,
  "includeFec (p. 175)" : enum
},
"hlsOutputSettings (p. 224)" : {
  "segmentModifier (p. 196)" : "string",
  "hlsSettings (p. 197)" : {
    "standardHlsSettings (p. 197)" : {
      "m3u8Settings (p. 228)" : {
        "pmtPid (p. 214)" : "string",
        "ecmPid (p. 214)" : "string",
        "scte35Behavior (p. 214)" : enum,
        "pcrPid (p. 215)" : "string",
        "audioPids (p. 215)" : "string",
        "audioFramesPerPes (p. 215)" : integer,
        "scte35Pid (p. 215)" : "string",
        "transportStreamId (p. 215)" : integer,
        "videoPid (p. 215)" : "string",
        "pcrControl (p. 216)" : enum,
        "pcrPeriod (p. 216)" : integer,
        "programNum (p. 216)" : integer,
        "pmtInterval (p. 216)" : integer,
        "patInterval (p. 216)" : integer,
        "timedMetadataBehavior (p. 216)" : enum
      },
      "audioOnlyHlsSettings (p. 197)" : {
        "audioTrackType (p. 148)" : enum,
        "audioGroupId (p. 149)" : "string",
        "audioOnlyImage (p. 149)" : {
          "passwordParam (p. 200)" : "string",
          "uri (p. 200)" : "string",
          "username (p. 200)" : "string"
        }
      }
    }
  }
}
"nameModifier (p. 197)" : "string",
}
],
"audioDescriptionNames (p. 221)" : [
"string"
]
},
"outputGroupSettings (p. 222)" : {
"archiveGroupSettings (p. 223)" : {
"destination (p. 143)" : {
"destinationRefId (p. 223)" : "string",
"rolloverInterval (p. 144)" : integer
},
"udpGroupSettings (p. 223)" : {
"inputLossAction (p. 230)" : enum,
"timedMetadataId3Frame (p. 231)" : enum,
"timedMetadataId3Period (p. 231)" : integer
},
"msSmoothGroupSettings (p. 223)" : {
"eventId (p. 217)" : "string",
"fragmentLength (p. 217)" : integer,
"timestampOffset (p. 218)" : "string",
"segmentationMode (p. 218)" : enum,
"numRetries (p. 218)" : integer,
"acquisitionPointId (p. 218)" : "string",
"eventStopBehavior (p. 218)" : enum,
"sparseTrackType (p. 218)" : enum,
"destination (p. 218)" : {
"destinationRefId (p. 223)" : "string"
},
"timestampOffsetMode (p. 219)" : enum,
"audioOnlyTimecodeControl (p. 219)" : enum,
"connectionRetryInterval (p. 219)" : integer,
"filecacheDuration (p. 219)" : integer,
"certificateMode (p. 219)" : enum,
"inputLossAction (p. 219)" : enum,
"sendDelayMs (p. 219)" : integer,
"eventIdMode (p. 220)" : enum,
"restartDelay (p. 220)" : integer,
"streamManifestBehavior (p. 220)" : enum
},
"hlsGroupSettings (p. 223)" : {
"segmentsPerSubdirectory (p. 189)" : integer,
"ivInManifest (p. 189)" : enum,
"outputSelection (p. 189)" : enum,
"encryptionType (p. 189)" : enum,
"destination (p. 189)" : {
"destinationRefId (p. 223)" : "string"
},
"indexNSegments (p. 189)" : integer,
"timedMetadataId3Frame (p. 190)" : enum,
"constantIv (p. 190)" : "string",
"baseUriManifest (p. 190)" : "string",
"captionLanguageSetting (p. 190)" : enum,
"minSegmentLength (p. 190)" : integer,
"mode (p. 190)" : enum,
"keyProviderSettings (p. 191)" : {
"staticKeySettings (p. 204)" : {
"staticKeyValue (p. 229)" : "string",
"keyProviderServer (p. 229)" : {
"passwordParam (p. 200)" : "string",
"uri (p. 200)" : "string",
"username (p. 200)" : "string"
}
```json
}

"manifestCompression (p. 191)" : enum,
"ivSource (p. 191)" : enum,
"tsFileMode (p. 191)" : enum,
"manifestDurationFormat (p. 191)" : enum,
"keyFormatVersions (p. 191)" : "string",
"streamInfResolution (p. 191)" : enum,
"timestampDeltaMilliseconds (p. 192)" : integer,
"baseUrlContent (p. 192)" : "string",
"segmentationMode (p. 192)" : enum,
"captionLanguageMappings (p. 192)" : [
  
  "languageDescription (p. 158)" : "string",
  "captionChannel (p. 158)" : integer,
  "languageCode (p. 158)" : "string"
]

"clientCache (p. 192)" : enum,
"codecSpecification (p. 192)" : enum,
"keepSegments (p. 192)" : integer,
"timedMetadataId3Period (p. 193)" : integer,
"programDateTime (p. 193)" : enum,
"directoryStructure (p. 193)" : enum,
"keyFormat (p. 193)" : "string",
"inputLossAction (p. 193)" : enum,
"adMarkers (p. 193)" : [

  "adMarkers (p. 193)" : [
    enum
  ]

  "programDateTimePeriod (p. 193)" : integer,
  "segmentLength (p. 194)" : integer,
  "hlsCdnSettings (p. 194)" : {
    "hlsAkamaiSettings (p. 188)" : {
      "httpTransferMode (p. 186)" : enum,
      "salt (p. 186)" : "string",
      "numRetries (p. 186)" : integer,
      "restartDelay (p. 186)" : integer,
      "connectionRetryInterval (p. 186)" : integer,
      "filecacheDuration (p. 186)" : integer,
      "token (p. 187)" : "string"
    }
  }
  "hlsWebdavSettings (p. 188)" : {
    "httpTransferMode (p. 198)" : enum,
    "numRetries (p. 198)" : integer,
    "restartDelay (p. 198)" : integer,
    "connectionRetryInterval (p. 198)" : integer,
    "filecacheDuration (p. 199)" : integer
  }
  "hlsBasicPutSettings (p. 188)" : {
    "numRetries (p. 187)" : integer,
    "restartDelay (p. 187)" : integer,
    "connectionRetryInterval (p. 187)" : integer,
    "filecacheDuration (p. 187)" : integer
  }
  "hlsMediaStoreSettings (p. 188)" : {
    "mediaStoreStorageClass (p. 195)" : enum,
    "numRetries (p. 195)" : integer,
    "restartDelay (p. 196)" : integer,
    "connectionRetryInterval (p. 196)" : integer,
    "filecacheDuration (p. 196)" : integer
  }
}

"name (p. 223)" : "string"
"audioDescriptions (p. 174)": [
    {
      "languageCodeControl (p. 145)": enum,
      "audioTypeControl (p. 145)": enum,
      "remixSettings (p. 145)": {
        "channelMappings (p. 224)": [
          {
            "outputChannel (p. 144)": integer,
            "inputChannelLevels (p. 144)": [
              {
                "inputChannel (p. 199)": integer,
                "gain (p. 199)": integer
              }
            ]
          },
          "channelsOut (p. 224)": integer,
          "channelsIn (p. 224)": integer
        }
      },
      "audioType (p. 146)": enum,
      "name (p. 146)": "string",
      "codecSettings (p. 146)": {
        "ac3Settings (p. 145)": {
          "drcProfile (p. 142)": enum,
          "dialnorm (p. 142)": integer,
          "codingMode (p. 142)": enum,
          "metadataControl (p. 142)": enum,
          "bitrate (p. 142)": number,
          "lfeFilter (p. 143)": enum,
          "bitstreamMode (p. 143)": enum
        },
        "aacSettings (p. 145)": {
          "vbrQuality (p. 139)": enum,
          "codingMode (p. 139)": enum,
          "profile (p. 140)": enum,
          "inputType (p. 140)": enum,
          "bitrate (p. 140)": number,
          "rawFormat (p. 140)": enum,
          "rateControlMode (p. 140)": enum,
          "sampleRate (p. 140)": number,
          "spec (p. 140)": enum
        },
        "eac3Settings (p. 145)": {
          "dialnorm (p. 169)": integer,
          "passThroughControl (p. 170)": enum,
          "drcLine (p. 170)": enum,
          "metadataControl (p. 170)": enum,
          "bitrate (p. 170)": number,
          "ltRtSurroundMixLevel (p. 170)": number,
          "surroundExMode (p. 170)": enum,
          "lfeControl (p. 170)": enum,
          "codingMode (p. 171)": enum,
          "surroundMode (p. 171)": enum,
          "attenuationControl (p. 171)": enum,
          "lfeFilter (p. 171)": enum,
          "dcFilter (p. 171)": enum,
          "phaseControl (p. 171)": enum,
          "ltRtCenterMixLevel (p. 171)": number,
          "stereoDownmix (p. 171)": enum,
          "bitstreamMode (p. 172)": enum,
          "loRoSurroundMixLevel (p. 172)": number,
          "drcRf (p. 172)": enum,
          "loRoCenterMixLevel (p. 172)": number
        },
        "passThroughSettings (p. 145)": {
          "drcProfile (p. 142)": enum,
"mp2Settings (p. 145)": {
  "codingMode (p. 217)": enum,
  "bitrate (p. 217)": number,
  "sampleRate (p. 217)": number
}
,
"languageCode (p. 146)": "string",
"streamName (p. 146)": "string",
"audioNormalizationSettings (p. 146)": {
  "targetLkfs (p. 148)": number,
  "algorithmControl (p. 148)": enum,
  "algorithm (p. 148)": enum
},
"audioSelectorName (p. 146)": "string"
},
"captionDescriptions (p. 174)": [
  {
    "captionSelectorName (p. 156)": "string",
    "languageDescription (p. 156)": "string",
    "name (p. 157)": "string",
    "destinationSettings (p. 157)": {
      "scte27DestinationSettings (p. 157)": {
      },
      "burnInDestinationSettings (p. 157)": {
        "xPosition (p. 153)": integer,
        "backgroundColor (p. 153)": enum,
        "yPosition (p. 153)": integer,
        "teletextGridControl (p. 153)": enum,
        "backgroundColorOpacity (p. 153)": integer,
        "fontOpacity (p. 153)": integer,
        "shadowOpacity (p. 154)": integer,
        "fontResolution (p. 154)": integer,
        "shadowXOffset (p. 154)": integer,
        "outlineSize (p. 154)": integer,
        "outlineColor (p. 154)": enum,
        "fontColor (p. 155)": enum,
        "font (p. 155)": {
          "passwordParam (p. 200)": "string",
          "uri (p. 200)": "string",
          "username (p. 200)": "string"
        }
      },
      "teletextDestinationSettings (p. 157)": {
      },
      "webvttDestinationSettings (p. 157)": {
      },
      "ttmlDestinationSettings (p. 157)": {
        "styleControl (p. 230)": enum
      },
      "smpteTtDestinationSettings (p. 158)": {
      },
      "embeddedPlusScte20DestinationSettings (p. 158)": {
      },
      "dvbSubDestinationSettings (p. 158)": {
        "xPosition (p. 164)": integer,
        "backgroundColor (p. 164)": enum,
        "yPosition (p. 164)": integer,
        "teletextGridControl (p. 164)": enum,
        "backgroundColorOpacity (p. 165)": integer,
        "fontOpacity (p. 165)": integer,
"shadowOpacity (p. 165)": integer,
"fontResolution (p. 165)": integer,
"shadowYOffset (p. 165)": integer,
"outlineSize (p. 165)": integer,
"outlineColor (p. 166)": enum,
"shadowXOffset (p. 166)": integer,
"align (p. 166)": enum,
"shadowColor (p. 166)": enum,
"fontColor (p. 166)": enum,
"font (p. 167)": {
  "passwordParam (p. 200)": "string",
  "uri (p. 200)": "string",
  "username (p. 200)": "string"
}
},
"embeddedDestinationSettings (p. 158)": {
},
"aribDestinationSettings (p. 158)": {
},
"scte20PlusEmbeddedDestinationSettings (p. 158)": {
},
"languageCode (p. 157)": "string"
],
"availConfiguration (p. 174)": {
  "availSettings (p. 151)": {
    "scte35TimeSignalApos (p. 151)": {
      "adAvailOffset (p. 227)": integer,
      "webDeliveryAllowedFlag (p. 227)": enum,
      "noRegionalBlackoutFlag (p. 227)": enum
    },
    "scte35SpliceInsert (p. 151)": {
      "adAvailOffset (p. 226)": integer,
      "webDeliveryAllowedFlag (p. 226)": enum,
      "noRegionalBlackoutFlag (p. 226)": enum
    }
  }
},
"globalConfiguration (p. 174)": {
  "inputLossBehavior (p. 176)": {
    "inputLossImageType (p. 201)": enum,
    "inputLossImageColor (p. 201)": "string",
    "inputLossImageSlate (p. 201)": {
      "passwordParam (p. 200)": "string",
      "uri (p. 200)": "string",
      "username (p. 200)": "string"
    },
    "repeatFrameMsec (p. 201)": integer,
    "blackFrameMsec (p. 202)": integer
  },
  "supportLowFramerateInputs (p. 176)": enum,
  "initialAudioGain (p. 176)": integer,
  "inputEndAction (p. 176)": enum,
  "outputTimingSource (p. 176)": enum
},
"videoDescriptions (p. 174)": [
{  
  "respondToAfd (p. 232)": enum,
  "scalingBehavior (p. 232)": enum,
  "name (p. 232)": "string",
  "width (p. 232)": integer,
  "sharpness (p. 232)": integer,
  "h264Settings (p. 232)": {
    "protectionProfile (p. 232)": enum,
  }
"minIInterval (p. 180)": integer,
"slices (p. 180)": integer,
"parNumerator (p. 180)": integer,
"gopSizeUnits (p. 180)": enum,
"maxBitrate (p. 180)": integer,
"bitrate (p. 180)": integer,
"bufFillPct (p. 181)": integer,
"temporalAq (p. 181)": enum,
"sfdSignaling (p. 181)": enum,
"timecodeInsertion (p. 181)": enum,
"bufSize (p. 181)": integer,
"softness (p. 181)": integer,
"framerateControl (p. 182)": enum,
"fixedAfd (p. 182)": enum,
"level (p. 182)": enum,
"lookAheadRateControl (p. 182)": enum,
"profile (p. 182)": enum,
"framerateNumerator (p. 182)": integer,
"gopClosedCadence (p. 182)": integer,
"framerateDenominator (p. 183)": integer,
".spatialAq (p. 183)": enum,
"entropyEncoding (p. 183)": enum,
"adaptiveQuantization (p. 183)": enum,
"colorMetadata (p. 183)": enum,
"gopSize (p. 183)": number,
"numRefFrames (p. 183)": integer,
"gopSReference (p. 184)": enum,
"syntax (p. 184)": enum,
"parControl (p. 184)": enum,
"parDenominator (p. 184)": integer,
"sceneChangeDetect (p. 184)": integer,
"scanType (p. 184)": enum,
"gopNumBFrames (p. 184)": integer,
"flickerAq (p. 185)": enum,
"rateControlMode (p. 185)": enum
},
"height (p. 233)": integer
}

"availBlanking (p. 174)": {
  "state (p. 150)": enum,
  "availBlankingImage (p. 150)": {
    "passwordParam (p. 200)": "string",
    "uri (p. 200)": "string",
    "username (p. 200)": "string"
  }
},
"blackoutSlate (p. 175)": {
  "networkEndBlackoutImage (p. 151)": {
    "passwordParam (p. 200)": "string",
    "uri (p. 200)": "string",
    "username (p. 200)": "string"
  },
  "networkEndBlackout (p. 151)": enum,
  "networkId (p. 151)": "string",
  "blackoutSlateImage (p. 152)": {
    "passwordParam (p. 200)": "string",
    "uri (p. 200)": "string",
    "username (p. 200)": "string"
  },
  "state (p. 152)": enum
},
"id (p. 160)": "string",
"pipelinesRunningCount (p. 161)": integer,
Properties

AacCodingMode (Enum)

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.

AD_RECEIVER_MIX
CODING_MODE_1_0
CODING_MODE_1_1
CODING_MODE_2_0
CODING_MODE_5_1

AacInputType (Enum)

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.

  BROADCASTER_MIXED_AD
  NORMAL

AacProfile (Enum)

AAC Profile.

  HEV1
  HEV2
  LC

AacRateControlMode (Enum)

Rate Control Mode.

  CBR
  VBR

AacRawFormat (Enum)

Sets LATM / LOAS AAC output for raw containers.

  LATM_LOAS
  NONE

AacSettings

vbrQuality

VBR Quality Level - Only used if rateControlMode is VBR.

  Type: AacVbrQuality (p. 141)
  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.
**profile**

AAC Profile.

Type: `AacProfile (p. 139)`
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. 139)`
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. 139)`
Required: False

**rateControlMode**

Rate Control Mode.

Type: `AacRateControlMode (p. 139)`
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. 141)`
**Required**: False

**AacSpec (Enum)**

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

- MPEG2
- MPEG4

**AacVbrQuality (Enum)**

VBR Quality Level - Only used if rateControlMode is VBR.

- HIGH
- LOW
- MEDIUM_HIGH
- MEDIUM_LOW

**Ac3BitstreamMode (Enum)**

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

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

**Ac3CodingMode (Enum)**

Dolby Digital coding mode. Determines number of channels.

- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_3_2_LFE

**Ac3DrcProfile (Enum)**

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

- FILM_STANDARD
- NONE

**Ac3LfeFilter (Enum)**

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

Ac3MetadataControl (Enum)

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.

FOLLOW_INPUT
USE_CONFIGURED

Ac3Settings

drcProfile

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

Type: Ac3DrcProfile (p. 141)
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. 141)
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. 142)
Required: False

bitrate

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

Type: number
Properties

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. 141)
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. 141)
Required: False

AccessDenied

message

Type: string
Required: False

AfdSignaling (Enum)

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.

AUTO
FIXED
NONE

ArchiveContainerSettings

m2tsSettings

Type: M2tsSettings (p. 207)
Required: False

ArchiveGroupSettings

destination

A directory and base filename where archive files should be written. If the base filename portion of the URI is left blank, the base filename of the first input will be automatically inserted.

Type: OutputLocationRef (p. 223)
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

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. 143)
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

AribSourceSettings

AudioChannelMapping

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. 199)
Required: True
AudioCodecSettings

ac3Settings

Type: Ac3Settings (p. 142)
Required: False

aacSettings

Type: AacSettings (p. 139)
Required: False

eac3Settings

Type: Eac3Settings (p. 169)
Required: False

passThroughSettings

Type: PassThroughSettings (p. 224)
Required: False

mp2Settings

Type: Mp2Settings (p. 217)
Required: False

AudioDescription

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. 147)
Required: False

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. 147)
Required: False

remixSettings

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

Type: RemixSettings (p. 224)
**Properties**

**Required:** False

**audioType**

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

*Type: AudioType (p. 150)*  
*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. 145)*  
*Required: True*

**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 (e.g. English, or Director Commentary).

*Type: string*  
*Required: False*

**audioNormalizationSettings**

Advanced audio normalization settings.

*Type: AudioNormalizationSettings (p. 148)*  
*Required: False*

**audioSelectorName**

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

*Type: string*  
*Required: True*
**AudioDescriptionAudioTypeControl (Enum)**

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.

- FOLLOW_INPUT
- USE_CONFIGURED

**AudioDescriptionLanguageCodeControl (Enum)**

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.

- FOLLOW_INPUT
- USE_CONFIGURED

**AudioLanguageSelection**

**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. 147)
- **Required**: False

**languageCode**

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

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

**AudioLanguageSelectionPolicy (Enum)**

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.

- LOOSE
- STRICT

**AudioNormalizationAlgorithm (Enum)**

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

ITU_1770_1
ITU_1770_2

AudioNormalizationAlgorithmControl (Enum)

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.

CORRECT_AUDIO

AudioNormalizationSettings

targetLkfs

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

Type: number

Required: False

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. 148)

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. 147)

Required: False

AudioOnlyHlsSettings

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. 149)
Properties

Required: False

audioGroupId

Specifies the group to which the audio Rendition belongs.

Type: string
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. 200)
Required: False

AudioOnlyHlsTrackType (Enum)

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

ALTERNATE_AUDIO_AUTO_SELECT
ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT
ALTERNATE_AUDIO_NOT_AUTO_SELECT
AUDIO_ONLY_VARIANT_STREAM

AudioPidSelection

pid

Selects a specific PID from within a source.

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

AudioSelector

name

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

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

**selectorSettings**

The audio selector settings.

- **Type**: AudioSelectorSettings (p. 150)  
  **Required**: False

**AudioSelectorSettings**

- **audioLanguageSelection**
  - **Type**: AudioLanguageSelection (p. 147)  
    **Required**: False

- **audioPidSelection**
  - **Type**: AudioPidSelection (p. 149)  
    **Required**: False

**AudioType (Enum)**

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

- CLEAN_EFFECTS
- HEARING_IMPAIRED
- UNDEFINED
- VISUAL_IMPAIRED_COMMENTARY

**AvailBlanking**

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

  - **Type**: AvailBlankingState (p. 150)  
    **Required**: False

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

  - **Type**: InputLocation (p. 200)  
    **Required**: False

**AvailBlankingState (Enum)**

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

**AvailConfiguration**

**availSettings**

Ad avail settings.

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

**AvailSettings**

**scte35TimeSignalApos**

*Type: Scte35TimeSignalApos (p. 227)*
*Required: False*

**scte35SpliceInsert**

*Type: Scte35SpliceInsert (p. 226)*
*Required: False*

**BlackoutSlate**

**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. 200)*
*Required: False*

**networkEndBlackout**

Setting to enabled causes the encoder to black out 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. 152)*
*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*
blackoutSlateImage

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

Type: InputLocation (p. 200)
Required: False

state

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

Type: BlackoutSlateState (p. 152)
Required: False

BlackoutSlateNetworkEndBlackout (Enum)

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".

DISABLED
ENABLED

BlackoutSlateState (Enum)

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

DISABLED
ENABLED

BurnInAlignment (Enum)

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.

CENTERED
LEFT
SMART

BurnInBackgroundColor (Enum)

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

BLACK
NONE
WHITE
**BurnInDestinationSettings**

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit 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. 152)  
*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. 156)  
*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.
Properties

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

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

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. 156)
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

**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

**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. 152)  
  * **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. 156)  
  * **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. 156)  
  * **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. 200)  
  * **Required**: False
**BurnInFontColor (Enum)**

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

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

**BurnInOutlineColor (Enum)**

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

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

**BurnInShadowColor (Enum)**

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

- BLACK
- NONE
- WHITE

**BurnInTeletextGridControl (Enum)**

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.

- FIXED
- SCALED

**CaptionDescription**

**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).
Properties

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

destinationSettings

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

Type: CaptionDestinationSettings (p. 157)  
Required: False

languageCode


Type: string  
Required: False

CaptionDestinationSettings

scte27DestinationSettings

Type: Scte27DestinationSettings (p. 225)  
Required: False

burnInDestinationSettings

Type: BurnInDestinationSettings (p. 153)  
Required: False

teletextDestinationSettings

Type: TeletextDestinationSettings (p. 229)  
Required: False

webvttDestinationSettings

Type: WebvttDestinationSettings (p. 235)  
Required: False

ttmlDestinationSettings

Type: TtmlDestinationSettings (p. 230)  
Required: False
smpteTtDestinationSettings
Type: SmpteTtDestinationSettings (p. 228)
Required: False

embeddedPlusScte20DestinationSettings
Type: EmbeddedPlusScte20DestinationSettings (p. 173)
Required: False

dvbSubDestinationSettings
Type: DvbSubDestinationSettings (p. 164)
Required: False

embeddedDestinationSettings
Type: EmbeddedDestinationSettings (p. 173)
Required: False

aribDestinationSettings
Type: AribDestinationSettings (p. 144)
Required: False

scte20PlusEmbeddedDestinationSettings
Type: Scte20PlusEmbeddedDestinationSettings (p. 225)
Required: False

CaptionLanguageMapping

languageDescription
Textual description of language
Type: string
Required: False

captionChannel
Channel to insert closed captions. Each channel mapping must have a unique channel number (maximum of 4)
Type: integer
Required: False
Minimum: 1
Maximum: 4

languageCode
Three character ISO 639-2 language code (see http://www.loc.gov/standards/iso639-2)
**Type**: string  
**Required**: False

### CaptionSelector

#### 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

#### 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. 159)  
**Required**: False

### CaptionSelectorSettings

#### embeddedSourceSettings

**Type**: EmbeddedSourceSettings (p. 173)  
**Required**: False

#### scte20SourceSettings

**Type**: Scte20SourceSettings (p. 225)  
**Required**: False

#### dvbSubSourceSettings

**Type**: DvbSubSourceSettings (p. 167)  
**Required**: False

#### teletextSourceSettings

**Type**: TeletextSourceSettings (p. 229)  
**Required**: False
**Properties**

**aribSourceSettings**
- **Type:** AribSourceSettings (p. 144)
- **Required:** False

**scte27SourceSettings**
- **Type:** Scte27SourceSettings (p. 225)
- **Required:** False

### Channel

**inputAttachments**

List of input attachments for channel.

- **Type:** Array of type InputAttachment (p. 199)
- **Required:** False

**roleArn**

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

- **Type:** string
- **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. 222)
- **Required:** False

**name**

The name of the channel. (user-mutable)

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

**encoderSettings**

- **Type:** EncoderSettings (p. 174)
- **Required:** False

**id**

The unique id of the channel.

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

The number of currently healthy pipelines.

Type: integer
Required: False

state

Type: ChannelState (p. 161)
Required: False

arn

The unique arn of the channel.

Type: string
Required: False

egressEndpoints

The endpoints where outgoing connections initiate from

Type: Array of type ChannelEgressEndpoint (p. 161)
Required: False

ChannelEgressEndpoint

sourceIp

Public IP of where a channel's output comes from

Type: string
Required: False

ChannelState (Enum)

CREATING
CREATE_FAILED
IDLE
STARTING
RUNNING
RECOVERING
STOPPING
DELETING
DELETED

DvbNitSettings

networkName

The network name text placed in the networkNameDescriptor inside the Network Information Table. Maximum length is 256 characters.
Type: string
Required: True

**networkId**

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

Type: integer
Required: True
Minimum: 0
Maximum: 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 (Enum)**

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.

SDT_FOLLOW
SDT_FOLLOW_IF_PRESENT
SDT_MANUAL
SDT_NONE

**DvbSdtSettings**

**serviceName**

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

Type: string
Required: False

**serviceProviderName**

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

Type: string
Required: False

**repInterval**

The number of milliseconds between instances of this table in the output transport stream.
**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. 162)

**Required:** False

**DvbSubDestinationAlignment (Enum)**

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.

- CENTERED
- LEFT
- SMART

**DvbSubDestinationBackgroundColor (Enum)**

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

- BLACK
- NONE
- WHITE

**DvbSubDestinationFontColor (Enum)**

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

- BLACK
- BLUE
- GREEN
- RED
- WHITE
- YELLOW
DvbSubDestinationOutlineColor (Enum)

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

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

DvbSubDestinationSettings

xPosition

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit 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. 163)
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. 167)
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

**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

**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

**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. 164)*  
*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*

**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*

**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. 163)*  
*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. 167)*  
*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. 163)
**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. 200)
**Required**: False

**DvbSubDestinationShadowColor (Enum)**

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

- BLACK
- NONE
- WHITE

**DvbSubDestinationTeletextGridControl (Enum)**

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.

- FIXED
- SCALED

**DvbSubSourceSettings**

**pid**

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

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

**DvbTdtSettings**

**repInterval**

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

**Type**: integer
**Required**: False
**Minimum**: 1000
**Maximum**: 30000
**Eac3AttenuationControl (Enum)**

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

- ATTENUATE_3_DB
- NONE

**Eac3BitstreamMode (Enum)**

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

- COMMENTARY
- COMPLETE_MAIN
- EMERGENCY
- HEARING_IMPAIRED
- VISUALLY_IMPAIRED

**Eac3CodingMode (Enum)**

Dolby Digital Plus coding mode. Determines number of channels.

- CODING_MODE_1_0
- CODING_MODE_2_0
- CODING_MODE_3_2

**Eac3DcFilter (Enum)**

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

- DISABLED
- ENABLED

**Eac3DrcLine (Enum)**

Sets the Dolby dynamic range compression profile.

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

**Eac3DrcRf (Enum)**

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

- FILM_LIGHT
- FILM_STANDARD
- MUSIC_LIGHT
- MUSIC_STANDARD
**Eac3LfeControl (Enum)**

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

- LFE
- NO_LFE

**Eac3LfeFilter (Enum)**

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

- DISABLED
- ENABLED

**Eac3MetadataControl (Enum)**

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.

- FOLLOW_INPUT
- USE_CONFIGURED

**Eac3PassthroughControl (Enum)**

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.

- NO_PASSTHROUGH
- WHEN_POSSIBLE

**Eac3PhaseControl (Enum)**

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

- 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. 169)
- **Required:** False

**drcLine**

Sets the Dolby dynamic range compression profile.

- **Type:** Eac3DrcLine (p. 168)
- **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. 169)
- **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. 172)
- **Required:** False

**lfeControl**

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

- **Type:** Eac3LfeControl (p. 169)
- **Required:** False
codingMode
Dolby Digital Plus coding mode. Determines number of channels.

- **Type:** Eac3CodingMode (p. 168)
- **Required:** False

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

- **Type:** Eac3SurroundMode (p. 172)
- **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. 168)
- **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. 169)
- **Required:** False

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

- **Type:** Eac3DcFilter (p. 168)
- **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. 169)
- **Required:** False

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

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

stereoDownmix
Stereo downmix preference. Only used for 3/2 coding mode.
Type: `Eac3StereoDownmix` (p. 172)
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. 168)
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. 168)
Required: False

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

Type: number
Required: False

**Eac3StereoDownmix (Enum)**
Stereo downmix preference. Only used for 3/2 coding mode.

DPL2
LO_RO
LT_RT
NOT_INDICATED

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

DISABLED
ENABLED
NOT_INDICATED

**Eac3SurroundMode (Enum)**
When encoding 2/0 audio, sets whether Dolby Surround is matrix encoded into the two channels.
**Properties**

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

**EmbeddedConvert608To708** (Enum)

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.

- **DISABLED**
- **UPCONVERT**

**EmbeddedDestinationSettings**

**EmbeddedPlusScte20DestinationSettings**

**EmbeddedScte20Detection** (Enum)

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

- **AUTO**
- **OFF**

**EmbeddedSourceSettings**

**scte20Detection**

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

- **Type**: EmbeddedScte20Detection (p. 173)
- **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. 173)
- **Required**: False

**source608TrackNumber**

This field is unused and deprecated.
Type: integer
Required: False
Minimum: 1
Maximum: 5

**EncoderSettings**

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

Type: TimecodeConfig (p. 229)
Required: True

**outputGroups**
Type: Array of type OutputGroup (p. 222)
Required: True

**audioDescriptions**
Type: Array of type AudioDescription (p. 145)
Required: True

**captionDescriptions**
Settings for caption descriptions
Type: Array of type CaptionDescription (p. 156)
Required: False

**availConfiguration**
Event-wide configuration settings for ad avail insertion.

Type: AvailConfiguration (p. 151)
Required: False

**globalConfiguration**
Configuration settings that apply to the event as a whole.

Type: GlobalConfiguration (p. 176)
Required: False

**videoDescriptions**
Type: Array of type VideoDescription (p. 232)
Required: True

**availBlanking**
Settings for ad avail blanking.
Type: `AvailBlanking (p. 150)`
Required: False

blackoutSlate
Settings for blackout slate.

Type: `BlackoutSlate (p. 151)`
Required: False

FecOutputIncludeFec (Enum)
Enables column only or column and row based FEC

- COLUMN
- COLUMN_AND_ROW

FecOutputSettings

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. 175)`
Required: False

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

inputLossBehavior

Settings for system actions when input is lost.

Type: InputLossBehavior (p. 201)
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. 177)
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 an input completes (e.g. end-of-file.) Options include immediately switching to the next sequential input (via "switchInput"), switching to the next input and looping back to the first input when last input ends (via "switchAndLoopInputs") or not switching inputs and instead transcoding black / color / slate images per the "Input Loss Behavior" configuration until an activateInput REST command is received (via "none").

Type: GlobalConfigurationInputEndAction (p. 177)
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. 177)
**Required:** False

**GlobalConfigurationInputEndAction (Enum)**
Indicates the action to take when an input completes (e.g. end-of-file.) Options include immediately switching to the next sequential input (via "switchInput"), switching to the next input and looping back to the first input when last input ends (via "switchAndLoopInputs") or not switching inputs and instead transcoding black / color / slate images per the "Input Loss Behavior" configuration until an activateInput REST command is received (via "none").

- NONE
- SWITCH_AND_LOOP_INPUTS

**GlobalConfigurationLowFramerateInputs (Enum)**
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.

- DISABLED
- ENABLED

**GlobalConfigurationOutputTimingSource (Enum)**
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.

- INPUT_CLOCK
- SYSTEM_CLOCK

**H264AdaptiveQuantization (Enum)**
Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

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

**H264ColorMetadata (Enum)**
Includes colorspace metadata in the output.

- IGNORE
- INSERT

**H264EntropyEncoding (Enum)**
Entropy encoding mode. Use cabac (must be in Main or High profile) or cavlc.

- CABAC
CAVLC

**H264FlickerAq (Enum)**

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

- DISABLED
- ENABLED

**H264FramerateControl (Enum)**

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.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**H264GopBReference (Enum)**

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

- DISABLED
- ENABLED

**H264GopSizeUnits (Enum)**

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.

- FRAMES
- SECONDS

**H264Level (Enum)**

H.264 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 (Enum)
Amount of lookahead. A value of low can decrease latency and memory usage, while high can produce better quality for certain content.

HIGH
LOW
MEDIUM

H264ParControl (Enum)
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.

INITIALIZE_FROM_SOURCE
SPECIFIED

H264Profile (Enum)
H.264 Profile.

BASELINE
HIGH
HIGH_10BIT
HIGH_422
HIGH_422_10BIT
MAIN

H264RateControlMode (Enum)
Rate control mode.

CBR
VBR

H264ScanType (Enum)
Sets the scan type of the output to progressive or top-field-first interlaced.

INTERLACED
PROGRESSIVE

H264SceneChangeDetect (Enum)
Scene change detection. Inserts I-frames on scene changes when enabled.

DISABLED
ENABLED
H264Settings

**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. 178)  
**Required:** False

**maxBitrate**

Maximum bitrate in bits/second (for VBR mode only).

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

**bitrate**

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

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

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

### temporalAq

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

- **Type:** H264TemporalAq (p. 185)
- **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. 143)
- **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. 185)
- **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
Properties

**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. 178)
*Required*: False

**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. 175)
*Required*: False

**level**

H.264 Level.

*Type*: H264Level (p. 178)
*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. 179)
*Required*: False

**profile**

H.264 Profile.

*Type*: H264Profile (p. 179)
*Required*: False

**framerateNumerator**

Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 \(\approx\) 23.976 fps.

*Type*: integer
*Required*: False

**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

**framerateDenominator**
Framerate denominator.

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

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

**Type**: H264SpatialAq (p. 185)  
**Required**: False

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

**Type**: H264EntropyEncoding (p. 177)  
**Required**: False

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

**Type**: H264AdaptiveQuantization (p. 177)  
**Required**: False

**colorMetadata**
Includes colorspace metadata in the output.

**Type**: H264ColorMetadata (p. 177)  
**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. 178)  
- **Required**: False

**syntax**

Produces a bitstream compliant with SMPTE RP-2027.

- **Type**: H264Syntax (p. 185)  
- **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. 179)  
- **Required**: False

**parDenominator**

Pixel Aspect Ratio denominator.

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

**sceneChangeDetect**

Scene change detection. Inserts I-frames on scene changes when enabled.

- **Type**: H264SceneChangeDetect (p. 179)  
- **Required**: False

**scanType**

Sets the scan type of the output to progressive or top-field-first interlaced.

- **Type**: H264ScanType (p. 179)  
- **Required**: False

**gopNumBFrames**

Number of B-frames between reference frames.

- **Type**: integer
flickerAq
If set to enabled, adjust quantization within each frame to reduce flicker or 'pop' on I-frames.

    Type: H264FlickerAq (p. 178)
    Required: False

rateControlMode
Rate control mode.

    Type: H264RateControlMode (p. 179)
    Required: False

H264SpatialAq (Enum)
If set to enabled, adjust quantization within each frame based on spatial variation of content complexity.

    DISABLED
    ENABLED

H264Syntax (Enum)
Produces a bitstream compliant with SMPTE RP-2027.

    DEFAULT
    RP2027

H264TemporalAq (Enum)
If set to enabled, adjust quantization within each frame based on temporal variation of content complexity.

    DISABLED
    ENABLED

H264TimecodeInsertionBehavior (Enum)
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

    DISABLED
    PIC_TIMING_SEI

HlsAdMarkers (Enum)

    ADOBE
    ELEMENTAL
    ELEMENTAL_SCTE35
**HlsAkamaiHttpTransferMode (Enum)**

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

- CHUNKED
- NON_CHUNKED

**HlsAkamaiSettings**

**httpTransferMode**

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

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

**salt**

Salt for authenticated Akamai.

- **Type:** string
- **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

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

Type: string
Required: False

HlsBasicPutSettings

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

INSERT
NONE
OMIT

**HlsCdnSettings**

**hlsAkamaiSettings**

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

**hlsWebdavSettings**

*Type: HlsWebdavSettings (p. 198)*
*Required: False*

**hlsBasicPutSettings**

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

**hlsMediaStoreSettings**

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

**HlsClientCache (Enum)**

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.

DISABLED
ENABLED

**HlsCodecSpecification (Enum)**

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

RFC_4281
RFC_6381

**HlsDirectoryStructure (Enum)**

Place segments in subdirectories.

SINGLE_DIRECTORY
SUBDIRECTORY_PER_STREAM
Properties

HlsEncryptionType (Enum)
Encrypts the segments with the given encryption scheme. Exclude this parameter if no encryption is desired.

- AES128
- SAMPLE_AES

HlsGroupSettings

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. 195)
- Required: False

outputSelection
Generates the .m3u8 playlist file for this HLS output group. The segmentsOnly option will output segments without the .m3u8 file.

- Type: HlsOutputSelection (p. 196)
- Required: False

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

- Type: HlsEncryptionType (p. 189)
- Required: False

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

- Type: OutputLocationRef (p. 223)
- Required: True

indexNSegments
Number of segments to keep in the playlist (.m3u8) file. mode must be "vod" for this setting to have an effect, and this number should be less than or equal to keepSegments.
**Properties**

**Type**: integer

**Required**: False

**Minimum**: 1

**timedMetadataId3Frame**

Indicates ID3 frame that has the timecode.

**Type**: HlsTimedMetadataId3Frame (p. 198)

**Required**: False

**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

**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-CAPTIONS 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. 187)

**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 set to "vod", keeps and indexes all segments starting with the first segment. If set to "live" segments will age out and only the last keepSegments number of segments will be retained.

**Type**: HlsMode (p. 196)
Properties

Required: False

keyProviderSettings

The key provider settings.

Type: KeyProviderSettings (p. 204)
Required: False

manifestCompression

When set to gzip, compresses HLS playlist.

Type: HlsManifestCompression (p. 195)
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. 195)
Required: False

tsFileMode

When set to "singleFile", emits the program as a single media resource (.ts) file, and uses #EXT-X-BYTERANGE tags to index segment for playback. Playback of VOD mode content during event is not guaranteed due to HTTP server caching.

Type: HlsTsFileMode (p. 198)
Required: False

manifestDurationFormat

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

Type: HlsManifestDurationFormat (p. 195)
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. 197)
**Properties**

**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**
When set to useInputSegmentation, the output segment or fragment points are set by the RAI markers from the input streams.

- **Type:** HlsSegmentationMode (p. 197)
- **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. 158)
- **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. 188)
- **Required:** False

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

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

**keepSegments**
Number of segments to retain in the destination directory. mode must be "live" for this setting to have an effect.

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

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. 197)
Required: False

directoryStructure
Place segments in subdirectories.

Type: HlsDirectoryStructure (p. 188)
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. 200)
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. 185)
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. 188)  
**Required**: False

**HlsInputSettings**

**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 (Enum)**

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.

- EXCLUDE
- INCLUDE

**HlsIVSource (Enum)**

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.

- EXPLICIT
- FOLLOWS_SEGMENT_NUMBER

**HlsManifestCompression (Enum)**

When set to gzip, compresses HLS playlist.

- GZIP
- NONE

**HlsManifestDurationFormat (Enum)**

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

- FLOATING_POINT
- INTEGER

**HlsMediaStoreSettings**

**mediaStoreStorageClass**

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

- Type: HlsMediaStoreStorageClass (p. 196)
- 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 (Enum)**

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

  - TEMPORAL

**HlsMode (Enum)**

If set to "vod", keeps and indexes all segments starting with the first segment. If set to "live" segments will age out and only the last keepSegments number of segments will be retained.

  - LIVE
  - VOD

**HlsOutputSelection (Enum)**

Generates the .m3u8 playlist file for this HLS output group. The segmentsOnly option will output segments without the .m3u8 file.

  - MANIFESTS_AND_SEGMENTS
  - SEGMENTS_ONLY

**HlsOutputSettings**

**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. 197)  
Required: True

nameModifier

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

Type: string  
Required: True

HlsProgramDateTime (Enum)

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

EXCLUDE
INCLUDE

HlsSegmentationMode (Enum)

When set to useInputSegmentation, the output segment or fragment points are set by the RAI markers from the input streams.

USE_INPUT_SEGMENTATION
USE_SEGMENT_DURATION

HlsSettings

standardHlsSettings

Type: StandardHlsSettings (p. 228)  
Required: False

audioOnlyHlsSettings

Type: AudioOnlyHlsSettings (p. 148)  
Required: False

HlsStreamInfResolution (Enum)

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

EXCLUDE
INCLUDE
**HlsTimedMetadataId3Frame (Enum)**

Indicates ID3 frame that has the timecode.

- NONE
- PRIV
- TDRL

**HlsTsFileMode (Enum)**

When set to "singleFile", emits the program as a single media resource (.ts) file, and uses #EXT-X-BYTERANGE tags to index segment for playback. Playback of VOD mode content during event is not guaranteed due to HTTP server caching.

- SEGMENTED_FILES
- SINGLE_FILE

**HlsWebdavHttpTransferMode (Enum)**

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

- CHUNKED
- NON_CHUNKED

**HlsWebdavSettings**

**httpTransferMode**

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

- **Type**: HlsWebdavHttpTransferMode (p. 198)
- **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.
**filecacheDuration**
Size in seconds of file cache for streaming outputs.

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

**InputAttachment**

**inputId**
The ID of the input

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

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

- **Type:** InputSettings (p. 202)
- **Required:** False

**InputChannelLevel**

**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

**InputDeblockFilter (Enum)**
Enable or disable the deblock filter when filtering.
DISABLED
ENABLED

**InputDenoiseFilter (Enum)**
Enable or disable the denoise filter when filtering.

DISABLED
ENABLED

**InputFilter (Enum)**
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

AUTO
DISABLED
FORCED

**InputLocation**

**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 rtmpEndpoint 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 (Enum)**
Parameter that control output group behavior on input loss.

EMIT_OUTPUT
PAUSE_OUTPUT
### InputLossActionForMsSmoothOut (Enum)

Parameter that control output group behavior on input loss.

- EMIT_OUTPUT
- PAUSE_OUTPUT

### InputLossActionForUdpOut (Enum)

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.

- DROP_PROGRAM
- DROP_TS
- EMIT_PROGRAM

### InputLossBehavior

#### inputLossImageType

Indicates whether to substitute a solid color or a slate into the output after input loss exceeds blackFrameMsec.

- **Type**: InputLossImageType (p. 202)
- **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

#### inputLossImageSlate

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

- **Type**: InputLocation (p. 200)
- **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 (Enum)**

Indicates whether to substitute a solid color or a slate into the output after input loss exceeds blackFrameMsec.

- COLOR
- SLATE

**InputSettings**

**sourceEndBehavior**

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

- **Type**: InputSourceEndBehavior (p. 203)
- **Required**: False

**audioSelectors**

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

- **Type**: Array of type AudioSelector (p. 149)
- **Required**: False

**deblockFilter**

Enable or disable the deblock filter when filtering.

- **Type**: InputDeblockFilter (p. 199)
- **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. 200)
- **Required**: False

**networkInputSettings**

Input settings.
**Type**: NetworkInputSettings (p. 221)
**Required**: True

**videoSelector**

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

**Type**: VideoSelector (p. 233)
**Required**: False

**filterStrength**

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

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

**captionSelectors**

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

**Type**: Array of type CaptionSelector (p. 159)
**Required**: False

**denoiseFilter**

Enable or disable the denoise filter when filtering.

**Type**: InputDenoiseFilter (p. 200)
**Required**: False

**InputSourceEndBehavior (Enum)**

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

CONTINUE
LOOP

**InternalServerError**

**message**

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

**InvalidRequest**

**message**

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

staticKeySettings

Type: StaticKeySettings (p. 229)
Required: False

LimitExceeded

message

Type: string
Required: False

M2tsAbsentInputAudioBehavior (Enum)

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.

DROP
ENCODE_SILENCE

M2tsArib (Enum)

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

DISABLED
ENABLED

M2tsAribCaptionsPidControl (Enum)

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.

AUTO
USE_CONFIGURED

M2tsAudioBufferModel (Enum)

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

ATSC
DVB

M2tsAudioInterval (Enum)

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.
**AWS Elemental MediaLive API Reference**

**Properties**

**VIDEO_AND_FIXED_INTERVALS**

**VIDEO_INTERVAL**

---

**M2tsAudioStreamType (Enum)**

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

- ATSC
- DVB

---

**M2tsBufferModel (Enum)**

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.

- MULTIPLEX
- NONE

---

**M2tsCcDescriptor (Enum)**

When set to enabled, generates captionServiceDescriptor in PMT.

- DISABLED
- ENABLED

---

**M2tsEbifControl (Enum)**

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

- NONE
- PASSTHROUGH

---

**M2tsEbpPlacement (Enum)**

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.

- VIDEO_AND_AUDIO_PIDS
- VIDEO_PID

---

**M2tsEsRateInPes (Enum)**

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

- EXCLUDE
- INCLUDE

---

**M2tsKlv (Enum)**

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

- NONE
- PASSTHROUGH
**M2tsPcrControl (Enum)**

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.

- `CONFIGURED_PCR_PERIOD`
- `PCR_EVERY_PES_PACKET`

**M2tsRateMode (Enum)**

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.

- `CBR`
- `VBR`

**M2tsScte35Control (Enum)**

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

- `NONE`
- `PASSTHROUGH`

**M2tsSegmentationMarkers (Enum)**

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.

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

**M2tsSegmentationStyle (Enum)**

The segmentation style parameter controls how segmentation markers are inserted into the transport stream. With avails, it is possible that segments may be truncated, which can influence where future segmentation markers are inserted. When a segmentation style of "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.

- `MAINTAIN_CADENCE`
- `RESET_CADENCE`
M2tsSettings

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. 205)
  Required: False

ecmPid
Packet Identifier (PID) for ECM in the transport stream. Only enabled when Simulcrypt is enabled. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20).8182 (or 0x1ff6).

  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. 204)
  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

segmentationTime
The length in seconds of each segment. Required unless markers is set to None_.

  Type: number
  Required: False
  Minimum: 1.0

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**

**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)..8192 (or 0x1ff6).

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

**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. 204)
- **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

**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)..8192 (or 0x1ff6).
**Properties**

**programNum**

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

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

**pcrPeriod**

Maximum time in milliseconds between Program Clock Reference (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. Valid values are 0, 10..1000.

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

**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. 206)
- **Required**: False

**ebif**

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

- **Type**: M2tsEbifControl (p. 205)
- **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. 204)
  Required: False

dvbNitSettings

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

  Type: DvbNitSettings (p. 161)
  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. 204)
  Required: False

timedMetadataBehavior

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

  Type: M2tsTimedMetadataBehavior (p. 214)
  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. 205)
Required: False

scte35Control

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

Type: M2tsScte35Control (p. 206)
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. 205)
Required: False

arib

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

Type: M2tsArib (p. 204)
Required: False

dvbSdtSettings

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

Type: DvbSdtSettings (p. 162)
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

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

**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

**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. 206)
- **Required:** False

**esRateInPes**

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

- **Type:** M2tsEsRateInPes (p. 205)
- **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. 206)
- **Required:** False

**klv**

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

- **Type:** M2tsKlv (p. 205)
- **Required:** False

**dvbTdtSettings**

Inserts DVB Time and Date Table (TDT) at the specified table repetition interval.
**Type:** DvbTdtSettings (p. 167)
**Required:** False

**ccDescriptor**
When set to enabled, generates captionServiceDescriptor in PMT.

**Type:** M2tsCcDescriptor (p. 205)
**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 (Enum)**

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

- NO_PASSTHROUGH
- PASSTHROUGH

**M3u8PcrControl (Enum)**

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.

- CONFIGURED_PCR_PERIOD
- PCR_EVERY_PES_PACKET

**M3u8Scte35Behavior (Enum)**

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

- NO_PASSTHROUGH
- PASSTHROUGH

**M3u8Settings**

**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**

ThePlatform-protected transport streams using ‘microsoft’ as Target Client include an ECM stream. This ECM stream contains the size, IV, and PTS of every sample in the transport stream. This stream PID is specified here. This PID has no effect on non ThePlatform-protected streams.

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

**scte35Behavior**

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

Type: M3u8Scte35Behavior (p. 214)
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

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
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. 214)
*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

**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. A value of "0" writes out the PMT once per segment file.

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

**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

**timedMetadataBehavior**

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

*Type*: M3u8TimedMetadataBehavior (p. 217)
*Required*: False
M3u8TimedMetadataBehavior (Enum)
When set to passthrough, timed metadata is passed through from input to output.

- NO_PASSTHROUGH
- PASSTHROUGH

Mp2CodingMode (Enum)
The MPEG2 Audio coding mode. Valid values are codingMode10 (for mono) or codingMode20 (for stereo).

- CODING_MODE_1_0
- CODING_MODE_2_0

Mp2Settings
codingMode
The MPEG2 Audio coding mode. Valid values are codingMode10 (for mono) or codingMode20 (for stereo).

  Type: Mp2CodingMode (p. 217)
  Required: False

bitrate
Average bitrate in bits/second.

  Type: number
  Required: False

sampleRate
Sample rate in Hz.

  Type: number
  Required: False

MsSmoothGroupSettings
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

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

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

  Type: string
  Required: False

segmentationMode
When set to useInputSegmentation, the output segment or fragment points are set by the RAI markers from the input streams.

  Type: SmoothGroupSegmentationMode (p. 228)
  Required: False

numRetries
Number of retry attempts.

  Type: integer
  Required: False
  Minimum: 0

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

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

  Type: SmoothGroupEventStopBehavior (p. 228)
  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. 228)
  Required: False

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

  Type: OutputLocationRef (p. 223)
  Required: True
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. 228)
Required: False

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. 227)
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 unless those certificates are manually added to the OS trusted keystore.

Type: SmoothGroupCertificateMode (p. 227)
Required: False

inputLossAction

Parameter that control output group behavior on input loss.

Type: InputLossActionForMsSmoothOut (p. 201)
Required: False

sendDelayMs

Outputs that are "output locked" can use this delay. Assign a delay to the output that is "secondary". Do not assign a delay to the "primary" output. The delay means that the primary output will always
reach the downstream system before the secondary, which helps ensure that the downstream system always uses the primary output. (If there were no delay, the downstream system might flip-flop between whichever output happens to arrive first.) If the primary fails, the downstream system will switch to the secondary output. When the primary is restarted, the downstream system will switch back to the primary (because once again it is always arriving first)

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. 227)
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. 228)
Required: False

MsSmoothOutputSettings

nameModifier

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

Type: string
Required: False

NetworkInputServerValidation (Enum)

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.

CHECK_CRYPTOGRAPHY_AND_VALIDATE_NAME
CHECK_CRYPTOGRAPHY_ONLY

**NetworkInputSettings**

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

*Type: HlsInputSettings (p. 194)*
*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. 220)*
*Required: False*

**Output**

**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*

**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. 223)*
*Required: True*

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

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

**OutputDestination**

**settings**

Destination settings for output; one for each redundant encoder.

**Type**: Array of type `OutputDestinationSettings` (p. 222)  
**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

**url**

A URL specifying a destination

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

**username**

username for destination

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

**OutputGroup**

**outputs**

**Type**: Array of type `Output` (p. 221)  
**Required**: True

**outputGroupSettings**

Settings associated with the output group.
**OutputGroupSettings**

*archiveGroupSettings*

- **Type:** ArchiveGroupSettings (p. 143)
- **Required:** False

*udpGroupSettings*

- **Type:** UdpGroupSettings (p. 230)
- **Required:** False

*msSmoothGroupSettings*

- **Type:** MsSmoothGroupSettings (p. 217)
- **Required:** False

*hlsGroupSettings*

- **Type:** HlsGroupSettings (p. 189)
- **Required:** False

**OutputLocationRef**

*destinationRefId*

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

**OutputSettings**

*archiveOutputSettings*

- **Type:** ArchiveOutputSettings (p. 144)
- **Required:** False

*msSmoothOutputSettings*

- **Type:** MsSmoothOutputSettings (p. 220)
- **Required:** False

**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
Properties

udpOutputSettings
  Type: UdpOutputSettings (p. 231)
  Required: False

hlsOutputSettings
  Type: HlsOutputSettings (p. 196)
  Required: False

PassThroughSettings

RemixSettings

channelMappings
Mapping of input channels to output channels, with appropriate gain adjustments.
  Type: Array of type AudioChannelMapping (p. 144)
  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

**Scte20Convert608To708 (Enum)**

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.

DISABLED
UPCONVERT

**Scte20PlusEmbeddedDestinationSettings**

**Scte20SourceSettings**

**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. 225)
Required: False

**Scte27DestinationSettings**

**Scte27SourceSettings**

**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 (Enum)**

When set to ignore, Segment Descriptors with noRegionalBlackoutFlag set to 0 will no longer trigger blackout or Ad Avail slates
Scte35AposWebDeliveryAllowedBehavior (Enum)

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

Scte35SpliceInsert

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. 226)
- **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. 226)
- **Required**: False

Scte35SpliceInsertNoRegionalBlackoutBehavior (Enum)

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

- FOLLOW
- IGNORE

Scte35SpliceInsertWebDeliveryAllowedBehavior (Enum)

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

- FOLLOW
- IGNORE
Scte35TimeSignalApos

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. 226)
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. 225)
Required: False

SmoothGroupAudioOnlyTimecodeControl (Enum)

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.

PASSTHROUGH
USE_CONFIGURED_CLOCK

SmoothGroupCertificateMode (Enum)

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 unless those certificates are manually added to the OS trusted keystore.

SELF_SIGNED
VERIFY_AUTHENTICITY

SmoothGroupEventIdMode (Enum)

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.

NO_EVENT_ID
USE_CONFIGURED
**USE_TIMESTAMP**

**SmoothGroupEventStopBehavior (Enum)**
When set to sendEos, send EOS signal to IIS server when stopping the event

- NONE
- SEND_EOS

**SmoothGroupSegmentationMode (Enum)**
When set to useInputSegmentation, the output segment or fragment points are set by the RAI markers from the input streams.

- USE_INPUT_SEGMENTATION
- USE_SEGMENT_DURATION

**SmoothGroupSparseTrackType (Enum)**
If set to scte35, use incoming SCTE-35 messages to generate a sparse track in this group of MS-Smooth outputs.

- NONE
- SCTE_35

**SmoothGroupStreamManifestBehavior (Enum)**
When set to send, send stream manifest so publishing point doesn’t start until all streams start.

- DO_NOT_SEND
- SEND

**SmoothGroupTimestampOffsetMode (Enum)**
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

- USE_CONFIGURED_OFFSET
- USE_EVENT_START_DATE

**SmpteTtDestinationSettings**

**StandardHlsSettings**

**m3u8Settings**

- **Type:** M3u8Settings (p. 214)
- **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 "."
Properties

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

**StaticKeySettings**

**staticKeyValue**
Static key value as a 32 character hexadecimal string.

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

**keyProviderServer**
The URL of the license server used for protecting content.

**Type**: `InputLocation` (p. 200)
**Required**: False

**TeletextDestinationSettings**

**TeletextSourceSettings**

**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**

**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.
**Properties**

**Type**: TimecodeConfigSource (p. 230)
**Required**: True

### TimecodeConfigSource (Enum)

Identifies the source for the timecode that will be associated with the events outputs.
- Embedded (embedded): Initialize the output timecode with timecode from 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.

- EMBEDDED
- SYSTEMCLOCK
- ZEROBASED

### TtmlDestinationSettings

**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. 230)
**Required**: False

### TtmlDestinationStyleControl (Enum)

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.

- PASSTHROUGH
- USE_CONFIGURED

### UdpContainerSettings

**m2tsSettings**

**Type**: M2tsSettings (p. 207)
**Required**: False

### UdpGroupSettings

**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. 201)
**Required**: False
timedMetadataId3Frame
Indicates ID3 frame that has the timecode.

Type: UdpTimedMetadataId3Frame (p. 231)
Required: False

timedMetadataId3Period
Timed Metadata interval in seconds.

Type: integer
Required: False
Minimum: 0

UdpOutputSettings

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. 223)
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. 230)
Required: True

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

Type: FecOutputSettings (p. 175)
Required: False

UdpTimedMetadataId3Frame (Enum)
Indicates ID3 frame that has the timecode.

NONE
PRIV
TDRL
**VideoCodecSettings**

**h264Settings**
- **Type:** H264Settings (p. 180)
- **Required:** False

**VideoDescription**

**respondToAfd**
Indicates how to respond to the AFD values in the input stream. Setting to "respond" causes input video to be clipped, depending on AFD value, input display aspect ratio and output display aspect ratio.

- **Type:** VideoDescriptionRespondToAfd (p. 233)
- **Required:** False

**scalingBehavior**
When set to "stretchToOutput", automatically configures the output position to stretch the video to the specified output resolution. This option will override any position value.

- **Type:** VideoDescriptionScalingBehavior (p. 233)
- **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). Leave out to use source video width. If left out, height must also be left out. Display aspect ratio is always preserved by letterboxing or pillarboxing when necessary.

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

**sharpness**
Changes the width of the anti-alias filter kernel used for scaling. Only applies if scaling is being performed and antiAlias is set to true. 0 is the softest setting, 100 the sharpest, and 50 recommended for most content.

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

**codecSettings**
Video codec settings.
Type: `VideoCodecSettings` (p. 232)
Required: True

**height**

Output video height (in pixels). Leave blank to use source video height. If left blank, width must also be unspecified.

Type: integer
Required: False

**VideoDescriptionRespondToAfd** (Enum)

Indicates how to respond to the AFD values in the input stream. Setting to "respond" causes input video to be clipped, depending on AFD value, input display aspect ratio and output display aspect ratio.

- NONE
- PASSTHROUGH
- RESPOND

**VideoDescriptionScalingBehavior** (Enum)

When set to "stretchToOutput", automatically configures the output position to stretch the video to the specified output resolution. This option will override any position value.

- DEFAULT
- STRETCH_TO_OUTPUT

**VideoSelector**

**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. 234)
Required: False

**selectorSettings**

The video selector settings.

Type: `VideoSelectorSettings` (p. 234)
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.
**VideoSelectorColorSpace (Enum)**

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

- FOLLOW
- REC_601
- REC_709

**VideoSelectorColorSpaceUsage (Enum)**

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.

- FALLBACK
- FORCE

**VideoSelectorPid**

pid

Selects a specific PID from within a video source.

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 8191

**VideoSelectorProgramId**

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**

videoSelectorPid

- Type: VideoSelectorPid (p. 234)
- Required: False
videoSelectorProgramId

Type: VideoSelectorProgramId (p. 234)
Required: False

WebvttDestinationSettings

Channels channelId Start

URI

/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. 236)</td>
<td>Successfully initiated start of the channel.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 246)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 247)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>None</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 246)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 246)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>504</td>
<td>None</td>
<td>Gateway Timeout Error</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 247)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 246)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
</tbody>
</table>
Schemas

Response Bodies

Example Channel

```json
{
  "inputAttachments (p. 268)": [
    {
      "inputId (p. 307)": "string",
      "inputSettings (p. 308)": {
        "sourceEndBehavior (p. 311)": enum,
        "audioSelectors (p. 311)": [
          {
            "name (p. 258)": "string",
            "selectorSettings (p. 258)": {
              "audioLanguageSelection (p. 258)": {
                "languageSelectionPolicy (p. 255)": enum,
                "languageCode (p. 256)": "string"
              },
              "audioPidSelection (p. 258)": {
                "pid (p. 258)": integer
              }
            }
          }
        ],
        "deblockFilter (p. 311)": enum,
        "inputFilter (p. 311)": enum,
        "networkInputSettings (p. 311)": {
          "hlsInputSettings (p. 329)": {
            "retries (p. 302)": integer,
            "bandwidth (p. 303)": integer,
            "retryInterval (p. 303)": integer,
            "bufferSegments (p. 303)": integer
          },
          "serverValidation (p. 330)": enum
        },
        "videoSelector (p. 311)": {
          "colorSpace (p. 342)": enum,
          "selectorSettings (p. 342)": {
            "videoSelectorPid (p. 343)": {
              "pid (p. 343)": integer
            },
            "videoSelectorProgramId (p. 343)": {
              "programId (p. 343)": integer
            }
          },
          "colorSpaceUsage (p. 342)": enum
        },
        "filterStrength (p. 311)": integer,
        "captionSelectors (p. 312)": [
          {
            "name (p. 267)": "string",
            "languageCode (p. 267)": "string",
            "selectorSettings (p. 267)": {
              "embeddedSourceSettings (p. 268)": {
                "scte20Detection (p. 281)": enum,
                "source608ChannelNumber (p. 282)": integer,
                "convert608To708 (p. 282)": enum,
                "source608TrackNumber (p. 282)": integer
              },
              "scte20SourceSettings (p. 268)": {
                "source608ChannelNumber (p. 334)": integer,
              }
            }
          }
        ]
    }
  ]
}
```
"convert608To708 (p. 334)": enum
},
"dvbSubSourceSettings (p. 268)": {
"pid (p. 276)": integer
},
"teletextSourceSettings (p. 268)": {
"pageNumber (p. 338)": "string"
},
"aribSourceSettings (p. 268)": {
},
"scte27SourceSettings (p. 268)": {
"pid (p. 334)": integer
}
},
"denoiseFilter (p. 312)": enum
},
"roleArn (p. 268)": "string",
"destinations (p. 268)": [
{ "settings (p. 330)": [
{ "passwordParam (p. 331)": "string",
"url (p. 331)": "string",
"username (p. 331)": "string"
}
],
"id (p. 331)": "string"
},
"name (p. 269)": "string",
"encoderSettings (p. 269)": {
"timecodeConfig (p. 282)": {
"syncThreshold (p. 338)": integer,
"source (p. 338)": enum
},
"outputGroups (p. 282)": [
{ "outputs (p. 331)": [
{ "videoDescriptionName (p. 330)": "string",
"outputName (p. 330)": "string",
"captionDescriptionNames (p. 330)": [ "string"
]
},
"archiveOutputSettings (p. 330)": {
"extension (p. 252)": "string",
"containerSettings (p. 252)": {
"m2tsSettings (p. 252)": {
"audioStreamType (p. 315)": enum,
"ecmPid (p. 315)": "string",
"dvbTeletextPid (p. 316)": "string",
"aribCaptionsPidControl (p. 316)": enum,
"bitrate (p. 316)": integer,
"segmentationTime (p. 316)": number,
"rateMode (p. 316)": enum,
"audioPids (p. 316)": "string",
"ebpLookaheadMs (p. 316)": integer,
"ebpAudioInterval (p. 317)": enum,
"audioFramesPerPes (p. 317)": integer,
"fragmentTime (p. 317)": number,
"scte35Pid (p. 317)": "string",
}]}]}},

237
"programNum (p. 317)" : integer,
"pcrPeriod (p. 317)" : integer,
"pmtInterval (p. 318)" : integer,
"segmentationStyle (p. 318)" : enum,
"ebif (p. 318)" : enum,
"audioBufferModel (p. 318)" : enum,
"dvbNitSettings (p. 318)" : {
  "networkName (p. 270)" : "string",
  "networkId (p. 270)" : integer,
  "repInterval (p. 270)" : integer
},
"absentInputAudioBehavior (p. 319)" : enum,
"timedMetadataBehavior (p. 319)" : enum,
"timedMetadataPid (p. 319)" : "string",
"pmtPid (p. 319)" : "string",
"etvSignalPid (p. 319)" : "string",
"bufferModel (p. 319)" : enum,
"scte35Control (p. 319)" : enum,
"ebpPlacement (p. 320)" : enum,
"arib (p. 320)" : enum,
"dvbSdtSettings (p. 320)" : {
  "serviceName (p. 271)" : "string",
  "serviceProviderName (p. 271)" : "string",
  "repInterval (p. 271)" : integer,
  "outputSdt (p. 271)" : enum
},
"nullPacketBitrate (p. 320)" : number,
"pcrPid (p. 320)" : "string",
"transportStreamId (p. 320)" : integer,
"videoPid (p. 320)" : "string",
"pcrControl (p. 321)" : enum,
"esRateInPes (p. 321)" : enum,
"segmentationMarkers (p. 321)" : enum,
"klv (p. 321)" : enum,
"dvbTdtSettings (p. 321)" : {
  "repInterval (p. 276)" : integer
},
"ccDescriptor (p. 321)" : enum,
"patInterval (p. 321)" : integer,
"etvPlatformPid (p. 322)" : "string",
"dvbSubPids (p. 322)" : "string",
"aribCaptionsPidControl (p. 316)" : enum,
"bitrate (p. 316)" : integer,
"segmentationTime (p. 316)" : number,
"rateMode (p. 316)" : enum,
"audioPids (p. 316)" : "string"
"ebpLookaheadMs (p. 316)": integer,
"ebpAudioInterval (p. 317)": enum,
"audioFramesPerPitch (p. 317)": integer,
"fragmentTime (p. 317)": number,
"scte35Pid (p. 317)": "string",
"programNum (p. 317)": integer,
"pcrPeriod (p. 317)": integer,
"pmtInterval (p. 318)": integer,
"segmentationStyle (p. 318)": enum,
"ebif (p. 318)": enum,
"audioBufferModel (p. 318)": enum,
"dvbNITSettings (p. 318)": {
  "networkName (p. 270)": "string",
  "networkId (p. 270)": integer,
  "repInterval (p. 270)": integer
},
"absentInputAudioBehavior (p. 319)": enum,
"timedMetadataBehavior (p. 319)": enum,
"timedMetadataPid (p. 319)": "string",
"pmtPid (p. 319)": "string",
"etvSignalPid (p. 319)": "string",
"bufferModel (p. 319)": enum,
"scte35Control (p. 319)": enum,
"ebpPlacement (p. 320)": enum,
"arib (p. 320)": enum,
"dvbSdtSettings (p. 320)": {
  "serviceName (p. 271)": "string",
  "serviceProviderName (p. 271)": "string",
  "repInterval (p. 271)": integer,
  "outputSdt (p. 271)": enum
},
"nullPacketBitrate (p. 320)": number,
"pcrPid (p. 320)": "string",
"transportStreamId (p. 320)": integer,
"videoPid (p. 320)": "string",
"pcrControl (p. 321)": enum,
"esRateInPes (p. 321)": enum,
"segmentationMarkers (p. 321)": enum,
"klv (p. 321)": enum,
"dvbTdtSettings (p. 321)": {
  "repInterval (p. 276)": integer
},
"ccDescriptor (p. 321)": enum,
"patInterval (p. 321)": integer,
"etvPlatformPid (p. 322)": "string",
"dvbSubPids (p. 322)": "string",
"aribCaptionsPid (p. 322)": "string",
"scte27Pids (p. 322)": "string",
"klvDataPids (p. 322)": "string"
},
"fecOutputSettings (p. 340)": {
  "rowLength (p. 283)": integer,
  "columnDepth (p. 284)": integer,
  "includeFec (p. 284)": enum
},
"hlsOutputSettings (p. 332)": {
  "segmentModifier (p. 305)": "string",
  "hlsSettings (p. 305)": {
    "standardHlsSettings (p. 306)": {
      "m3u8Settings (p. 337)": {
        "pmtPid (p. 323)": "string",
        "ecmPid (p. 323)": "string",
        "scte35Behavior (p. 323)": enum,
        "pcrPid (p. 323)": "string",
      }
    }
  }
}
"audioPids (p. 324)": "string",
"audioFramesPerPES (p. 324)": integer,
"scte35Pid (p. 324)": "string",
"transportStreamId (p. 324)": integer,
"videoPid (p. 324)": "string",
"pcrControl (p. 324)": enum,
"pcrPeriod (p. 324)": integer,
"programNum (p. 325)": integer,
"pmtInterval (p. 325)": integer,
"patInterval (p. 325)": integer,
"timedMetadataBehavior (p. 325)": enum,
"audioRenditionSets (p. 337)": "string",
"audioOnlyHlsSettings (p. 306)": {
  "audioTrackType (p. 257)": enum,
  "audioGroupId (p. 257)": "string",
  "audioOnlyImage (p. 257)": {
    "passwordParam (p. 309)": "string",
    "uri (p. 309)": "string",
    "username (p. 309)": "string"
  }
},
"nameModifier (p. 305)": "string"
},
"audioDescriptionNames (p. 330)": ["string"]
],
"outputGroupSettings (p. 331)": {
  "archiveGroupSettings (p. 332)": {
    "destination (p. 252)": {
      "destinationRefId (p. 332)": "string"
    },
    "rolloverInterval (p. 252)": integer
  },
  "udpGroupSettings (p. 332)": {
    "inputLossAction (p. 339)": enum,
    "timedMetadataId3Frame (p. 339)": enum,
    "timedMetadataId3Period (p. 339)": integer
  },
  "msSmoothGroupSettings (p. 332)": {
    "eventId (p. 326)": "string",
    "fragmentLength (p. 326)": integer,
    "timestampOffset (p. 326)": "string",
    "segmentationMode (p. 327)": enum,
    "numRetries (p. 327)": integer,
    "acquisitionPointId (p. 327)": "string",
    "eventStopBehavior (p. 327)": enum,
    "sparseTrackType (p. 327)": enum,
    "destination (p. 327)": {
      "destinationRefId (p. 332)": "string"
    },
    "timestampOffsetMode (p. 327)": enum,
    "audioOnlyTimecodeControl (p. 328)": enum,
    "connectionRetryInterval (p. 328)": integer,
    "filecacheDuration (p. 328)": integer,
    "certificateMode (p. 328)": enum,
    "inputLossAction (p. 328)": enum,
    "sendDelayMs (p. 328)": integer,
    "eventIdMode (p. 329)": enum,
    "restartDelay (p. 329)": integer,
    "streamManifestBehavior (p. 329)": enum
"hlsGroupSettings (p. 332)": {
  "segmentsPerSubdirectory (p. 297)": integer,
  "ivarInManifest (p. 297)": enum,
  "outputSelection (p. 298)": enum,
  "encryptionType (p. 298)": enum,
  "destination (p. 298)": {
    "destinationRefId (p. 332)": "string"
  },
  "indexNSegments (p. 298)": integer,
  "timedMetadataId3Frame (p. 298)": enum,
  "constantIv (p. 298)": "string",
  "baseUrlManifest (p. 298)": "string",
  "captionLanguageSetting (p. 299)": enum,
  "minSegmentLength (p. 299)": integer,
  "mode (p. 299)": enum,
  "keyProviderSettings (p. 299)": {
    "staticKeySettings (p. 312)": {
      "staticKeyValue (p. 337)": "string",
      "keyProviderServer (p. 338)": {
        "passwordParam (p. 309)": "string",
        "uri (p. 309)": "string",
        "username (p. 309)": "string"
      }
    }
  },
  "manifestCompression (p. 299)": enum,
  "ivSource (p. 299)": enum,
  "tsFileMode (p. 300)": enum,
  "manifestDurationFormat (p. 300)": enum,
  "streamInfResolution (p. 300)": enum,
  "tsFileMode (p. 300)": enum,
  "streamInfResolution (p. 300)": enum,
  "timestampDeltaMilliseconds (p. 300)": integer,
  "baseUrlContent (p. 300)": "string",
  "segmentationMode (p. 300)": enum,
  "captionLanguageMappings (p. 301)": [
    {
      "languageDescription (p. 267)": "string",
      "captionChannel (p. 267)": integer,
      "languageCode (p. 267)": "string"
    }
  ],
  "clientCache (p. 301)": enum,
  "codecSpecification (p. 301)": enum,
  "inputLossAction (p. 302)": enum,
  "adMarkers (p. 302)": enum,
  "programDateTime (p. 301)": enum,
  "directoryStructure (p. 301)": enum,
  "keyFormat (p. 302)": "string",
  "input lossAction (p. 302)": enum,
  "segmentLength (p. 302)": integer,
  "hlsCdnSettings (p. 302)": {
    "hlsAkamaiSettings (p. 296)": {
      "httpTransferMode (p. 294)": enum,
      "salt (p. 294)": "string",
      "numRetries (p. 294)": integer,
      "restartDelay (p. 295)": integer,
      "connectionRetryInterval (p. 295)": integer,
      "fileCacheDuration (p. 295)": integer,
      "token (p. 295)": "string"
    }
  }
}

"httpTransferMode (p. 307)": enum,
"numRetries (p. 307)": integer,
"restartDelay (p. 307)": integer,
"connectionRetryInterval (p. 307)": integer,
"filecacheDuration (p. 307)": integer},
"hlsBasicPutSettings (p. 296)": {
"numRetries (p. 295)": integer,
"restartDelay (p. 295)": integer,
"connectionRetryInterval (p. 296)": integer,
"filecacheDuration (p. 296)": integer},
"hlsMediaStoreSettings (p. 297)": {
"mediaStoreStorageClass (p. 304)": enum,
"numRetries (p. 304)": integer,
"restartDelay (p. 304)": integer,
"connectionRetryInterval (p. 304)": integer,
"filecacheDuration (p. 304)": integer
}
"name (p. 331)": "string"
],
"audioDescriptions (p. 282)": [
{
"languageCodeControl (p. 254)": enum,
"audioTypeControl (p. 254)": enum,
"remixSettings (p. 254)": {
"channelMappings (p. 333)": [
{ "outputChannel (p. 253)": integer,
"inputChannelLevels (p. 253)": [
{ "inputChannel (p. 308)": integer,
"gain (p. 308)": integer
}
]
"channelsOut (p. 333)": integer,
"channelsIn (p. 333)": integer
},
"audioType (p. 254)": enum,
"name (p. 254)": "string",
"codecSettings (p. 254)": {
"ac3Settings (p. 253)": {
"drcProfile (p. 250)": enum,
"dialnorm (p. 250)": integer,
"codingMode (p. 251)": enum,
"metadataControl (p. 251)": enum,
"bitrate (p. 251)": number,
"lfeFilter (p. 251)": enum,
"bitstreamMode (p. 251)": enum
},
"aacSettings (p. 253)": {
"vbrQuality (p. 248)": enum,
"codingMode (p. 248)": enum,
"profile (p. 248)": enum,
"inputType (p. 248)": enum,
"bitrate (p. 248)": number,
"rawFormat (p. 248)": enum,
"rateControlMode (p. 249)": enum,
"sampleRate (p. 249)": number,
"spec (p. 249)": enum
},
"name (p. 331)": "string"}]}
"eac3Settings (p. 253)": {
  "dialnorm (p. 278)": integer,
  "passthroughControl (p. 278)": enum,
  "drcLine (p. 278)": enum,
  "metadataControl (p. 278)": enum,
  "bitrate (p. 278)": number,
  "ltRtSurroundMixLevel (p. 279)": number,
"surroundExMode (p. 279)": enum,
  "lfeControl (p. 279)": enum,
  "codingMode (p. 279)": enum,
  "surroundMode (p. 279)": enum,
  "attenuationControl (p. 279)": enum,
  "lfeFilter (p. 279)": enum,
  "dcFilter (p. 279)": enum,
  "phaseControl (p. 280)": enum,
  "ltRtCenterMixLevel (p. 280)": number,
  "stereoDownmix (p. 280)": enum,
  "bitstreamMode (p. 280)": enum,
  "loRoSurroundMixLevel (p. 280)": number,
  "drcRf (p. 280)": enum,
  "loRoCenterMixLevel (p. 280)": number
},
"passThroughSettings (p. 253)": {
},
"mp2Settings (p. 253)": {
  "codingMode (p. 326)": enum,
  "bitrate (p. 326)": number,
  "sampleRate (p. 326)": number
},
"languageCode (p. 254)": "string",
"streamName (p. 255)": "string",
"audioNormalizationSettings (p. 255)": {
  "targetLkfs (p. 256)": number,
  "algorithmControl (p. 256)": enum,
  "algorithm (p. 257)": enum
},
"audioSelectorName (p. 255)": "string"
],
"captionDescriptions (p. 282)": [
  {
    "captionSelectorName (p. 265)": "string",
    "languageDescription (p. 265)": "string",
    "name (p. 265)": "string",
    "destinationSettings (p. 265)": {
      "scte27DestinationSettings (p. 265)": {
      },
      "burnInDestinationSettings (p. 266)": {
        "xPosition (p. 261)": integer,
        "backgroundColor (p. 261)": enum,
        "yPosition (p. 261)": integer,
        "teletextGridControl (p. 262)": enum,
        "backgroundColorOpacity (p. 262)": integer,
        "fontOpacity (p. 262)": integer,
        "shadowOpacity (p. 262)": integer,
        "fontResolution (p. 263)": integer,
        "shadowYOffset (p. 263)": integer,
        "outlineSize (p. 263)": integer,
        "outlineColor (p. 263)": enum,
        "fontSize (p. 263)": "string",
        "shadowXOffset (p. 263)": integer,
        "alignment (p. 263)": enum,
        "shadowColor (p. 263)": enum,
        "fontColor (p. 264)": enum,
      }
    }
  }
]
"font (p. 264)": {
  "passwordParam (p. 309)": "string",
  "uri (p. 309)": "string",
  "username (p. 309)": "string"
},
"teletextDestinationSettings (p. 266)": {
},
"webvttDestinationSettings (p. 266)": {
},
"ttmlDestinationSettings (p. 266)": {
  "styleControl (p. 339)": enum
},
"smpteTtDestinationSettings (p. 266)": {
},
"embeddedPlusScte20DestinationSettings (p. 266)": {
},
"dvbSubDestinationSettings (p. 266)": {
  "xPosition (p. 272)": integer,
  "backgroundColor (p. 272)": enum,
  "yPosition (p. 273)": integer,
  "teletextGridControl (p. 273)": enum,
  "backgroundColorOpacity (p. 273)": integer,
  "fontOpacity (p. 273)": integer,
  "shadowOpacity (p. 273)": integer,
  "fontResolution (p. 274)": integer,
  "shadowXOffset (p. 274)": integer,
  "outlineSize (p. 274)": integer,
  "outlineColor (p. 274)": enum,
  "fontSize (p. 274)": "string",
  "shadowYOffset (p. 274)": integer,
  "outlineColor (p. 274)": enum,
  "font (p. 275)": {
    "passwordParam (p. 309)": "string",
    "uri (p. 309)": "string",
    "username (p. 309)": "string"
  }
},
"embeddedDestinationSettings (p. 266)": {
},
"aribDestinationSettings (p. 266)": {
},
"scte20PlusEmbeddedDestinationSettings (p. 266)": {
},
"languageCode (p. 265)": "string"
},
"availConfiguration (p. 283)": {
  "availSettings (p. 259)": {
    "scte35TimeSignalApos (p. 259)": {
      "advAvailOffset (p. 335)": integer,
      "webDeliveryAllowedFlag (p. 336)": enum,
      "noRegionalBlackoutFlag (p. 336)": enum
    },
    "scte35SpliceInsert (p. 259)": {
      "advAvailOffset (p. 335)": integer,
      "webDeliveryAllowedFlag (p. 335)": enum,
      "noRegionalBlackoutFlag (p. 335)": enum
    }
  },
  "globalConfiguration (p. 283)": {
    "inputLossBehavior (p. 284)": {
      }
"inputLossImageType (p. 310)": enum,
"inputLossImageColor (p. 310)": "string",
"inputLossImageSlate (p. 310)": {
  "passwordParam (p. 309)": "string",
  "uri (p. 309)": "string",
  "username (p. 309)": "string"
},
"repeatFrameMsec (p. 310)": integer,
"blackFrameMsec (p. 310)": integer
},
"supportLowFramerateInputs (p. 284)": enum,
"initialAudioGain (p. 285)": integer,
"inputEndAction (p. 285)": enum,
"outputTimingSource (p. 285)": enum
},
"videoDescriptions (p. 283)": [
{
  "respondToAfd (p. 340)": enum,
  "scalingBehavior (p. 341)": enum,
  "name (p. 341)": "string",
  "width (p. 341)": integer,
  "sharpness (p. 341)": integer,
  "codecSettings (p. 341)": {
    "h264Settings (p. 340)": {
      "minIInterval (p. 288)": integer,
      "slices (p. 288)": integer,
      "parNumerator (p. 288)": integer,
      "gopSizeUnits (p. 289)": enum,
      "maxBitrate (p. 289)": integer,
      "bitrate (p. 289)": integer,
      "bufFillPct (p. 289)": integer,
      "temporalAq (p. 289)": enum,
      "afdSignaling (p. 289)": enum,
      "timecodeInsertion (p. 290)": enum,
      "bufSize (p. 290)": integer,
      "softness (p. 290)": integer,
      "framerateControl (p. 290)": enum,
      "fixedAfd (p. 290)": enum,
      "level (p. 290)": enum,
      "lookAheadRateControl (p. 290)": enum,
      "profile (p. 291)": enum,
      "framerateNumerator (p. 291)": integer,
      "gopClosedCadence (p. 291)": integer,
      "framerateDenominator (p. 291)": integer,
      "spatialAq (p. 291)": enum,
      "entropyEncoding (p. 291)": enum,
      "adaptiveQuantization (p. 291)": enum,
      "colorMetadata (p. 292)": enum,
      "gopSize (p. 292)": number,
      "numRefFrames (p. 292)": integer,
      "gopBReference (p. 292)": enum,
      "syntax (p. 292)": enum,
      "parControl (p. 292)": enum,
      "parDenominator (p. 292)": integer,
      "sceneChangeDetect (p. 293)": enum,
      "scanType (p. 293)": enum,
      "gopNumBFrames (p. 293)": integer,
      "flickerAq (p. 293)": enum,
      "flickerAq (p. 293)": enum,
      "rateControlMode (p. 293)": enum
    }
  }
},
"height (p. 341)": integer
]
},
"availBlanking (p. 283)": {
  "state (p. 259)": enum,
"availBlankingImage (p. 259)": {
  "passwordParam (p. 309)": "string",
  "uri (p. 309)": "string",
  "username (p. 309)": "string"
},
"blackoutSlate (p. 283)": {
  "networkEndBlackoutImage (p. 259)": {
    "passwordParam (p. 309)": "string",
    "uri (p. 309)": "string",
    "username (p. 309)": "string"
  },
  "networkEndBlackout (p. 260)": enum,
  "networkId (p. 260)": "string",
  "blackoutSlateImage (p. 260)": {
    "passwordParam (p. 309)": "string",
    "uri (p. 309)": "string",
    "username (p. 309)": "string"
  },
  "state (p. 260)": enum
},
"id (p. 269)": "string",
"pipelinesRunningCount (p. 269)": integer,
"state (p. 269)": enum,
"arn (p. 269)": "string",
"egressEndpoints (p. 269)": [
  {
    "sourceIp (p. 269)": "string"
  }
]

Example InvalidRequest

```
{
  "message (p. 312)": "string"
}
```

Example AccessDenied

```
{
  "message (p. 251)": "string"
}
```

Example ResourceNotFound

```
{
  "message (p. 333)": "string"
}
```

Example ResourceConflict

```
{
  "message (p. 333)": "string"
}
```
Example LimitExceeded

```json
{
    "message (p. 312)": "string"
}
```

Example InternalServiceError

```json
{
    "message (p. 312)": "string"
}
```

Properties

**AacCodingMode (Enum)**

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.

- AD_RECEIVER_MIX
- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_5_1

**AacInputType (Enum)**

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.

- BROADCASTER_MIXED_AD
- NORMAL

**AacProfile (Enum)**

AAC Profile.

- HEV1
- HEV2
- LC

**AacRateControlMode (Enum)**

Rate Control Mode.

- CBR
- VBR
AacRawFormat (Enum)

Sets LATM / LOAS AAC output for raw containers.

- LATM_LOAS
- NONE

AacSettings

vbrQuality

VBR Quality Level - Only used if rateControlMode is VBR.

Type: AacVbrQuality (p. 249)

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. 247)

Required: False

profile

AAC Profile.

Type: AacProfile (p. 247)

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. 247)

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.
Properties

Type: **AacRawFormat** (p. 248)  
Required: False

**rateControlMode**

Rate Control Mode.

Type: **AacRateControlMode** (p. 247)  
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. 249)  
Required: False

**AacSpec (Enum)**

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

- MPEG2
- MPEG4

**AacVbrQuality (Enum)**

VBR Quality Level - Only used if rateControlMode is VBR.

- HIGH
- LOW
- MEDIUM_HIGH
- MEDIUM_LOW

**Ac3BitstreamMode (Enum)**

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

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

**Ac3CodingMode (Enum)**

Dolby Digital coding mode. Determines number of channels.

- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_3_2_LFE

**Ac3DrcProfile (Enum)**

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

- FILM_STANDARD
- NONE

**Ac3LfeFilter (Enum)**

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

- DISABLED
- ENABLED

**Ac3MetadataControl (Enum)**

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.

- FOLLOW_INPUT
- USE_CONFIGURED

**Ac3Settings**

**drcProfile**

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

- Type: Ac3DrcProfile (p. 250)
- 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. 250)
  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. 250)
  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. 250)
  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. 249)
  Required: False

AccessDenied

message

  Type: string
  Required: False

AfdSignaling (Enum)
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.
ArchiveContainerSettings

m2tsSettings

Type: M2tsSettings (p. 315)
Required: False

ArchiveGroupSettings

destination

A directory and base filename where archive files should be written. If the base filename portion of the URI is left blank, the base filename of the first input will be automatically inserted.

Type: OutputLocationRef (p. 332)
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

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. 252)
Required: True

nameModifier

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

Type: string
Properties

Required: False

AribDestinationSettings

AribSourceSettings

AudioChannelMapping

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. 308)
  Required: True

AudioCodecSettings

ac3Settings

  Type: Ac3Settings (p. 250)
  Required: False

aacSettings

  Type: AacSettings (p. 248)
  Required: False

eac3Settings

  Type: Eac3Settings (p. 278)
  Required: False

passThroughSettings

  Type: PassThroughSettings (p. 333)
  Required: False

mp2Settings

  Type: Mp2Settings (p. 326)
  Required: False
AudioDescription

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. 255)
  Required: False

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. 255)
  Required: False

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

  Type: RemixSettings (p. 333)
  Required: False

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

  Type: AudioType (p. 258)
  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. 253)
  Required: True

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.
Properties

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. 256)
Required: False

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

Type: string
Required: True

AudioDescriptionAudioTypeControl (Enum)
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.

FOLLOW_INPUT
USE_CONFIGURED

AudioDescriptionLanguageCodeControl (Enum)
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.

FOLLOW_INPUT
USE_CONFIGURED

AudioLanguageSelection

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. 256)
**languageCode**

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

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

**AudioLanguageSelectionPolicy (Enum)**

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.

- LOOSE
- STRICT

**AudioNormalizationAlgorithm (Enum)**

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

- ITU_1770_1
- ITU_1770_2

**AudioNormalizationAlgorithmControl (Enum)**

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.

- CORRECT_AUDIO

**AudioNormalizationSettings**

**targetLkfs**

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

- **Type**: number
- **Required**: False
- **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. 256)
- **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. 256)
- **Required**: False

AudioOnlyHlsSettings

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 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. 257)
- **Required**: False

audioGroupId

Specifies the group to which the audio Rendition belongs.

- **Type**: string
- **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. 309)
- **Required**: False

AudioOnlyHlsTrackType (Enum)

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

- **ALTERNATE_AUDIO_AUTO_SELECT**
Properties

ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT
ALTERNATE_AUDIO_NOT_AUTO_SELECT
AUDIO_ONLY VARIANT STREAM

AudioPidSelection

pid

Selects a specific PID from within a source.

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

AudioSelector

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

selectorSettings

The audio selector settings.

Type: AudioSelectorSettings (p. 258)
Required: False

AudioSelectorSettings

audioLanguageSelection

Type: AudioLanguageSelection (p. 255)
Required: False

audioPidSelection

Type: AudioPidSelection (p. 258)
Required: False

AudioType (Enum)

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

CLEAN_EFFECTS
HEARING_IMPAIRED
UNDEFINED
VISUAL_IMPAIRED_COMMENTARY

AvailBlanking

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

  Type: AvailBlankingState (p. 259)
  Required: False

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

  Type: InputLocation (p. 309)
  Required: False

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

  DISABLED
  ENABLED

AvailConfiguration

availSettings
Ad avail settings.

  Type: AvailSettings (p. 259)
  Required: False

AvailSettings

scte35TimeSignalApos

  Type: Scte35TimeSignalApos (p. 335)
  Required: False

scte35SpliceInsert

  Type: Scte35SpliceInsert (p. 335)
  Required: False

BlackoutSlate

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. 309)`
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. 260)`
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

**blackoutSlateImage**

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

Type: `InputLocation (p. 309)`
Required: False

**state**

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

Type: `BlackoutSlateState (p. 260)`
Required: False

**BlackoutSlateNetworkEndBlackout (Enum)**

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".

DISABLED
ENABLED

**BlackoutSlateState (Enum)**

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

DISABLED
ENABLED

BurnInAlignment (Enum)

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.

- CENTERED
- LEFT
- SMART

BurnInBackgroundColor (Enum)

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

- BLACK
- NONE
- WHITE

BurnInDestinationSettings

xPosition

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit 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. 261)
  - 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. 265)
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

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

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

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
**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. 264)
- **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

**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

**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. 261)
- **Required:** False

**shadowColor**

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

**Type**: BurnInShadowColor (p. 264)  
**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. 264)  
**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. 309)  
**Required**: False

**BurnInFontColor (Enum)**

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

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

**BurnInOutlineColor (Enum)**

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

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

**BurnInShadowColor (Enum)**

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

- BLACK  
- NONE
**BurnInTeletextGridControl (Enum)**

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.

- FIXED
- SCALED

## CaptionDescription

**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

**destinationSettings**

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

- **Type:** CaptionDestinationSettings (p. 265)
- **Required:** False

**languageCode**


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

## CaptionDestinationSettings

**scte27DestinationSettings**

- **Type:** Scte27DestinationSettings (p. 334)
Properties

**Required**: False

**burnInDestinationSettings**
- **Type**: BurnInDestinationSettings (p. 261)
- **Required**: False

**teletextDestinationSettings**
- **Type**: TeletextDestinationSettings (p. 338)
- **Required**: False

**webvttDestinationSettings**
- **Type**: WebvttDestinationSettings (p. 344)
- **Required**: False

**ttmlDestinationSettings**
- **Type**: TtmlDestinationSettings (p. 339)
- **Required**: False

**smpteTtDestinationSettings**
- **Type**: SmpteTtDestinationSettings (p. 337)
- **Required**: False

**embeddedPlusScte20DestinationSettings**
- **Type**: EmbeddedPlusScte20DestinationSettings (p. 281)
- **Required**: False

**dvbSubDestinationSettings**
- **Type**: DvbSubDestinationSettings (p. 272)
- **Required**: False

**embeddedDestinationSettings**
- **Type**: EmbeddedDestinationSettings (p. 281)
- **Required**: False

**aribDestinationSettings**
- **Type**: AribDestinationSettings (p. 253)
- **Required**: False

**scte20PlusEmbeddedDestinationSettings**
- **Type**: Scte20PlusEmbeddedDestinationSettings (p. 334)
- **Required**: False
CaptionLanguageMapping

languageDescription
Textual description of language
  Type: string
  Required: False

captionChannel
Channel to insert closed captions. Each channel mapping must have a unique channel number (maximum of 4)
  Type: integer
  Required: False
  Minimum: 1
  Maximum: 4

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

CaptionSelector

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

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. 268)
  Required: False
CaptionSelectorSettings

embeddedSourceSettings
Type: EmbeddedSourceSettings (p. 281)
Required: False

scte20SourceSettings
Type: Scte20SourceSettings (p. 334)
Required: False

dvbSubSourceSettings
Type: DvbSubSourceSettings (p. 276)
Required: False

teletextSourceSettings
Type: TeletextSourceSettings (p. 338)
Required: False

aribSourceSettings
Type: AribSourceSettings (p. 253)
Required: False

scte27SourceSettings
Type: Scte27SourceSettings (p. 334)
Required: False

Channel

inputAttachments
List of input attachments for channel.
Type: Array of type InputAttachment (p. 307)
Required: False

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

**name**
The name of the channel. (user-mutable)

*Type:* string
*Required:* False

**encoderSettings**

*Type:* EncoderSettings (p. 282)
*Required:* False

**id**
The unique id of the channel.

*Type:* string
*Required:* False

**pipelinesRunningCount**
The number of currently healthy pipelines.

*Type:* integer
*Required:* False

**state**

*Type:* ChannelState (p. 270)
*Required:* False

**arn**
The unique arn of the channel.

*Type:* string
*Required:* False

**egressEndpoints**
The endpoints where outgoing connections initiate from

*Type:* Array of type ChannelEgressEndpoint (p. 269)
*Required:* False

### ChannelEgressEndpoint

**sourceIp**
Public IP of where a channel's output comes from
Properties

Type: string
Required: False

ChannelState (Enum)

CREATING
CREATE_FAILED
IDLE
STARTING
RUNNING
RECOVERING
STOPPING
DELETING
DELETED

DvbNitSettings

networkName

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

Type: string
Required: True

networkId

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

Type: integer
Required: True
Minimum: 0
Maximum: 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 (Enum)

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.

SDT_FOLLOW
SDT_FOLLOW_IF_PRESENT
Properties

**DvbSdtSettings**

**serviceName**

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

*Type:* string  
*Required:* False

**serviceProviderName**

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

*Type:* string  
*Required:* False

**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. 270)  
*Required:* False

**DvbSubDestinationAlignment (Enum)**

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.

- CENTERED
- LEFT
- SMART
**DvbSubDestinationBackgroundColor (Enum)**

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

- BLACK
- NONE
- WHITE

**DvbSubDestinationFontColor (Enum)**

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

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

**DvbSubDestinationOutlineColor (Enum)**

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

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

**DvbSubDestinationSettings**

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit 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.
**Properties**

**Type:** DvbSubDestinationBackgroundColor (p. 272)  
**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. 275)  
**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

**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
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

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. 272)
- **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

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
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. 271)
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. 275)
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. 272)
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. 309)
Required: False

DvbSubDestinationShadowColor (Enum)

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

BLACK
NONE
WHITE

DvbSubDestinationTeletextGridControl (Enum)

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.

FIXED
SCALLED

**DvbSubSourceSettings**

**pid**

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

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

**DvbTdtSettings**

**repInterval**

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

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

**Eac3AttenuationControl (Enum)**

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

ATTENUATE_3_DB
NONE

**Eac3BitstreamMode (Enum)**

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

COMMENTARY
COMPLETE_MAIN
EMERGENCY
HEARING_IMPAIRED
VISUALLY_IMPAIRED

**Eac3CodingMode (Enum)**

Dolby Digital Plus coding mode. Determines number of channels.

CODING_MODE_1_0
CODING_MODE_2_0
CODING_MODE_3_2

**Eac3DcFilter (Enum)**

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

Eac3DrcLine (Enum)
Sets the Dolby dynamic range compression profile.

FILM_LIGHT
FILM_STANDARD
MUSIC_LIGHT
MUSIC_STANDARD
NONE
SPEECH

Eac3DrcRf (Enum)
Sets the profile for heavy Dolby dynamic range compression, ensures that the instantaneous signal peaks do not exceed specified levels.

FILM_LIGHT
FILM_STANDARD
MUSIC_LIGHT
MUSIC_STANDARD
NONE
SPEECH

Eac3LfeControl (Enum)
When encoding 3/2 audio, setting to lfe enables the LFE channel

LFE
NO_LFE

Eac3LfeFilter (Enum)
When set to enabled, applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with codingMode32 coding mode.

DISABLED
ENABLED

Eac3MetadataControl (Enum)
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.

FOLLOW_INPUT
USE_CONFIGURED

Eac3PassthroughControl (Enum)
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.
NO_PASSTHROUGH
WHEN_POSSIBLE

**Eac3PhaseControl (Enum)**

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

- 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. 277)
- **Required:** False

**drcLine**

Sets the Dolby dynamic range compression profile.

- **Type:** Eac3DrcLine (p. 277)
- **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. 277)
- **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. 281)
- **Required:** False

**lfeControl**

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

- **Type:** Eac3LfeControl (p. 277)
- **Required:** False

**codingMode**

Dolby Digital Plus coding mode. Determines number of channels.

- **Type:** Eac3CodingMode (p. 276)
- **Required:** False

**surroundMode**

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

- **Type:** Eac3SurroundMode (p. 281)
- **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. 276)
- **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. 277)
- **Required:** False

**dcFilter**

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

- **Type**: Eac3DcFilter (p. 276)
  - **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. 278)
  - **Required**: False

**ltRtCenterMixLevel**

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

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

**stereoDownmix**

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

- **Type**: Eac3StereoDownmix (p. 281)
  - **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. 276)
  - **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. 277)
  - **Required**: False

**loRoCenterMixLevel**

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

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

Eac3StereoDownmix (Enum)

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

- DPL2
- LO_RO
- LT_RT
- NOT_INDICATED

Eac3SurroundExMode (Enum)

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

- DISABLED
- ENABLED
- NOT_INDICATED

Eac3SurroundMode (Enum)

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

- DISABLED
- ENABLED
- NOT_INDICATED

EmbeddedConvert608To708 (Enum)

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.

- DISABLED
- UPCONVERT

EmbeddedDestinationSettings

EmbeddedPlusScte20DestinationSettings

EmbeddedScte20Detection (Enum)

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

- AUTO
- OFF

EmbeddedSourceSettings

scte20Detection

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

**Type**: EmbeddedScte20Detection (p. 281)
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. 281)
Required: False

source608TrackNumber

This field is unused and deprecated.

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

EncoderSettings

timecodeConfig

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

Type: TimecodeConfig (p. 338)
Required: True

outputGroups

Type: Array of type OutputGroup (p. 331)
Required: True

audioDescriptions

Type: Array of type AudioDescription (p. 254)
Required: True

captionDescriptions

Settings for caption descriptions

Type: Array of type CaptionDescription (p. 265)
Required: False

**availConfiguration**

Event-wide configuration settings for ad avail insertion.

- **Type:** AvailConfiguration (p. 259)
- **Required:** False

**globalConfiguration**

Configuration settings that apply to the event as a whole.

- **Type:** GlobalConfiguration (p. 284)
- **Required:** False

**videoDescriptions**

- **Type:** Array of type VideoDescription (p. 340)
- **Required:** True

**availBlanking**

Settings for ad avail blanking.

- **Type:** AvailBlanking (p. 259)
- **Required:** False

**blackoutSlate**

Settings for blackout slate.

- **Type:** BlackoutSlate (p. 259)
- **Required:** False

**FecOutputIncludeFec (Enum)**

Enables column only or column and row based FEC

- COLUMN
- COLUMN_AND_ROW

**FecOutputSettings**

**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
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. 283)
- **Required**: False

FixedAfd (Enum)

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

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

GlobalConfiguration

inputLossBehavior

Settings for system actions when input is lost.

- **Type**: InputLossBehavior (p. 310)
- **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. 285)
- **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 an input completes (e.g. end-of-file.) Options include immediately switching to the next sequential input (via "switchInput"), switching to the next input and looping back to the first input when last input ends (via "switchAndLoopInputs") or not switching inputs and instead transcoding black / color / slate images per the "Input Loss Behavior" configuration until an activateInput REST command is received (via "none").

- **Type**: GlobalConfigurationInputEndAction (p. 285)
- **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. 285)
- **Required**: False

GlobalConfigurationInputEndAction (Enum)

Indicates the action to take when an input completes (e.g. end-of-file.) Options include immediately switching to the next sequential input (via "switchInput"), switching to the next input and looping back to the first input when last input ends (via "switchAndLoopInputs") or not switching inputs and instead transcoding black / color / slate images per the "Input Loss Behavior" configuration until an activateInput REST command is received (via "none").

- **NONE**
- **SWITCH_AND_LOOP_INPUTS**

GlobalConfigurationLowFramerateInputs (Enum)

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.

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

GlobalConfigurationOutputTimingSource (Enum)

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.

- **INPUT_CLOCK**
SYSTEM_CLOCK

H264AdaptiveQuantization (Enum)
Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

  HIGH
  HIGHER
  LOW
  MAX
  MEDIUM
  OFF

H264ColorMetadata (Enum)
Includes colorspace metadata in the output.

  IGNORE
  INSERT

H264EntropyEncoding (Enum)
Entropy encoding mode. Use cabac (must be in Main or High profile) or cavlc.

  CABAC
  CAVLC

H264FlickerAq (Enum)
If set to enabled, adjust quantization within each frame to reduce flicker or 'pop' on I-frames.

  DISABLED
  ENABLED

H264FramerateControl (Enum)
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.

  INITIALIZE_FROM_SOURCE
  SPECIFIED

H264GopBReference (Enum)
If enabled, use reference B frames for GOP structures that have B frames > 1.

  DISABLED
  ENABLED

H264GopSizeUnits (Enum)
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.
FRAMES
SECONDS

**H264Level (Enum)**

H.264 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 (Enum)**

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

HIGH
LOW
MEDIUM

**H264ParControl (Enum)**

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.

INITIALIZE_FROM_SOURCE
SPECIFIED

**H264Profile (Enum)**

H.264 Profile.

BASELINE
HIGH
HIGH_10BIT
HIGH_422
HIGH_422_10BIT
H264RateControlMode (Enum)
Rate control mode.
- CBR
- VBR

H264ScanType (Enum)
Sets the scan type of the output to progressive or top-field-first interlaced.
- INTERLACED
- PROGRESSIVE

H264SceneChangeDetect (Enum)
Scene change detection. Inserts I-frames on scene changes when enabled.
- DISABLED
- ENABLED

H264Settings

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.
Properties

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. 286)
Required: False

maxBitrate

Maximum bitrate in bits/second (for VBR mode only).

Type: integer
Required: False
Minimum: 1000

bitrate

Average bitrate in bits/second. Required for VBR, CBR, and ABR. For MS Smooth outputs, bitrates must be unique when 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. 294)
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. 251)
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. 294)
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. 286)
Required: False

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. 284)
Required: False

level

H.264 Level.

Type: H264Level (p. 287)
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.
Properties

profile
H.264 Profile.

Type: H264Profile (p. 287)
Required: False

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

Type: integer
Required: False

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

framerateDenominator
Framerate denominator.

Type: integer
Required: False

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

Type: H264SpatialAq (p. 293)
Required: False

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

Type: H264EntropyEncoding (p. 286)
Required: False

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

Type: H264AdaptiveQuantization (p. 286)


**Required**: False

**colorMetadata**
Includes colorspace metadata in the output.

  **Type**: H264ColorMetadata (p. 286)
  **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. 286)
  **Required**: False

**syntax**
Produces a bitstream compliant with SMPTE RP-2027.

  **Type**: H264Syntax (p. 293)
  **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. 287)
  **Required**: False

**parDenominator**
Pixel Aspect Ratio denominator.
sceneChangeDetect

Scene change detection. Inserts I-frames on scene changes when enabled.

Type: H264SceneChangeDetect (p. 288)
Required: False

scanType

Sets the scan type of the output to progressive or top-field-first interlaced.

Type: H264ScanType (p. 288)
Required: False

gopNumBFrames

Number of B-frames between reference frames.

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

flickerAq

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

Type: H264FlickerAq (p. 286)
Required: False

rateControlMode

Rate control mode.

Type: H264RateControlMode (p. 288)
Required: False

H264SpatialAq (Enum)

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

DISABLED
ENABLED

H264Syntax (Enum)

Produces a bitstream compliant with SMPTE RP-2027.

DEFAULT
**H264TemporalAq (Enum)**

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

- DISABLED
- ENABLED

**H264TimecodeInsertionBehavior (Enum)**

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

- DISABLED
- PIC_TIMING_SEI

**HlsAdMarkers (Enum)**

- ADOBE
- ELEMENTAL
- ELEMENTAL_SCTE35

**HlsAkamaiHttpTransferMode (Enum)**

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

- CHUNKED
- NON_CHUNKED

**HlsAkamaiSettings**

**httpTransferMode**

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

- **Type**: HlsAkamaiHttpTransferMode (p. 294)
- **Required**: False

**salt**

Salt for authenticated Akamai.

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

**numRetries**

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

**Type**
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**

**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

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 (Enum)

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

INSERT
NONE
OMIT

HlsCdnSettings

hlsAkamaiSettings

Type: HlsAkamaiSettings (p. 294)
Required: False

hlsWebdavSettings

Type: HlsWebdavSettings (p. 307)
Required: False

hlsBasicPutSettings

Type: HlsBasicPutSettings (p. 295)
Required: False
hlsMediaStoreSettings

Type: HlsMediaStoreSettings (p. 304)
Required: False

HlsClientCache (Enum)

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.

DISABLED
ENABLED

HlsCodecSpecification (Enum)

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

RFC_4281
RFC_6381

HlsDirectoryStructure (Enum)

Place segments in subdirectories.

SINGLE_DIRECTORY
SUBDIRECTORY_PER_STREAM

HlsEncryptionType (Enum)

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

AES128
SAMPLE_AES

HlsGroupSettings

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. 303)
**outputSelection**
Generates the .m3u8 playlist file for this HLS output group. The segmentsOnly option will output segments without the .m3u8 file.

*Type:* HlsOutputSelection (p. 305)
*Required:* False

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

*Type:* HlsEncryptionType (p. 297)
*Required:* False

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

*Type:* OutputLocationRef (p. 332)
*Required:* True

**indexNSegments**
Number of segments to keep in the playlist (.m3u8) file. mode must be "vod" for this setting to have an effect, and this number should be less than or equal to keepSegments.

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

**timedMetadataId3Frame**
Indicates ID3 frame that has the timecode.

*Type:* HlsTimedMetadataId3Frame (p. 306)
*Required:* False

**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

**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. 296)
- **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 set to "vod", keeps and indexes all segments starting with the first segment. If set to "live" segments will age out and only the last keepSegments number of segments will be retained.

- **Type**: HlsMode (p. 305)
- **Required**: False

**keyProviderSettings**

The key provider settings.

- **Type**: KeyProviderSettings (p. 312)
- **Required**: False

**manifestCompression**

When set to gzip, compresses HLS playlist.

- **Type**: HlsManifestCompression (p. 303)
- **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.

- **Type**: HlsIvSource (p. 303)
Properties

Required: False

tsFileMode

When set to "singleFile", emits the program as a single media resource (.ts) file, and uses #EXT-X-BYTERANGE tags to index segment for playback. Playback of VOD mode content during event is not guaranteed due to HTTP server caching.

Type: HlsTsFileMode (p. 306)
Required: False

manifestDurationFormat

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

Type: HlsManifestDurationFormat (p. 304)
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. 306)
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

When set to useInputSegmentation, the output segment or fragment points are set by the RAI markers from the input streams.

Type: HlsSegmentationMode (p. 306)
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. 267)
  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. 297)
  Required: False

codecSpecification

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

  Type: HlsCodecSpecification (p. 297)
  Required: False

keepSegments

Number of segments to retain in the destination directory. mode must be "live" for this setting to have an effect.

  Type: integer
  Required: False
  Minimum: 1

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. 305)
  Required: False

directoryStructure

Place segments in subdirectories.

  Type: HlsDirectoryStructure (p. 297)
  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. 309)
- **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. 294)
- **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. 296)
- **Required**: False

**HlsInputSettings**

**retries**

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

**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 (Enum)**

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.

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

**HlsIvSource (Enum)**

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.

- **EXPLICIT**  
- **FOLLOWS_SEGMENT_NUMBER**

**HlsManifestCompression (Enum)**

When set to gzip, compresses HLS playlist.
GZIP
NONE

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

HlsMediaStoreSettings

mediaStoreStorageClass
When set to temporal, output files are stored in non-persistent memory for faster reading and writing.
  - Type: HlsMediaStoreStorageClass (p. 305)
  - 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 (Enum)
When set to temporal, output files are stored in non-persistent memory for faster reading and writing.

TEMPORAL

HlsMode (Enum)
If set to "vod", keeps and indexes all segments starting with the first segment. If set to "live" segments will age out and only the last keepSegments number of segments will be retained.

LIVE
VOD

HlsOutputSelection (Enum)
Generates the .m3u8 playlist file for this HLS output group. The segmentsOnly option will output segments without the .m3u8 file.

MANIFESTS_AND_SEGMENTS
SEGMENTS_ONLY

HlsOutputSettings

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. 306)
Required: True

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

Type: string
Required: True

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

When set to `useInputSegmentation`, the output segment or fragment points are set by the RAI markers from the input streams.

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

**HlsSettings**

**standardHlsSettings**

*Type:* StandardHlsSettings (p. 337)  
*Required:* False

**audioOnlyHlsSettings**

*Type:* AudioOnlyHlsSettings (p. 257)  
*Required:* False

**HlsStreamInfResolution (Enum)**

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

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

**HlsTimedMetadataId3Frame (Enum)**

Indicates ID3 frame that has the timecode.

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

**HlsTsFileMode (Enum)**

When set to "singleFile", emits the program as a single media resource (.ts) file, and uses #EXT-X-BYTERANGE tags to index segment for playback. Playback of VOD mode content during event is not guaranteed due to HTTP server caching.

- **SEGMENTED_FILES**
- **SINGLE_FILE**

**HlsWebdavHttpTransferMode (Enum)**

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

- **CHUNKED**
## NON_CHUNKED

### HlsWebdavSettings

#### httpTransferMode

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

- **Type:** HlsWebdavHttpTransferMode (p. 306)
- **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

### InputAttachment

#### inputId

The ID of the input
**Properties**

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

**inputSettings**

Settings of an input (caption selector, etc.)

**Type**: InputSettings (p. 311)  
**Required**: False

**InputChannelLevel**

**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

**InputDeblockFilter (Enum)**

Enable or disable the deblock filter when filtering.

DISABLED  
ENABLED

**InputDenoiseFilter (Enum)**

Enable or disable the denoise filter when filtering.

DISABLED  
ENABLED

**InputFilter (Enum)**

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

AUTO  
DISABLED
InputLocation

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 rtmpEndpoint 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 (Enum)

Parameter that controls output group behavior on input loss.

EMIT_OUTPUT
PAUSE_OUTPUT

InputLossActionForMsSmoothOut (Enum)

Parameter that controls output group behavior on input loss.

EMIT_OUTPUT
PAUSE_OUTPUT

InputLossActionForUdpOut (Enum)

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.

DROP_PROGRAM
DROP_TS
EMIT_PROGRAM

**InputLossBehavior**

**inputLossImageType**
Indicates whether to substitute a solid color or a slate into the output after input loss exceeds blackFrameMsec.

- **Type**: `InputLossImageType (p. 310)`
- **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

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

- **Type**: `InputLocation (p. 309)`
- **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 (Enum)**
Indicates whether to substitute a solid color or a slate into the output after input loss exceeds blackFrameMsec.

COLOR
SLATE

**InputSettings**

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

- **Type**: `InputSourceEndBehavior (p. 312)`
- **Required**: False

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

- **Type**: Array of type `AudioSelector (p. 258)`
- **Required**: False

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

- **Type**: `InputDeblockFilter (p. 308)`
- **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. 308)`
- **Required**: False

**networkInputSettings**
Input settings.

- **Type**: `NetworkInputSettings (p. 329)`
- **Required**: True

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

- **Type**: `VideoSelector (p. 342)`
- **Required**: False

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

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

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

Type: Array of type CaptionSelector (p. 267)
Required: False

denoiseFilter
Enable or disable the denoise filter when filtering.

Type: InputDenoiseFilter (p. 308)
Required: False

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

CONTINUE
LOOP

InternalServiceError
message

Type: string
Required: False

InvalidRequest
message

Type: string
Required: False

KeyProviderSettings
staticKeySettings

Type: StaticKeySettings (p. 337)
Required: False

LimitExceeded
message

Type: string
Required: False
M2tsAbsentInputAudioBehavior (Enum)
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.

   DROP
   ENCODE_SILENCE

M2tsArib (Enum)
When set to enabled, uses ARIB-compliant field muxing and removes video descriptor.

   DISABLED
   ENABLED

M2tsAribCaptionsPidControl (Enum)
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.

   AUTO
   USE_CONFIGURED

M2tsAudioBufferModel (Enum)
When set to dvb, uses DVB buffer model for Dolby Digital audio. When set to atsc, the ATSC model is used.

   ATSC
   DVB

M2tsAudioInterval (Enum)
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.

   VIDEO_AND_FIXED_INTERVALS
   VIDEO_INTERVAL

M2tsAudioStreamType (Enum)
When set to atsc, uses stream type = 0x81 for AC3 and stream type = 0x87 for EAC3. When set to dvb, uses stream type = 0x06.

   ATSC
   DVB

M2tsBufferModel (Enum)
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.
MULTIPLEX
NONE

M2tsCcDescriptor (Enum)
When set to enabled, generates captionServiceDescriptor in PMT.

DISABLED
ENABLED

M2tsEbifControl (Enum)
If set to passthrough, passes any EBIF data from the input source to this output.

NONE
PASSTHROUGH

M2tsEbpPlacement (Enum)
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.

VIDEO_AND_AUDIO_PIDS
VIDEO_PID

M2tsEsRateInPes (Enum)
Include or exclude the ES Rate field in the PES header.

EXCLUDE
INCLUDE

M2tsKlv (Enum)
If set to passthrough, passes any KLV data from the input source to this output.

NONE
PASSTHROUGH

M2tsPcrControl (Enum)
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.

CONFIGURED_PCR_PERIOD
PCR_EVERY_PES_PACKET

M2tsRateMode (Enum)
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.

CBR
VBR
M2tsScte35Control (Enum)

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

   NONE
   PASSTHROUGH

M2tsSegmentationMarkers (Enum)

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.

   EBP
   EBP_LEGACY
   NONE
   PSI_SEGSTART
   RAI_ADAPT
   RAI_SEGSTART

M2tsSegmentationStyle (Enum)

The segmentation style parameter controls how segmentation markers are inserted into the transport stream. With avails, it is possible that segments may be truncated, which can influence where future segmentation markers are inserted. When a segmentation style of "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.

   MAINTAIN_CADENCE
   RESET_CADENCE

M2tsSettings

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. 313)
   Required: False

ecmPid

Packet Identifier (PID) for ECM in the transport stream. Only enabled when Simulcrypt is enabled. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20).8182 (or 0xff).

   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. 313)
- **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

**segmentationTime**

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

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

**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. 314)
- **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. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

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

**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. 313)  
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

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

pcrPeriod
Maximum time in milliseconds between Program Clock Reference (PCRs) inserted into the transport stream.
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>pmtInterval</td>
<td>integer</td>
<td>False</td>
<td>0</td>
<td>1000</td>
</tr>
<tr>
<td>segmentationStyle</td>
<td>M2tsSegmentationStyle (p. 315)</td>
<td>False</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebif</td>
<td>M2tsEbifControl (p. 314)</td>
<td>False</td>
<td></td>
<td></td>
</tr>
<tr>
<td>audioBufferModel</td>
<td>M2tsAudioBufferModel (p. 313)</td>
<td>False</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dvbNitSettings</td>
<td>DvbNitSettings (p. 270)</td>
<td>False</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**pmtInterval**

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

**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 $\text{segmentationTime}$ 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 $\text{segmentationTime}$ seconds. Note that EBP lookahead is a slight exception to this rule.

**ebif**

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

**audioBufferModel**

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

**dvbNitSettings**

Inserts DVB Network Information Table (NIT) at the specified table repetition interval.
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. 313)
**Required:** False

timedMetadataBehavior

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

**Type:** M2tsTimedMetadataBehavior (p. 322)
**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. 313)
**Required:** False

scte35Control

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

**Type:** M2tsScte35Control (p. 315)
**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. 314)
Required: False

arib

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

Type: M2tsArib (p. 313)
Required: False

dvbSdtSettings

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

Type: DvbSdtSettings (p. 271)
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

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

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

### 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. 314)
**Required**: False

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

**Type**: M2tsEsRateInPes (p. 314)
**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. 315)
**Required**: False

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

**Type**: M2tsKlv (p. 314)
**Required**: False

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

**Type**: DvbTdtSettings (p. 276)
**Required**: False

### ccDescriptor
When set to enabled, generates captionServiceDescriptor in PMT.

**Type**: M2tsCcDescriptor (p. 314)
**Required**: False

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

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 (Enum)

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

NO_PASSTHROUGH
PASSTHROUGH
**M3u8PcrControl (Enum)**

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.

- `CONFIGURED_PCR_PERIOD`
- `PCR_EVERY_PES_PACKET`

**M3u8Scte35Behavior (Enum)**

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

- `NO_PASSTHROUGH`
- `PASSTHROUGH`

**M3u8Settings**

**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**

The Platform-protected transport streams using 'microsoft' as Target Client include an ECM stream. This ECM stream contains the size, IV, and PTS of every sample in the transport stream. This stream PID is specified here. This PID has no effect on non ThePlatform-protected streams.

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

**scte35Behavior**

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

- **Type**: `M3u8Scte35Behavior (p. 323)`
- **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

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

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. 323)
Required: False

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

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

**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. A value of \"0\" writes out the PMT once per segment file.

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

**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

**timedMetadataBehavior**

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

**Type**: `M3u8TimedMetadataBehavior` (p. 325)  
**Required**: False

**M3u8TimedMetadataBehavior (Enum)**

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

- NO_PASSTHROUGH  
- PASSTHROUGH

**Mp2CodingMode (Enum)**

The MPEG2 Audio coding mode. Valid values are codingMode10 (for mono) or codingMode20 (for stereo).
CODING_MODE_1_0
CODING_MODE_2_0

Mp2Settings

codingMode

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

Type: Mp2CodingMode (p. 325)
Required: False

bitrate

Average bitrate in bits/second.

Type: number
Required: False

sampleRate

Sample rate in Hz.

Type: number
Required: False

MsSmoothGroupSettings

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

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

timestampOffset

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

Type: string
Required: False
**segmentationMode**

When set to useInputSegmentation, the output segment or fragment points are set by the RAI markers from the input streams.

*Type: SmoothGroupSegmentationMode (p. 336)*

*Required: False*

**numRetries**

Number of retry attempts.

*Type: integer*

*Required: False*

*Minimum: 0*

**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*

**eventStopBehavior**

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

*Type: SmoothGroupEventStopBehavior (p. 336)*

*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. 337)*

*Required: False*

**destination**

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

*Type: OutputLocationRef (p. 332)*

*Required: True*

**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. 337)*

*Required: False*
**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. 336)
- **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 unless those certificates are manually added to the OS trusted keystore.

- **Type**: SmoothGroupCertificateMode (p. 336)
- **Required**: False

**inputLossAction**

Parameter that control output group behavior on input loss.

- **Type**: InputLossActionForMsSmoothOut (p. 309)
- **Required**: False

**sendDelayMs**

Outputs that are "output locked" can use this delay. Assign a delay to the output that is "secondary". Do not assign a delay to the "primary" output. The delay means that the primary output will always reach the downstream system before the secondary, which helps ensure that the downstream system always uses the primary output. (If there were no delay, the downstream system might flip-flop between whichever output happens to arrive first.) If the primary fails, the downstream system will switch to the secondary output. When the primary is restarted, the downstream system will switch back to the primary (because once again it is always arriving first).

- **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. 336)
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. 337)
Required: False

MsSmoothOutputSettings

nameModifier

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

Type: string
Required: False

NetworkInputServerValidation (Enum)

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.

CHECK_CRYPTOGRAPHY_AND_VALIDATE_NAME
CHECK_CRYPTOGRAPHY_ONLY

NetworkInputSettings

hlsInputSettings

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

Type: HlsInputSettings (p. 302)
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. 329)
- **Required**: False

Output

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

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. 332)
- **Required**: True

audioDescriptionNames

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

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

OutputDestination

settings

Destination settings for output; one for each redundant encoder.

- **Type**: Array of type OutputDestinationSettings (p. 331)
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

url
A URL specifying a destination

   Type: string
   Required: False

username
username for destination

   Type: string
   Required: False

OutputGroup

outputs

   Type: Array of type Output (p. 330)
   Required: True

outputGroupSettings

Settings associated with the output group.

   Type: OutputGroupSettings (p. 332)
   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

**OutputGroupSettings**

*archiveGroupSettings*

Type: ArchiveGroupSettings (p. 252)  
Required: False

*udpGroupSettings*

Type: UdpGroupSettings (p. 339)  
Required: False

*msSmoothGroupSettings*

Type: MsSmoothGroupSettings (p. 326)  
Required: False

*hlsGroupSettings*

Type: HlsGroupSettings (p. 297)  
Required: False

**OutputLocationRef**

*destinationRefId*

Type: string  
Required: False

**OutputSettings**

*archiveOutputSettings*

Type: ArchiveOutputSettings (p. 252)  
Required: False

*msSmoothOutputSettings*

Type: MsSmoothOutputSettings (p. 329)  
Required: False

*udpOutputSettings*

Type: UdpOutputSettings (p. 340)  
Required: False

*hlsOutputSettings*

Type: HlsOutputSettings (p. 305)
**PassThroughSettings**

**RemixSettings**

**channelMappings**

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

- **Type:** Array of type `AudioChannelMapping (p. 253)`
- **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

**Scte20Convert608To708 (Enum)**

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.

- **DISABLED**
- **UPCONVERT**
**Scte20PlusEmbeddedDestinationSettings**

**Scte20SourceSettings**

**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. 333)
- **Required:** False

**Scte27DestinationSettings**

**Scte27SourceSettings**

**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** *(Enum)*

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

- FOLLOW
- IGNORE

**Scte35AposWebDeliveryAllowedBehavior** *(Enum)*

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

- FOLLOW
- IGNORE
**Scte35SpliceInsert**

**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. 335)
- **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. 335)
- **Required**: False

**Scte35SpliceInsertNoRegionalBlackoutBehavior (Enum)**

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

- FOLLOW
- IGNORE

**Scte35SpliceInsertWebDeliveryAllowedBehavior (Enum)**

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

- FOLLOW
- IGNORE

**Scte35TimeSignalApos**

**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. 334)`  
*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. 334)`  
*Required:* False

**SmoothGroupAudioOnlyTimecodeControl (Enum)**

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.

- PASSTHROUGH
- USE_CONFIGURED_CLOCK

**SmoothGroupCertificateMode (Enum)**

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 unless those certificates are manually added to the OS trusted keystore.

- SELF_SIGNED
- VERIFY_AUTHENTICITY

**SmoothGroupEventIdMode (Enum)**

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.

- NO_EVENT_ID
- USE_CONFIGURED
- USE_TIMESTAMP

**SmoothGroupEventStopBehavior (Enum)**

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

- NONE
- SEND_EOS

**SmoothGroupSegmentationMode (Enum)**

When set to `useInputSegmentation`, the output segment or fragment points are set by the RAI markers from the input streams.
USE_INPUT_SEGMENTATION
USE_SEGMENT_DURATION

**SmoothGroupSparseTrackType (Enum)**
If set to scte35, use incoming SCTE-35 messages to generate a sparse track in this group of MS-Smooth outputs.

NONE
SCTE_35

**SmoothGroupStreamManifestBehavior (Enum)**
When set to send, send stream manifest so publishing point doesn't start until all streams start.

DO_NOT_SEND
SEND

**SmoothGroupTimestampOffsetMode (Enum)**
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

USE_CONFIGURED_OFFSET
USE_EVENT_START_DATE

**SmpteTtDestinationSettings**

**StandardHlsSettings**

**m3u8Settings**
- **Type:** M3u8Settings (p. 323)
- **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**

**staticKeyValue**
Static key value as a 32 character hexadecimal string.
- **Type:** string
- **Required:** True
keyProviderServer

The URL of the license server used for protecting content.

Type: InputLocation (p. 309)
Required: False

TeletextDestinationSettings

TeletextSourceSettings

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

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. 338)
Required: True

TimecodeConfigSource (Enum)

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.

EMBEDDED
SYSTEMCLOCK
**TtmlDestinationSettings**

**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. 339)
- **Required**: False

**TtmlDestinationStyleControl (Enum)**
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.

- **PASSTHROUGH**
- **USE_CONFIGURED**

**UdpContainerSettings**

**m2tsSettings**
- **Type**: M2tsSettings (p. 315)
- **Required**: False

**UdpGroupSettings**

**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. 309)
- **Required**: False

**timedMetadataId3Frame**
Indicates ID3 frame that has the timecode.

- **Type**: UdpTimedMetadataId3Frame (p. 340)
- **Required**: False

**timedMetadataId3Period**
Timed Metadata interval in seconds.

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

destination

Destination address and port number for RTP or UDP packets. Can be unicast or multicast RTP or UDP (e.g. rtp://239.10.10.10:5001 or udp://10.100.100.100:5002).

Type: OutputLocationRef (p. 332)
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. 339)
Required: True

fecOutputSettings

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

Type: FecOutputSettings (p. 283)
Required: False

UdpTimedMetadataId3Frame (Enum)

Indicates ID3 frame that has the timecode.

NONE
PRIV
TDRL

VideoCodecSettings

h264Settings

Type: H264Settings (p. 288)
Required: False

VideoDescription

respondToAfd

Indicates how to respond to the AFD values in the input stream. Setting to "respond" causes input video to be clipped, depending on AFD value, input display aspect ratio and output display aspect ratio.
**Properties**

- **Type**: `VideoDescriptionRespondToAfd` *(p. 342)*
  - **Required**: False

**scalingBehavior**

When set to "stretchToOutput", automatically configures the output position to stretch the video to the specified output resolution. This option will override any position value.

- **Type**: `VideoDescriptionScalingBehavior` *(p. 342)*
  - **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). Leave out to use source video width. If left out, height must also be left out. Display aspect ratio is always preserved by letterboxing or pillarboxing when necessary.

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

**sharpness**

Changes the width of the anti-alias filter kernel used for scaling. Only applies if scaling is being performed and antiAlias is set to true. 0 is the softest setting, 100 the sharpest, and 50 recommended for most content.

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

**codecSettings**

Video codec settings.

- **Type**: `VideoCodecSettings` *(p. 340)*
  - **Required**: True

**height**

Output video height (in pixels). Leave blank to use source video height. If left blank, width must also be unspecified.

- **Type**: integer
  - **Required**: False
**VideoDescriptionRespondToAfd (Enum)**

Indicates how to respond to the AFD values in the input stream. Setting to "respond" causes input video to be clipped, depending on AFD value, input display aspect ratio and output display aspect ratio.

- NONE
- PASSTHROUGH
- RESPOND

**VideoDescriptionScalingBehavior (Enum)**

When set to "stretchToOutput", automatically configures the output position to stretch the video to the specified output resolution. This option will override any position value.

- DEFAULT
- STRETCH_TO_OUTPUT

**VideoSelector**

**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. 342)
- **Required**: False

**selectorSettings**

The video selector settings.

- **Type**: VideoSelectorSettings (p. 343)
- **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. 343)
- **Required**: False

**VideoSelectorColorSpace (Enum)**

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

- FOLLOW
- REC_601
- REC_709
VideoSelectorColorSpaceUsage (Enum)

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.

FALLBACK
FORCE

VideoSelectorPid

pid

Selects a specific PID from within a video source.

Type: integer
Required: False
Minimum: 0
Maximum: 8191

VideoSelectorProgramId

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

videoSelectorPid

Type: VideoSelectorPid (p. 343)
Required: False

videoSelectorProgramId

Type: VideoSelectorProgramId (p. 343)
Required: False
WebvttDestinationSettings

Channels `channelId` Stop

**URI**

/`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><code>channelId</code></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. 345)</td>
<td>Successfully initiated stop of the channel.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 355)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 355)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>None</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 355)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFoundException (p. 355)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>504</td>
<td>None</td>
<td>Gateway Timeout Error</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 356)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 355)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
</tbody>
</table>
Schemas

Response Bodies

Example Channel

```json
{
  "inputAttachments (p. 377)": [
    {
      "inputId (p. 416)": "string",
      "inputSettings (p. 417)": {
        "sourceEndBehavior (p. 420)": enum,
        "audioSelectors (p. 420)": [
          {
            "name (p. 367)": "string",
            "selectorSettings (p. 367)": {
              "audioLanguageSelection (p. 367)": {
                "languageSelectionPolicy (p. 364)": enum,
                "languageCode (p. 365)": "string"
              },
              "audioPidSelection (p. 367)": {
                "pid (p. 367)": integer
              }
            }
          }
        ],
        "deblockFilter (p. 420)": enum,
        "inputFilter (p. 420)": enum,
        "networkInputSettings (p. 420)": {
          "hlsInputSettings (p. 438)": {
            "retries (p. 411)": integer,
            "bandwidth (p. 412)": integer,
            "retryInterval (p. 412)": integer,
            "bufferSegments (p. 412)": integer
          },
          "serverValidation (p. 439)": enum
        },
        "videoSelector (p. 420)": {
          "colorSpace (p. 451)": enum,
          "selectorSettings (p. 451)": {
            "videoSelectorPid (p. 452)": {
              "pid (p. 452)": integer
            },
            "videoSelectorProgramId (p. 452)": {
              "programId (p. 452)": integer
            }
          },
          "colorSpaceUsage (p. 451)": enum
        },
        "filterStrength (p. 420)": integer,
        "captionSelectors (p. 421)": [
          {
            "name (p. 376)": "string",
            "languageCode (p. 376)": "string",
            "selectorSettings (p. 376)": {
              "embeddedSourceSettings (p. 377)": {
                "scte20Detection (p. 390)": enum,
                "source608ChannelNumber (p. 391)": integer,
                "convert608To708 (p. 391)": enum,
                "source608TrackNumber (p. 391)": integer
              },
              "scte20SourceSettings (p. 377)": {
                "source608ChannelNumber (p. 443)": integer,
              }
            }
          }
        ]
      }
    }
  ]
}
```
AWS Elemental MediaLive API Reference
Schemas

"convert608To708 (p. 443)": enum
},
"dvbSubSourceSettings (p. 377)": {
"pid (p. 385)": integer
},
"teletextSourceSettings (p. 377)": {
"pageNumber (p. 447)": "string"
},
"aribSourceSettings (p. 377)": {
},
"scte27SourceSettings (p. 377)": {
"pid (p. 443)": integer
}
],
"denoiseFilter (p. 421)": enum
},
"roleArn (p. 377)": "string",
"destinations (p. 377)": [
{
"settings (p. 439)": [
{
"passwordParam (p. 440)": "string",
"url (p. 440)": "string",
"username (p. 440)": "string"
}
],
"id (p. 440)": "string"
}
],
"name (p. 378)": "string",
"encoderSettings (p. 378)": {
"timecodeConfig (p. 391)": {
"syncThreshold (p. 447)": integer,
"source (p. 447)": enum
},
"outputGroups (p. 391)": [
{
"outputs (p. 440)": [
{
"videoDescriptionName (p. 439)": "string",
"outputName (p. 439)": "string",
"captionDescriptionNames (p. 439)": [
"string"
]
},
"archiveOutputSettings (p. 441)": {
"extension (p. 361)": "string",
"containerSettings (p. 361)": {
"m2tsSettings (p. 361)": {
"audioStreamType (p. 424)": enum,
"ecmPid (p. 424)": "string",
"dvbTeletextPid (p. 425)": "string",
"aribCaptionsPidControl (p. 425)": enum,
"bitrate (p. 425)": integer,
"segmentationTime (p. 425)": number,
"rateMode (p. 425)": enum,
"audioPids (p. 425)": "string",
"ebpLookaheadMs (p. 425)": integer,
"ebpAudioInterval (p. 426)": enum,
"audioFramesPerPes (p. 426)": integer,
"fragmentTime (p. 426)": number,
"scte35Pid (p. 426)": "string",
"containerSettings (p. 361)": {
"m2tsSettings (p. 361)": {
"audioStreamType (p. 424)": enum,
"ecmPid (p. 424)": "string",
"dvbTeletextPid (p. 425)": "string",
"aribCaptionsPidControl (p. 425)": enum,
"bitrate (p. 425)": integer,
"segmentationTime (p. 425)": number,
"rateMode (p. 425)": enum,
"audioPids (p. 425)": "string",
"ebpLookaheadMs (p. 425)": integer,
"ebpAudioInterval (p. 426)": enum,
"audioFramesPerPes (p. 426)": integer,
"fragmentTime (p. 426)": number,
"scte35Pid (p. 426)": "string",
"containerSettings (p. 361)": {
"m2tsSettings (p. 361)": {
"audioStreamType (p. 424)": enum,
"ecmPid (p. 424)": "string",
"dvbTeletextPid (p. 425)": "string",
"aribCaptionsPidControl (p. 425)": enum,
"bitrate (p. 425)": integer,
"segmentationTime (p. 425)": number,
"rateMode (p. 425)": enum,
"audioPids (p. 425)": "string",
"ebpLookaheadMs (p. 425)": integer,
"ebpAudioInterval (p. 426)": enum,
"audioFramesPerPes (p. 426)": integer,
"fragmentTime (p. 426)": number,
"scte35Pid (p. 426)": "string",
"containerSettings (p. 361)": {
"m2tsSettings (p. 361)": {
"audioStreamType (p. 424)": enum,
"ecmPid (p. 424)": "string",
"dvbTeletextPid (p. 425)": "string",
"aribCaptionsPidControl (p. 425)": enum,
"bitrate (p. 425)": integer,
"segmentationTime (p. 425)": number,
"rateMode (p. 425)": enum,
"audioPids (p. 425)": "string",
"ebpLookaheadMs (p. 425)": integer,
"ebpAudioInterval (p. 426)": enum,
"audioFramesPerPes (p. 426)": integer,
"fragmentTime (p. 426)": number,
"scte35Pid (p. 426)": "string",
"containerSettings (p. 361)": {
"m2tsSettings (p. 361)": {
"audioStreamType (p. 424)": enum,
"ecmPid (p. 424)": "string",
"dvbTeletextPid (p. 425)": "string",
"aribCaptionsPidControl (p. 425)": enum,
"bitrate (p. 425)": integer,
"segmentationTime (p. 425)": number,
"rateMode (p. 425)": enum,
"audioPids (p. 425)": "string",
"ebpLookaheadMs (p. 425)": integer,
"ebpAudioInterval (p. 426)": enum,
"audioFramesPerPes (p. 426)": integer,
"fragmentTime (p. 426)": number,
"scte35Pid (p. 426)": "string",
"containerSettings (p. 361)": {
"m2tsSettings (p. 361)": {
"audioStreamType (p. 424)": enum,
"ecmPid (p. 424)": "string",
"dvbTeletextPid (p. 425)": "string",
"aribCaptionsPidControl (p. 425)": enum,
"bitrate (p. 425)": integer,
"segmentationTime (p. 425)": number,
"rateMode (p. 425)": enum,
"audioPids (p. 425)": "string",
"ebpLookaheadMs (p. 425)": integer,
"ebpAudioInterval (p. 426)": enum,
"audioFramesPerPes (p. 426)": integer,
"fragmentTime (p. 426)": number,
"scte35Pid (p. 426)": "string",
"containerSettings (p. 361)": {
"m2tsSettings (p. 361)": {
"audioStreamType (p. 424)": enum,
"ecmPid (p. 424)": "string",
"dvbTeletextPid (p. 425)": "string",
"aribCaptionsPidControl (p. 425)": enum,
"programNum (p. 426)" : integer,
"pcrPeriod (p. 426)" : integer,
"pmtInterval (p. 427)" : integer,
"segmentationStyle (p. 427)" : enum,
"ebif (p. 427)" : enum,
"audioBufferModel (p. 427)" : enum,
"dvbNitSettings (p. 427)" :
  "networkName (p. 379)" : "string",
  "networkId (p. 379)" : integer,
  "repInterval (p. 379)" : integer
},
"absentInputAudioBehavior (p. 428)" : enum,
"timedMetadataBehavior (p. 428)" : enum,
"timedMetadataPid (p. 428)" : "string",
"pmtPid (p. 428)" : "string",
"etvSignalPid (p. 428)" : "string",
"bufferModel (p. 428)" : enum,
"scte35Control (p. 428)" : enum,
"ebpPlacement (p. 429)" : enum,
"arib (p. 429)" : enum,
"dvbSdtSettings (p. 429)" :
  "serviceName (p. 380)" : "string",
  "serviceProviderName (p. 380)" : "string",
  "repInterval (p. 380)" : integer,
  "outputSdt (p. 380)" : enum
},
"nullPacketBitrate (p. 429)" : number,
"pcrPid (p. 429)" : "string",
"transportStreamId (p. 429)" : integer,
"videoPid (p. 429)" : "string",
"pcrControl (p. 430)" : enum,
"esRateInPes (p. 430)" : enum,
"segmentationMarkers (p. 430)" : enum,
"klv (p. 430)" : enum,
"dvbTdtSettings (p. 430)" :
  "repInterval (p. 385)" : integer
},
"ccDescriptor (p. 430)" : enum,
"patInterval (p. 430)" : integer,
"etvPlatformPid (p. 431)" : "string",
"dvbSubPids (p. 431)" : "string",
"scte27Pids (p. 431)" : "string",
"aribCaptionsPid (p. 431)" : "string",
"klvDataPids (p. 431)" : "string"
},
"nameModifier (p. 361)" : "string"
},
"msSmoothOutputSettings (p. 441)" :
  "nameModifier (p. 438)" : "string"
},
"udpOutputSettings (p. 441)" :
  "destination (p. 449)" :
    "destinationRefId (p. 441)" : "string"
},
"bufferMsec (p. 449)" : integer,
"containerSettings (p. 449)" :
  "m2tsSettings (p. 448)" :
    "audioStreamType (p. 424)" : enum,
    "ecmPid (p. 424)" : "string",
    "dvbTeletextPid (p. 425)" : "string",
    "aribCaptionsPidControl (p. 425)" : enum,
    "bitrate (p. 425)" : integer,
    "segmentationTime (p. 425)" : number,
    "rateMode (p. 425)" : enum,
    "audioPids (p. 425)" : "string"
"ebpLookaheadMs (p. 425)" : integer,
"ebpAudioInterval (p. 426)" : enum,
"audioFramesPerPes (p. 426)" : integer,
"fragmentTime (p. 426)" : number,
"scte35Pid (p. 426)" : "string",
"programNum (p. 426)" : integer,
"pcrPeriod (p. 426)" : integer,
"pmtInterval (p. 427)" : integer,
"segmentationStyle (p. 427)" : enum,
"ebif (p. 427)" : enum,
"audioBufferModel (p. 427)" : enum,
"dvbNitSettings (p. 427)" : {
  "networkName (p. 379)" : "string",
  "networkId (p. 379)" : integer,
  "repInterval (p. 379)" : integer
},
"absentInputAudioBehavior (p. 428)" : enum,
"timedMetadataBehavior (p. 428)" : enum,
"timedMetadataPid (p. 428)" : "string",
"pmtPid (p. 428)" : "string",
"etvSignalPid (p. 428)" : "string",
"bufferModel (p. 428)" : enum,
"scte35Control (p. 428)" : enum,
"ebpPlacement (p. 429)" : enum,
"arib (p. 429)" : enum,
"dvbSdtSettings (p. 429)" : {
  "serviceName (p. 380)" : "string",
  "serviceProviderName (p. 380)" : "string",
  "repInterval (p. 380)" : integer,
  "outputSdt (p. 380)" : enum
},
"nullPacketBitrate (p. 429)" : number,
"pcrPid (p. 429)" : "string",
"transportStreamId (p. 429)" : integer,
"videoPid (p. 429)" : "string",
"pcrControl (p. 430)" : enum,
"esRateInPes (p. 430)" : enum,
"segmentationMarkers (p. 430)" : enum,
"klv (p. 430)" : enum,
"dvbTdtSettings (p. 430)" : {
  "repInterval (p. 385)" : integer
},
"ccDescriptor (p. 430)" : enum,
"patInterval (p. 430)" : integer,
"etvPlatformPid (p. 431)" : "string",
"dvbSubPids (p. 431)" : "string",
"aribCaptionsPid (p. 431)" : "string",
"scte27Pids (p. 431)" : "string",
"klvDataPids (p. 431)" : "string"
}},{
"fecOutputSettings (p. 449)" : {
  "rowLength (p. 392)" : integer,
  "columnDepth (p. 393)" : integer,
  "includeFec (p. 393)" : enum
},
"hlsOutputSettings (p. 441)" : {
  "segmentModifier (p. 414)" : "string",
  "hlsSettings (p. 414)" : {
    "standardHlsSettings (p. 415)" : {
      "m3u8Settings (p. 446)" : {
        "pmtPid (p. 432)" : "string",
        "ecmPid (p. 432)" : "string",
        "scte35Behavior (p. 432)" : enum,
        "pcrPid (p. 432)" : "string",
        "pcrControl (p. 430)" : enum,
        "esRateInPes (p. 430)" : enum,
        "segmentationMarkers (p. 430)" : enum,
        "klv (p. 430)" : enum,
        "dvbTdtSettings (p. 430)" : {
          "repInterval (p. 385)" : integer
        },
        "ccDescriptor (p. 430)" : enum,
        "patInterval (p. 430)" : integer,
        "etvPlatformPid (p. 431)" : "string",
        "dvbSubPids (p. 431)" : "string",
        "aribCaptionsPid (p. 431)" : "string",
        "scte27Pids (p. 431)" : "string",
        "klvDataPids (p. 431)" : "string"
      }
    }
  }
}
"audioPids (p. 433)": "string",
"audioFramesPerPes (p. 433)": integer,
"scte35Pid (p. 433)": "string",
"transportStreamId (p. 433)": integer,
"videoPid (p. 433)": "string",
"pcrControl (p. 433)": enum,
"pcrPeriod (p. 433)": integer,
"programNum (p. 434)": integer,
"pmtInterval (p. 434)": integer,
"patInterval (p. 434)": integer,
"timedMetadataBehavior (p. 434)": enum,
"audioRenditionSets (p. 446)": "string",
"audioOnlyHlsSettings (p. 415)": {
  "audioTrackType (p. 366)": enum,
  "audioGroupId (p. 366)": "string",
  "audioOnlyImage (p. 366)": {
    "passwordParam (p. 418)": "string",
    "uri (p. 418)": "string",
    "username (p. 418)": "string"
  }
}
"nameModifier (p. 414)": "string"
},
"audioDescriptionNames (p. 439)": [
"string"
],
"audioGroupSettings (p. 440)": {
  "archiveGroupSettings (p. 441)": {
    "destination (p. 361)": {
      "destinationRefId (p. 441)": "string"
    },
    "rolloverInterval (p. 361)": integer
  },
  "udpGroupSettings (p. 441)": {
    "inputLossAction (p. 448)": enum,
    "timedMetadataId3Frame (p. 448)": enum,
    "timedMetadataId3Period (p. 448)": integer
  },
  "msSmoothGroupSettings (p. 441)": {
    "eventId (p. 435)": "string",
    "segmentationMode (p. 436)": enum,
    "numRetries (p. 436)": integer,
    "acquisitionPointId (p. 436)": "string",
    "eventStopBehavior (p. 436)": enum,
    "sparseTrackType (p. 436)": enum,
    "destination (p. 436)": {
      "destinationRefId (p. 441)": "string"
    },
    "timestampOffsetMode (p. 436)": enum,
    "audioOnlyTimecodeControl (p. 437)": enum,
    "connectionRetryInterval (p. 437)": integer,
    "fileCacheDuration (p. 437)": integer,
    "certificateMode (p. 437)": enum,
    "inputLossAction (p. 437)": enum,
    "sendDelayMs (p. 437)": integer,
    "eventIdMode (p. 438)": enum,
    "restartDelay (p. 438)": integer,
    "streamManifestBehavior (p. 438)": enum
"hlsGroupSettings (p. 441)": {
  "segmentsPerSubdirectory (p. 406)": integer,
  "ivInManifest (p. 406)": enum,
  "outputSelection (p. 407)": enum,
  "encryptionType (p. 407)": enum,
  "destination (p. 407)": {
    "destinationRefId (p. 441)": "string"
  },
  "segmentsPerSubdirectory (p. 406)": integer,
  "ivInManifest (p. 406)": enum,
  "outputSelection (p. 407)": enum,
  "encryptionType (p. 407)": enum,
  "destination (p. 407)": {
    "destinationRefId (p. 441)": "string"
  },
  "indexNSegments (p. 407)": integer,
  "timedMetadataId3Frame (p. 407)": enum,
  "constantIv (p. 407)": "string",
  "baseUrlManifest (p. 407)": "string",
  "captionLanguageSetting (p. 408)": enum,
  "minSegmentLength (p. 408)": integer,
  "mode (p. 408)": enum,
  "keyProviderSettings (p. 408)": {
    "staticKeySettings (p. 421)": {
      "staticKeyValue (p. 446)": "string",
      "keyProviderServer (p. 447)": {
        "passwordParam (p. 418)": "string",
        "uri (p. 418)": "string",
        "username (p. 418)": "string"
      }
    },
    "keyFormat (p. 411)": "string",
    "inputLossAction (p. 411)": enum,
    "programDateTime (p. 410)": enum,
    "directoryStructure (p. 410)": enum,
    "adMarkers (p. 411)": enum,
    "programDateTimePeriod (p. 411)": integer,
    "segmentLength (p. 411)": integer,
    "hlsCdnSettings (p. 411)": {
      "hlsAkamaiSettings (p. 405)": {
        "httpTransferMode (p. 403)": enum,
        "salt (p. 403)": "string",
        "numRetries (p. 403)": integer,
        "restartDelay (p. 404)": integer,
        "connectionRetryInterval (p. 404)": integer,
        "filecacheDuration (p. 404)": integer,
        "token (p. 404)": "string"
      },
      "hlsWebdavSettings (p. 405)": {
        "baseUrl (p. 405)": "string",
        "uri (p. 405)": "string",
        "username (p. 405)": "string",
        "passwordParam (p. 405)": "string"
      }
    }
  }
}
"httpTransferMode (p. 416)": enum,
"numRetries (p. 416)": integer,
"restartDelay (p. 416)": integer,
"connectionRetryInterval (p. 416)": integer,
"filecacheDuration (p. 416)": integer
},
"hlsBasicPutSettings (p. 405)": {
"numRetries (p. 404)": integer,
"restartDelay (p. 404)": integer,
"connectionRetryInterval (p. 405)": integer,
"filecacheDuration (p. 405)": integer
},
"hlsMediaStoreSettings (p. 406)": {
"mediaStoreStorageClass (p. 413)": enum,
"numRetries (p. 413)": integer,
"restartDelay (p. 413)": integer,
"connectionRetryInterval (p. 413)": integer,
"filecacheDuration (p. 413)": integer
}
},
"name (p. 440)": "string"
],
"audioDescriptions (p. 391)": [
{
"languageCodeControl (p. 363)": enum,
"audioTypeControl (p. 363)": enum,
"remixSettings (p. 363)": {
"channelMappings (p. 442)": [ 
{
"outputChannel (p. 362)": integer,
"inputChannelLevels (p. 362)": [ 
{
"inputChannel (p. 417)": integer,
"gain (p. 417)": integer
}
] 
},
"channelsOut (p. 442)": integer,
"channelsIn (p. 442)": integer
},
"audioType (p. 363)": enum,
"name (p. 363)": "string",
"codecSettings (p. 363)": {
"ac3Settings (p. 362)": {
"drcProfile (p. 359)": enum,
"dialnorm (p. 359)": integer,
"codingMode (p. 360)": enum,
"metadataControl (p. 360)": enum,
"bitrate (p. 360)": number,
"lfeFilter (p. 360)": enum,
"bitstreamMode (p. 360)": enum
},
"aacSettings (p. 362)": {
"vbrQuality (p. 357)": enum,
"codingMode (p. 357)": enum,
"profile (p. 357)": enum,
"inputType (p. 357)": enum,
"bitrate (p. 357)": number,
"rawFormat (p. 357)": enum,
"rateControlMode (p. 358)": enum,
"sampleRate (p. 358)": number,
"spec (p. 358)": enum
}
"eac3Settings (p. 362)": {
  "dialnorm (p. 387)": integer,
  "passthroughControl (p. 387)": enum,
  "drcLine (p. 387)": enum,
  "metadataControl (p. 387)": enum,
  "bitrate (p. 387)": number,
  "ltRtSurroundMixLevel (p. 388)": number,
  "surroundExMode (p. 388)": enum,
  "lfeControl (p. 388)": enum,
  "codingMode (p. 388)": enum,
  "surroundMode (p. 388)": enum,
  "attenuationControl (p. 388)": enum,
  "lfeFilter (p. 388)": enum,
  "dcFilter (p. 388)": enum,
  "phaseControl (p. 389)": enum,
  "ltRtCenterMixLevel (p. 389)": number,
  "stereoDownmix (p. 389)": enum,
  "bitstreamMode (p. 389)": enum,
  "loRoSurroundMixLevel (p. 389)": number,
  "drcRf (p. 389)": enum,
  "loRoCenterMixLevel (p. 389)": number
},
"passThroughSettings (p. 362)": {
},
"mp2Settings (p. 362)": {
  "codingMode (p. 435)": enum,
  "bitrate (p. 435)": number,
  "sampleRate (p. 435)": number
},
"languageCode (p. 363)": "string",
"streamName (p. 364)": "string",
"audioNormalizationSettings (p. 364)": {
  "targetLkfs (p. 365)": number,
  "algorithmControl (p. 365)": enum,
  "algorithm (p. 366)": enum
},
"audioSelectorName (p. 364)": "string"
},
"captionDescriptions (p. 391)": [
  {
    "captionSelectorName (p. 374)": "string",
    "languageDescription (p. 374)": "string",
    "name (p. 374)": "string",
    "destinationSettings (p. 374)": {
      "scte27DestinationSettings (p. 374)": {
      },
      "burnInDestinationSettings (p. 375)": {
        "xPosition (p. 370)": integer,
        "backgroundColor (p. 370)": enum,
        "yPosition (p. 370)": integer,
        "teletextGridControl (p. 371)": enum,
        "backgroundOpacity (p. 371)": integer,
        "fontOpacity (p. 371)": integer,
        "shadowOpacity (p. 371)": integer,
        "fontResolution (p. 371)": integer,
        "shadowYOffset (p. 371)": integer,
        "outlineSize (p. 372)": integer,
        "outlineColor (p. 372)": enum,
        "fontSize (p. 372)": "string",
        "shadowXOffset (p. 372)": integer,
        "alignment (p. 372)": enum,
        "shadowColor (p. 372)": enum,
        "fontColor (p. 373)": enum,
      }
    }
  }
]
"font (p. 373)": {
  "passwordParam (p. 418)": "string",
  "uri (p. 418)": "string",
  "username (p. 418)": "string"
},
"teletextDestinationSettings (p. 375)": {
},
"webvttDestinationSettings (p. 375)": {
},
"ttmlDestinationSettings (p. 375)": {
  "styleControl (p. 448)": enum
},
"smpteTtDestinationSettings (p. 375)": {
},
"embeddedPlusScte20DestinationSettings (p. 375)": {
},
"dvbSubDestinationSettings (p. 375)": {
  "xPosition (p. 381)": integer,
  "backgroundColor (p. 381)": enum,
  "yPosition (p. 382)": integer,
  "teletextGridControl (p. 382)": enum,
  "backgroundOpacity (p. 382)": integer,
  "fontOpacity (p. 382)": integer,
  "shadowOpacity (p. 382)": integer,
  "fontResolution (p. 383)": integer,
  "shadowXOffset (p. 383)": integer,
  "outlineSize (p. 383)": integer,
  "outlineColor (p. 383)": enum,
  "fontSize (p. 383)": "string",
  "shadowYOffset (p. 383)": integer,
  "outlineColor (p. 383)": enum,
  "shadowColor (p. 384)": enum,
  "fontColor (p. 384)": enum,
  "font (p. 384)": {
    "passwordParam (p. 418)": "string",
    "uri (p. 418)": "string",
    "username (p. 418)": "string"
  }
},
"embeddedDestinationSettings (p. 375)": {
},
"aribDestinationSettings (p. 375)": {
},
"scte20PlusEmbeddedDestinationSettings (p. 375)": {
},
"languageCode (p. 374)": "string"
],
"availConfiguration (p. 392)": {
  "availSettings (p. 368)": {
    "scte35TimeSignalApos (p. 368)": {
      "adAvailOffset (p. 444)": integer,
      "webDeliveryAllowedFlag (p. 445)": enum,
      "noRegionalBlackoutFlag (p. 445)": enum
    },
    "scte35SpliceInsert (p. 368)": {
      "adAvailOffset (p. 444)": integer,
      "webDeliveryAllowedFlag (p. 444)": enum,
      "noRegionalBlackoutFlag (p. 444)": enum
    }
  },
  "globalConfiguration (p. 392)": {
    "inputLossBehavior (p. 393)": {
"inputLossImageType (p. 419)" : enum,
"inputLossImageColor (p. 419)" : "string",
"inputLossImageSlate (p. 419)" : {
  "passwordParam (p. 418)" : "string",
  "uri (p. 418)" : "string",
  "username (p. 418)" : "string"
},
"repeatFrameMsec (p. 419)" : integer,
"blackFrameMsec (p. 419)" : integer
},
"supportLowFramerateInputs (p. 393)" : enum,
"initialAudioGain (p. 394)" : integer,
"inputEndAction (p. 394)" : enum,
"outputTimingSource (p. 394)" : enum
},
"videoDescriptions (p. 392)" : [
  {
  "respondToAfd (p. 449)" : enum,
  "scalingBehavior (p. 450)" : enum,
  "name (p. 450)" : "string",
  "width (p. 450)" : integer,
  "sharpness (p. 450)" : integer,
  "codecSettings (p. 450)" : {
    "h264Settings (p. 449)" : {
      "minIInterval (p. 397)" : integer,
      "slices (p. 397)" : integer,
      "parNumerator (p. 397)" : integer,
      "gopSizeUnits (p. 398)" : enum,
      "maxBitrate (p. 398)" : integer,
      "bitrate (p. 398)" : integer,
      "bufFillPct (p. 398)" : integer,
      "temporalAq (p. 398)" : enum,
      "afdSignaling (p. 398)" : enum,
      "timecodeInsertion (p. 399)" : enum,
      "bufSize (p. 399)" : integer,
      "softness (p. 399)" : integer,
      "framerateControl (p. 399)" : enum,
      "fixedAfd (p. 399)" : enum,
      "level (p. 399)" : enum,
      "lookAheadRateControl (p. 399)" : enum,
      "profile (p. 400)" : enum,
      "framerateNumerator (p. 400)" : integer,
      "gopClosedCadence (p. 400)" : integer,
      "framerateDenominator (p. 400)" : integer,
      "spatialAq (p. 400)" : enum,
      "entropyEncoding (p. 400)" : enum,
      "adaptiveQuantization (p. 400)" : enum,
      "colorMetadata (p. 401)" : enum,
      "gopSize (p. 401)" : number,
      "numRefFrames (p. 401)" : integer,
      "gopBReference (p. 401)" : enum,
      "syntax (p. 401)" : enum,
      "parControl (p. 401)" : enum,
      "parDenominator (p. 401)" : integer,
      "sceneChangeDetect (p. 402)" : enum,
      "scanType (p. 402)" : enum,
      "gopNumBFrames (p. 402)" : integer,
      "flickerAq (p. 402)" : enum,
      "flickerAq (p. 402)" : enum,
      "rateControlMode (p. 402)" : enum
    }
  },
  "height (p. 450)" : integer
}
],
"availBlanking (p. 392)" : {
  "state (p. 368)" : enum,
"availBlankingImage (p. 368)": {
    "passwordParam (p. 418)": "string",
    "uri (p. 418)": "string",
    "username (p. 418)": "string"
},
"blackoutSlate (p. 392)": {
    "networkEndBlackoutImage (p. 368)": {
        "passwordParam (p. 418)": "string",
        "uri (p. 418)": "string",
        "username (p. 418)": "string"
    },
    "networkEndBlackout (p. 369)": enum,
    "networkId (p. 369)": "string",
    "blackoutSlateImage (p. 369)": {
        "passwordParam (p. 418)": "string",
        "uri (p. 418)": "string",
        "username (p. 418)": "string"
    },
    "state (p. 369)": enum
},
"id (p. 378)": "string",
"pipelinesRunningCount (p. 378)": integer,
"state (p. 378)": enum,
"arn (p. 378)": "string",
"egressEndpoints (p. 378)": [
    {
        "sourceIp (p. 378)": "string"
    }
]

Example InvalidRequest

{
    "message (p. 421)": "string"
}

Example AccessDenied

{
    "message (p. 360)": "string"
}

Example ResourceNotFound

{
    "message (p. 442)": "string"
}

Example ResourceConflict

{
    "message (p. 442)": "string"
}
Example LimitExceeded

```json
{
  "message (p. 421)": "string"
}
```

Example InternalServiceError

```json
{
  "message (p. 421)": "string"
}
```

Properties

**AacCodingMode (Enum)**

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.

- AD_RECEIVER_MIX
- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_5_1

**AacInputType (Enum)**

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.

- BROADCASTER_MIXED_AD
- NORMAL

**AacProfile (Enum)**

AAC Profile.

- HEV1
- HEV2
- LC

**AacRateControlMode (Enum)**

Rate Control Mode.

- CBR
- VBR
**AacRawFormat (Enum)**

Sets LATM / LOAS AAC output for raw containers.

- LATM, LOAS
- NONE

**AacSettings**

**vbrQuality**

VBR Quality Level - Only used if rateControlMode is VBR.

- **Type**: AacVbrQuality (p. 358)
- **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. 356)
- **Required**: False

**profile**

AAC Profile.

- **Type**: AacProfile (p. 356)
- **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. 356)
- **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.
**Properties**

*Type:* AacRawFormat (p. 357)  
*Required:* False

**rateControlMode**

Rate Control Mode.

*Type:* AacRateControlMode (p. 356)  
*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. 358)  
*Required:* False

**AacSpec (Enum)**

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

- MPEG2
- MPEG4

**AacVbrQuality (Enum)**

VBR Quality Level - Only used if rateControlMode is VBR.

- HIGH
- LOW
- MEDIUM_HIGH
- MEDIUM_LOW

**Ac3BitstreamMode (Enum)**

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

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

**Ac3CodingMode (Enum)**

Dolby Digital coding mode. Determines number of channels.

- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_3_2_LFE

**Ac3DrcProfile (Enum)**

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

- FILM_STANDARD
- NONE

**Ac3LfeFilter (Enum)**

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

- DISABLED
- ENABLED

**Ac3MetadataControl (Enum)**

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.

- FOLLOW_INPUT
- USE_CONFIGURED

**Ac3Settings**

**drcProfile**

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

- Type: Ac3DrcProfile (p. 359)
- 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. 359)
  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. 359)
  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. 359)
  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. 358)
  Required: False

AccessDenied

message

  Type: string
  Required: False

AfdSignaling (Enum)
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.
ArchiveContainerSettings

m2tsSettings

Type: M2tsSettings (p. 424)
Required: False

ArchiveGroupSettings

destination

A directory and base filename where archive files should be written. If the base filename portion of the URI is left blank, the base filename of the first input will be automatically inserted.

Type: OutputLocationRef (p. 441)
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

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. 361)
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

AribSourceSettings

AudioChannelMapping

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. 417)
Required: True

AudioCodecSettings

ac3Settings

Type: Ac3Settings (p. 359)
Required: False

aacSettings

Type: AacSettings (p. 357)
Required: False

eac3Settings

Type: Eac3Settings (p. 387)
Required: False

passThroughSettings

Type: PassThroughSettings (p. 442)
Required: False

mp2Settings

Type: Mp2Settings (p. 435)
Required: False
AudioDescription

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. 364)
    Required: False

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. 364)
    Required: False

remixSettings

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

    Type: RemixSettings (p. 442)
    Required: False

audioType

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

    Type: AudioType (p. 367)
    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. 362)
    Required: True

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.
**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. 365)
- **Required**: False

**audioSelectorName**

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

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

**AudioDescriptionAudioTypeControl (Enum)**

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.

- FOLLOW_INPUT
- USE_CONFIGURED

**AudioDescriptionLanguageCodeControl (Enum)**

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.

- FOLLOW_INPUT
- USE_CONFIGURED

**AudioLanguageSelection**

**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. 365)
**Properties**

**Required**: False

**languageCode**

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

*Type*: string  
*Required*: True

**AudioLanguageSelectionPolicy (Enum)**

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.

- LOOSE
- STRICT

**AudioNormalizationAlgorithm (Enum)**

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

- ITU_1770_1
- ITU_1770_2

**AudioNormalizationAlgorithmControl (Enum)**

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.

- CORRECT_AUDIO

**AudioNormalizationSettings**

**targetLkfs**

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

*Type*: number  
*Required*: False  
*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. 365)  
*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. 365)
**Required:** False

AudioOnlyHlsSettings

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. 366)
**Required:** False

audioGroupId

Specifies the group to which the audio Rendition belongs.

**Type:** string
**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. 418)
**Required:** False

AudioOnlyHlsTrackType (Enum)

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

ALTERNATE_AUDIO_AUTO_SELECT
**ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT**
**ALTERNATE_AUDIO_NOT_AUTO_SELECT**
**AUDIO_ONLY_VARIANT_STREAM**

### AudioPidSelection

**pid**

Selects a specific PID from within a source.

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

### AudioSelector

**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

**selectorSettings**

The audio selector settings.

- **Type**: AudioSelectorSettings (p. 367)
- **Required**: False

### AudioSelectorSettings

**audioLanguageSelection**

- **Type**: AudioLanguageSelection (p. 364)
- **Required**: False

**audioPidSelection**

- **Type**: AudioPidSelection (p. 367)
- **Required**: False

### AudioType (Enum)

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

- CLEAN_EFFECTS
- HEARING_IMPAIRED
- UNDEFINED
**AvailBlanking**

*state*

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

  * **Type**: AvailBlankingState (p. 368)
  * **Required**: False

**availBlankingImage**

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

  * **Type**: InputLocation (p. 418)
  * **Required**: False

**AvailBlankingState (Enum)**

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

  * DISABLED
  * ENABLED

**AvailConfiguration**

*availSettings*

Ad avail settings.

  * **Type**: AvailSettings (p. 368)
  * **Required**: False

**AvailSettings**

*scte35TimeSignalApos*

  * **Type**: Scte35TimeSignalApos (p. 444)
  * **Required**: False

*scte35SpliceInsert*

  * **Type**: Scte35SpliceInsert (p. 444)
  * **Required**: False

**BlackoutSlate**

*networkEndBlackoutImage*

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

**Type:** InputLocation (p. 418)  
**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. 369)  
**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

**blackoutSlateImage**

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

**Type:** InputLocation (p. 418)  
**Required:** False

**state**

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

**Type:** BlackoutSlateState (p. 369)  
**Required:** False

**BlackoutSlateNetworkEndBlackout (Enum)**

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".

- DISABLED
- ENABLED

**BlackoutSlateState (Enum)**

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

- DISABLED
ENABLED

**BurnInAlignment (Enum)**

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.

- CENTERED
- LEFT
- SMART

**BurnInBackgroundColor (Enum)**

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

- BLACK
- NONE
- WHITE

**BurnInDestinationSettings**

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit 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. 370)
- **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. 374)
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

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

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

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
**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. 373)
- **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

**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

**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. 370)
- **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. 373)
**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. 373)
**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. 418)
**Required**: False

**BurnInFontColor (Enum)**

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

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

**BurnInOutlineColor (Enum)**

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

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

**BurnInShadowColor (Enum)**

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

BLACK
NONE
WHITE

**BurnInTeletextGridControl (Enum)**
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.

FIXED
SCALED

**CaptionDescription**

**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

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

_Type:_ CaptionDestinationSettings (p. 374)  
_Required:_ False

**languageCode**

_Type:_ string  
_Required:_ False

**CaptionDestinationSettings**

**scte27DestinationSettings**

_Type:_ Scte27DestinationSettings (p. 443)
Required: False

**burnInDestinationSettings**

Type: BurnInDestinationSettings (p. 370)
Required: False

**teletextDestinationSettings**

Type: TeletextDestinationSettings (p. 447)
Required: False

**webvttDestinationSettings**

Type: WebvttDestinationSettings (p. 453)
Required: False

**ttmlDestinationSettings**

Type: TtmlDestinationSettings (p. 448)
Required: False

**smpteTtDestinationSettings**

Type: SmpteTtDestinationSettings (p. 446)
Required: False

**embeddedPlusScte20DestinationSettings**

Type: EmbeddedPlusScte20DestinationSettings (p. 390)
Required: False

**dvbSubDestinationSettings**

Type: DvbSubDestinationSettings (p. 381)
Required: False

**embeddedDestinationSettings**

Type: EmbeddedDestinationSettings (p. 390)
Required: False

**aribDestinationSettings**

Type: AribDestinationSettings (p. 362)
Required: False

**scte20PlusEmbeddedDestinationSettings**

Type: Scte20PlusEmbeddedDestinationSettings (p. 443)
Required: False
**CaptionLanguageMapping**

**languageDescription**

Textual description of language

*Type:* string  
*Required:* False

**captionChannel**

Channel to insert closed captions. Each channel mapping must have a unique channel number (maximum of 4)

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

**languageCode**

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

*Type:* string  
*Required:* False

**CaptionSelector**

**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

**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. 377)  
*Required:* False
CaptionSelectorSettings

**embeddedSourceSettings**

Type: EmbeddedSourceSettings (p. 390)
Required: False

**scte20SourceSettings**

Type: Scte20SourceSettings (p. 443)
Required: False

**dvbSubSourceSettings**

Type: DvbSubSourceSettings (p. 385)
Required: False

**teletextSourceSettings**

Type: TeletextSourceSettings (p. 447)
Required: False

**aribSourceSettings**

Type: AribSourceSettings (p. 362)
Required: False

**scte27SourceSettings**

Type: Scte27SourceSettings (p. 443)
Required: False

Channel

**inputAttachments**

List of input attachments for channel.

Type: Array of type InputAttachment (p. 416)
Required: False

**roleArn**

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

Type: string
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.
Properties

Type: Array of type OutputDestination (p. 439)
Required: False

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

coderSettings
Type: EncoderSettings (p. 391)
Required: False

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

pipelinesRunningCount
The number of currently healthy pipelines.
Type: integer
Required: False

state
Type: ChannelState (p. 379)
Required: False

arn
The unique arn of the channel.
Type: string
Required: False

egressEndpoints
The endpoints where outgoing connections initiate from
Type: Array of type ChannelEgressEndpoint (p. 378)
Required: False

ChannelEgressEndpoint

sourceIp
Public IP of where a channel's output comes from
**Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>string</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>ChannelState</td>
<td>Enum</td>
<td></td>
<td>CREATING, CREATE_FAILED, IDLE, STARTING, RUNNING, RECOVERING, STOPPING,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DELETING, DELETED</td>
</tr>
<tr>
<td>DvbNitSettings</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>networkName</td>
<td>string</td>
<td>True</td>
<td>The network name text placed in the networkNameDescriptor inside the Network Information Table. Maximum length is 256 characters.</td>
</tr>
<tr>
<td>networkId</td>
<td>integer</td>
<td>True</td>
<td>The numeric value placed in the Network Information Table (NIT).</td>
</tr>
<tr>
<td>repInterval</td>
<td>integer</td>
<td>False</td>
<td>The number of milliseconds between instances of this table in the output transport stream.</td>
</tr>
<tr>
<td>DvbSdtOutputSdt</td>
<td>Enum</td>
<td></td>
<td>SDT_FOLLOW, SDT_FOLLOW_IF_PRESENT, SDT_MANUAL, SDT_NONE</td>
</tr>
</tbody>
</table>

**ChannelState (Enum)**

- CREATING
- CREATE_FAILED
- IDLE
- STARTING
- RUNNING
- RECOVERING
- STOPPING
- DELETING
- DELETED

**DvbNitSettings**

**networkName**

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

**networkId**

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

**repInterval**

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

**DvbSdtOutputSdt (Enum)**

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.

- SDT_FOLLOW
- SDT_FOLLOW_IF_PRESENT
Properties

**DvbSdtSettings**

**serviceName**

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

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

**serviceProviderName**

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

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

**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. 379)
- **Required:** False

**DvbSubDestinationAlignment (Enum)**

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.

- CENTERED
- LEFT
- SMART
DvbSubDestinationBackgroundColor (Enum)

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

- BLACK
- NONE
- WHITE

DvbSubDestinationFontColor (Enum)

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

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

DvbSubDestinationOutlineColor (Enum)

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

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

DvbSubDestinationSettings

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit 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.
Properties

**Type:** DvbSubDestinationBackgroundColor (p. 381)
**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. 384)
**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

**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
- **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

- **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. 381)  
  **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

- **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
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. 380)
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. 384)
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. 381)
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. 418)
Required: False

DvbSubDestinationShadowColor (Enum)

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

BLACK
NONE
WHITE

DvbSubDestinationTeletextGridControl (Enum)

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.

FIXED
**SCALED**

### DvbSubSourceSettings

**pid**

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

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

### DvbTdtSettings

**repInterval**

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

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

### Eac3AttenuationControl (Enum)

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

- ATTENUATE_3_DB
- NONE

### Eac3BitstreamMode (Enum)

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

- COMMENTARY
- COMPLETE_MAIN
- EMERGENCY
- HEARING_IMPAIRED
- VISUALLY_IMPAIRED

### Eac3CodingMode (Enum)

Dolby Digital Plus coding mode. Determines number of channels.

- CODING_MODE_1_0
- CODING_MODE_2_0
- CODING_MODE_3_2

### Eac3DcFilter (Enum)

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

**Eac3DrcLine (Enum)**

Sets the Dolby dynamic range compression profile.

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

**Eac3DrcRf (Enum)**

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

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

**Eac3LfeControl (Enum)**

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

- LFE
- NO_LFE

**Eac3LfeFilter (Enum)**

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

- DISABLED
- ENABLED

**Eac3MetadataControl (Enum)**

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.

- FOLLOW_INPUT
- USE_CONFIGURED

**Eac3PassthroughControl (Enum)**

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 transcoding. Inputs that alternate between DD+ and non-DD+ content will have a consistent DD+ output as the system alternates between passthrough and encoding.
NO_PASSTHROUGH
WHEN_POSSIBLE

Eac3PhaseControl (Enum)
When set to shift90Degrees, applies a 90-degree phase shift to the surround channels. Only used for 3/2
coding mode.

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. 386)
Required: False

drcLine
Sets the Dolby dynamic range compression profile.

Type: Eac3DrcLine (p. 386)
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. 386)
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. 390)
Required: False

lfeControl
When encoding 3/2 audio, setting to lfe enables the LFE channel
Type: Eac3LfeControl (p. 386)
Required: False

codingMode
Dolby Digital Plus coding mode. Determines number of channels.
Type: Eac3CodingMode (p. 385)
Required: False

surroundMode
When encoding 2/0 audio, sets whether Dolby Surround is matrix encoded into the two channels.
Type: Eac3SurroundMode (p. 390)
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. 385)
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. 386)
Required: False

dcFilter
When set to enabled, activates a DC highpass filter for all input channels.
**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. 387)*
*Required: False*

**ltRtCenterMixLevel**

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

*Type: number*
*Required: False*

**stereoDownmix**

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

*Type: Eac3StereoDownmix (p. 390)*
*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. 385)*
*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. 386)*
*Required: False*

**loRoCenterMixLevel**

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

*Type: number*
Required: False

Eac3StereoDownmix (Enum)
Stereo downmix preference. Only used for 3/2 coding mode.

- DPL2
- LO_RO
- LT_RT
- NOT_INDICATED

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

- DISABLED
- ENABLED
- NOT_INDICATED

Eac3SurroundMode (Enum)
When encoding 2/0 audio, sets whether Dolby Surround is matrix encoded into the two channels.

- DISABLED
- ENABLED
- NOT_INDICATED

EmbeddedConvert608To708 (Enum)
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.

- DISABLED
- UPCONVERT

EmbeddedDestinationSettings
EmbeddedPlusScte20DestinationSettings

EmbeddedScte20Detection (Enum)
Set to "auto" to handle streams with intermittent and/or non-aligned SCTE-20 and Embedded captions.

- AUTO
- OFF

EmbeddedSourceSettings

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

Type: EmbeddedScte20Detection (p. 390)
**Properties**

**source608ChannelNumber**
Specifications 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. 390)
- **Required:** False

**source608TrackNumber**
This field is unused and deprecated.

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

**EncoderSettings**

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

- **Type:** TimecodeConfig (p. 447)
- **Required:** True

**outputGroups**

- **Type:** Array of type OutputGroup (p. 440)
- **Required:** True

**audioDescriptions**

- **Type:** Array of type AudioDescription (p. 363)
- **Required:** True

**captionDescriptions**
Settings for caption descriptions

- **Type:** Array of type CaptionDescription (p. 374)
Properties

Required: False

availConfiguration

Event-wide configuration settings for ad avail insertion.

Type: AvailConfiguration (p. 368)
Required: False

globalConfiguration

Configuration settings that apply to the event as a whole.

Type: GlobalConfiguration (p. 393)
Required: False

videoDescriptions

Type: Array of type VideoDescription (p. 449)
Required: True

availBlanking

Settings for ad avail blanking.

Type: AvailBlanking (p. 368)
Required: False

blackoutSlate

Settings for blackout slate.

Type: BlackoutSlate (p. 368)
Required: False

FecOutputIncludeFec (Enum)

Enables column only or column and row based FEC

COLUMN
COLUMN_AND_ROW

FecOutputSettings

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
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. 392)
Required: False

FixedAfd (Enum)

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

AFD_0000
AFD_0010
AFD_0011
AFD_0100
AFD_1000
AFD_1001
AFD_1010
AFD_1011
AFD_1101
AFD_1110
AFD_1111

GlobalConfiguration

inputLossBehavior

Settings for system actions when input is lost.

Type: InputLossBehavior (p. 419)
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. 394)
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 an input completes (e.g. end-of-file.) Options include immediately switching to the next sequential input (via "switchInput"), switching to the next input and looping back to the first input when last input ends (via "switchAndLoopInputs") or not switching inputs and instead transcoding black / color / slate images per the "Input Loss Behavior" configuration until an activateInput REST command is received (via "none").

- **Type:** GlobalConfigurationInputEndAction (p. 394)
- **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. 394)
- **Required:** False

GlobalConfigurationInputEndAction (Enum)

Indicates the action to take when an input completes (e.g. end-of-file.) Options include immediately switching to the next sequential input (via "switchInput"), switching to the next input and looping back to the first input when last input ends (via "switchAndLoopInputs") or not switching inputs and instead transcoding black / color / slate images per the "Input Loss Behavior" configuration until an activateInput REST command is received (via "none").

- **NONE**
- **SWITCH_AND_LOOP_INPUTS**

GlobalConfigurationLowFramerateInputs (Enum)

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.

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

GlobalConfigurationOutputTimingSource (Enum)

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.

- **INPUT_CLOCK**
SYSTEM_CLOCK

**H264AdaptiveQuantization (Enum)**
Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.
- HIGH
- HIGHER
- LOW
- MAX
- MEDIUM
- OFF

**H264ColorMetadata (Enum)**
Includes colorspace metadata in the output.
- IGNORE
- INSERT

**H264EntropyEncoding (Enum)**
Entropy encoding mode. Use cabac (must be in Main or High profile) or cavlc.
- CABAC
- CAVLC

**H264FlickerAq (Enum)**
If set to enabled, adjust quantization within each frame to reduce flicker or 'pop' on I-frames.
- DISABLED
- ENABLED

**H264FramerateControl (Enum)**
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.
- INITIALIZE_FROM_SOURCE
- SPECIFIED

**H264GopBReference (Enum)**
If enabled, use reference B frames for GOP structures that have B frames > 1.
- DISABLED
- ENABLED

**H264GopSizeUnits (Enum)**
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.
FRAMES
SECONDS

**H264Level (Enum)**

H.264 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 (Enum)**

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

- HIGH
- LOW
- MEDIUM

**H264ParControl (Enum)**

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.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**H264Profile (Enum)**

H.264 Profile.

- BASELINE
- HIGH
- HIGH_10BIT
- HIGH_422
- HIGH_422_10BIT
H264RateControlMode (Enum)
Rate control mode.
- CBR
- VBR

H264ScanType (Enum)
Sets the scan type of the output to progressive or top-field-first interlaced.
- INTERLACED
- PROGRESSIVE

H264SceneChangeDetect (Enum)
Scene change detection. Inserts I-frames on scene changes when enabled.
- DISABLED
- ENABLED

H264Settings

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. 395)
Required: False

**maxBitrate**

Maximum bitrate in bits/second (for VBR mode only).

Type: integer
Required: False
Minimum: 1000

**bitrate**

Average bitrate in bits/second. Required for VBR, CBR, and ABR. For MS Smooth outputs, bitrates must be unique when 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. 403)
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. 360)
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. 403)
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. 395)
Required: False

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. 393)
Required: False

level

H.264 Level.

Type: H264Level (p. 396)
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.
Properties

**Type**: H264LookAheadRateControl (p. 396)
**Required**: False

**profile**
H.264 Profile.
**Type**: H264Profile (p. 396)
**Required**: False

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

**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

**framerateDenominator**
Framerate denominator.
**Type**: integer
**Required**: False

**spatialAq**
If set to enabled, adjust quantization within each frame based on spatial variation of content complexity.
**Type**: H264SpatialAq (p. 402)
**Required**: False

**entropyEncoding**
Entropy encoding mode. Use cabac (must be in Main or High profile) or cavlc.
**Type**: H264EntropyEncoding (p. 395)
**Required**: False

**adaptiveQuantization**
Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.
**Type**: H264AdaptiveQuantization (p. 395)
**Required**: False

**colorMetadata**
Includes colorspace metadata in the output.

  * **Type**: H264ColorMetadata (p. 395)
  * **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. 395)
  * **Required**: False

**syntax**
Produces a bitstream compliant with SMPTE RP-2027.

  * **Type**: H264Syntax (p. 402)
  * **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. 396)
  * **Required**: False

**parDenominator**
Pixel Aspect Ratio denominator.
sceneChangeDetect

Scene change detection. Inserts I-frames on scene changes when enabled.

- **Type**: `H264SceneChangeDetect (p. 397)`
- **Required**: False

scanType

Sets the scan type of the output to progressive or top-field-first interlaced.

- **Type**: `H264ScanType (p. 397)`
- **Required**: False

gopNumBFrames

Number of B-frames between reference frames.

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

flickerAq

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

- **Type**: `H264FlickerAq (p. 395)`
- **Required**: False

rateControlMode

Rate control mode.

- **Type**: `H264RateControlMode (p. 397)`
- **Required**: False

H264SpatialAq (Enum)

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

- `DISABLED`
- `ENABLED`

H264Syntax (Enum)

Produces a bitstream compliant with SMPTE RP-2027.

- `DEFAULT`
RP2027

H264TemporalAq (Enum)
If set to enabled, adjust quantization within each frame based on temporal variation of content complexity.

DISABLED
ENABLED

H264TimecodeInsertionBehavior (Enum)
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

DISABLED
PIC_TIMING_SEI

HlsAdMarkers (Enum)

ADOBE
ELEMENTAL
ELEMENTAL_SCTE35

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

CHUNKED
NON_CHUNKED

HlsAkamaiSettings

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

Type: HlsAkamaiHttpTransferMode (p. 403)
Required: False

salt
Salt for authenticated Akamai.

Type: string
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**

**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.
AWS Elemental MediaLive API Reference
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 (Enum)**

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

```
INSERT
NONE
OMIT
```

**HlsCdnSettings**

**hlsAkamaiSettings**

**Type**: HlsAkamaiSettings (p. 403)
**Required**: False

**hlsWebdavSettings**

**Type**: HlsWebdavSettings (p. 416)
**Required**: False

**hlsBasicPutSettings**

**Type**: HlsBasicPutSettings (p. 404)
**Required**: False
hlsMediaStoreSettings

Type: HlsMediaStoreSettings (p. 413)
Required: False

HlsClientCache (Enum)
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.

DISABLED
ENABLED

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

RFC_4281
RFC_6381

HlsDirectoryStructure (Enum)
Place segments in subdirectories.

SINGLE_DIRECTORY
SUBDIRECTORY_PER_STREAM

HlsEncryptionType (Enum)
Encrypts the segments with the given encryption scheme. Exclude this parameter if no encryption is desired.

AES128
SAMPLE_AES

HlsGroupSettings

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. 412)
### Required: False

#### outputSelection

Generates the .m3u8 playlist file for this HLS output group. The segmentsOnly option will output segments without the .m3u8 file.

- **Type**: HlsOutputSelection (p. 414)
- **Required**: False

#### encryptionType

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

- **Type**: HlsEncryptionType (p. 406)
- **Required**: False

#### destination

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

- **Type**: OutputLocationRef (p. 441)
- **Required**: True

#### indexNSegments

Number of segments to keep in the playlist (.m3u8) file. mode must be “vod” for this setting to have an effect, and this number should be less than or equal to keepSegments.

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

#### timedMetadataId3Frame

Indicates ID3 frame that has the timecode.

- **Type**: HlsTimedMetadataId3Frame (p. 415)
- **Required**: False

#### 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

#### 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. 405)
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 set to "vod", keeps and indexes all segments starting with the first segment. If set to "live" segments will age out and only the last keepSegments number of segments will be retained.

Type: HlsMode (p. 414)
Required: False

keyProviderSettings

The key provider settings.

Type: KeyProviderSettings (p. 421)
Required: False

manifestCompression

When set to gzip, compresses HLS playlist.

Type: HlsManifestCompression (p. 412)
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. 412)
**tsFileMode**

When set to "singleFile", emits the program as a single media resource (.ts) file, and uses #EXT-X-BYTERANGE tags to index segment for playback. Playback of VOD mode content during event is not guaranteed due to HTTP server caching.

- **Type**: HlsTsFileMode (p. 415)
- **Required**: False

**manifestDurationFormat**

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

- **Type**: HlsManifestDurationFormat (p. 413)
- **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. 415)
- **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**

When set to useInputSegmentation, the output segment or fragment points are set by the RAI markers from the input streams.

- **Type**: HlsSegmentationMode (p. 415)
- **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. 376)
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. 406)
Required: False

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

Type: HlsCodecSpecification (p. 406)
Required: False

keepSegments
Number of segments to retain in the destination directory. mode must be "live" for this setting to have an effect.

Type: integer
Required: False
Minimum: 1

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. 414)
Required: False

directoryStructure
Place segments in subdirectories.

Type: HlsDirectoryStructure (p. 406)
Required: False
**Properties**

**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. 418)
- **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. 403)
- **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. 405)
- **Required:** False

**HlsInputSettings**

**retries**

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

411
Properties

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 (Enum)

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.

EXCLUDE
INCLUDE

HlsIvSource (Enum)

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.

EXPLICIT
FOLLOW_SEGMENT_NUMBER

HlsManifestCompression (Enum)

When set to gzip, compresses HLS playlist.
GZIP
NONE

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

**HlsMediaStoreSettings**

**mediaStoreStorageClass**
When set to temporal, output files are stored in non-persistent memory for faster reading and writing.
  - **Type:** HlsMediaStoreStorageClass (p. 414)
  - **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 (Enum)**

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

- TEMPORAL

**HlsMode (Enum)**

If set to "vod", keeps and indexes all segments starting with the first segment. If set to "live" segments will age out and only the last keepSegments number of segments will be retained.

- LIVE
- VOD

**HlsOutputSelection (Enum)**

Generates the .m3u8 playlist file for this HLS output group. The segmentsOnly option will output segments without the .m3u8 file.

- MANIFESTS_AND_SEGMENTS
- SEGMENTS_ONLY

**HlsOutputSettings**

**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. 415)
- Required: True

**nameModifier**

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

- Type: string
- Required: True

**HlsProgramDateTime (Enum)**

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

**HlsSegmentationMode (Enum)**

When set to useInputSegmentation, the output segment or fragment points are set by the RAI markers from the input streams.

- USE_INPUT_SEGMENTATION
- USE_SEGMENT_DURATION

**HlsSettings**

**standardHlsSettings**

*Type:* StandardHlsSettings (p. 446)  
*Required:* False

**audioOnlyHlsSettings**

*Type:* AudioOnlyHlsSettings (p. 366)  
*Required:* False

**HlsStreamInfResolution (Enum)**

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

- EXCLUDE
- INCLUDE

**HlsTimedMetadataId3Frame (Enum)**

Indicates ID3 frame that has the timecode.

- NONE
- PRIV
- TDRL

**HlsTsFileMode (Enum)**

When set to "singleFile", emits the program as a single media resource (.ts) file, and uses #EXT-X-BYTERANGE tags to index segment for playback. Playback of VOD mode content during event is not guaranteed due to HTTP server caching.

- SEGMENTED_FILES
- SINGLE_FILE

**HlsWebdavHttpTransferMode (Enum)**

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

- CHUNKED
NON_CHUNKED

HlsWebdavSettings

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

Type: HlsWebdavHttpTransferMode (p. 415)
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

InputAttachment

inputId
The ID of the input
Type: string
Required: False

inputSettings
Settings of an input (caption selector, etc.)

Type: InputSettings (p. 420)
Required: False

InputChannelLevel

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

InputDeblockFilter (Enum)
Enable or disable the deblock filter when filtering.

DISABLED
ENABLED

InputDenoiseFilter (Enum)
Enable or disable the denoise filter when filtering.

DISABLED
ENABLED

InputFilter (Enum)
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

AUTO
DISABLED
FORCED

**InputLocation**

**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 rtmpEndpoint 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 (Enum)**

Parameter that control output group behavior on input loss.

  - EMIT_OUTPUT
  - PAUSE_OUTPUT

**InputLossActionForMsSmoothOut (Enum)**

Parameter that control output group behavior on input loss.

  - EMIT_OUTPUT
  - PAUSE_OUTPUT

**InputLossActionForUdpOut (Enum)**

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.

  - DROP_PROGRAM
  - DROP_TS
InputLossBehavior

InputLossImageType
Indicates whether to substitute a solid color or a slate into the output after input loss exceeds blackFrameMsec.

Type: InputLossImageType (p. 419)
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

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

Type: InputLocation (p. 418)
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 (Enum)
Indicates whether to substitute a solid color or a slate into the output after input loss exceeds blackFrameMsec.

COLOR
SLATE

**InputSettings**

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

*Type:* InputSourceEndBehavior (p. 421)
*Required:* False

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

*Type:* Array of type AudioSelector (p. 367)
*Required:* False

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

*Type:* InputDeblockFilter (p. 417)
*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. 417)
*Required:* False

**networkInputSettings**
Input settings.

*Type:* NetworkInputSettings (p. 438)
*Required:* True

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

*Type:* VideoSelector (p. 451)
*Required:* False

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

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

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

Type: Array of type CaptionSelector (p. 376)
Required: False

denoiseFilter
Enable or disable the denoise filter when filtering.

Type: InputDenoiseFilter (p. 417)
Required: False

InputSourceEndBehavior (Enum)
Loop input if it is a file. This allows a file input to be streamed indefinitely.

CONTINUE
LOOP

InternalServiceError
message

Type: string
Required: False

InvalidRequest
message

Type: string
Required: False

KeyProviderSettings
staticKeySettings

Type: StaticKeySettings (p. 446)
Required: False

LimitExceeded
message

Type: string
Required: False
M2tsAbsentInputAudioBehavior (Enum)
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.

DROP
ENCODE_SILENCE

M2tsArib (Enum)
When set to enabled, uses ARIB-compliant field muxing and removes video descriptor.

DISABLED
ENABLED

M2tsAribCaptionsPidControl (Enum)
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.

AUTO
USE_CONFIGURED

M2tsAudioBufferModel (Enum)
When set to dvb, uses DVB buffer model for Dolby Digital audio. When set to atsc, the ATSC model is used.

ATSC
DVB

M2tsAudioInterval (Enum)
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.

VIDEO_AND_FIXED_INTERVALS
VIDEO_INTERVAL

M2tsAudioStreamType (Enum)
When set to atsc, uses stream type = 0x81 for AC3 and stream type = 0x87 for EAC3. When set to dvb, uses stream type = 0x06.

ATSC
DVB

M2tsBufferModel (Enum)
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.
MULTIPLEX
NONE

**M2tsCcDescriptor (Enum)**

When set to enabled, generates `captionServiceDescriptor` in PMT.

- DISABLED
- ENABLED

**M2tsEbifControl (Enum)**

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

- NONE
- PASSTHROUGH

**M2tsEbpPlacement (Enum)**

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.

- VIDEO_AND_AUDIO_PIDS
- VIDEO_PID

**M2tsEsRateInPes (Enum)**

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

- EXCLUDE
- INCLUDE

**M2tsKlv (Enum)**

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

- NONE
- PASSTHROUGH

**M2tsPcrControl (Enum)**

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.

- CONFIGURED_PCR_PERIOD
- PCR_EVERY_PES_PACKET

**M2tsRateMode (Enum)**

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.

- CBR
- VBR
M2tsScte35Control (Enum)

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

NONE
PASSTHROUGH

M2tsSegmentationMarkers (Enum)

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.

EBP
EBP_LEGACY
NONE
PSI_SEGSTART
RAI_ADAPT
RAI_SEGSTART

M2tsSegmentationStyle (Enum)

The segmentation style parameter controls how segmentation markers are inserted into the transport stream. With avails, it is possible that segments may be truncated, which can influence where future segmentation markers are inserted. When a segmentation style of "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.

MAINTAIN_CADENCE
RESET_CADENCE

M2tsSettings

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. 422)
Required: False

ecmPid

Packet Identifier (PID) for ECM in the transport stream. Only enabled when Simulcrypt is enabled. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0xff).

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. 422)`
- **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

**segmentationTime**

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

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

**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. 423)`
- **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. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

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

**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. 422)
  - **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

**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

**pcrPeriod**

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

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

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

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. 424)
Required: False

ebif

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

Type: M2tsEbifControl (p. 423)
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. 422)
Required: False

dvbNitSettings

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

Type: DvbNitSettings (p. 379)
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. 422)
Required: False

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

Type: M2tsTimedMetadataBehavior (p. 431)
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. 422)
Required: False

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

Type: M2tsScte35Control (p. 424)
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. 423)
- **Required:** False

arib

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

**Type:** M2tsArib (p. 422)
- **Required:** False

dvbSdtSettings

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

**Type:** DvbSdtSettings (p. 380)
- **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

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

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

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. 423)
Required: False

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

Type: M2tsEsRateInPes (p. 423)
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. 424)
Required: False

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

Type: M2tsKlv (p. 423)
Required: False

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

Type: DvbTdtSettings (p. 385)
Required: False

ccDescriptor
When set to enabled, generates captionServiceDescriptor in PMT.

Type: M2tsCcDescriptor (p. 423)
Required: False

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

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

NO_PASSTHROUGH
PASSTHROUGH
M3u8PcrControl (Enum)

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.

- CONFIGURED_PCR_PERIOD
- PCR_EVERY_PES_PACKET

M3u8Scte35Behavior (Enum)

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

- NO_PASSTHROUGH
- PASSTHROUGH

M3u8Settings

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

ThePlatform-protected transport streams using 'microsoft' as Target Client include an ECM stream. This ECM stream contains the size, IV, and PTS of every sample in the transport stream. This stream PID is specified here. This PID has no effect on non ThePlatform-protected streams.

- Type: string
- Required: False

scte35Behavior

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

- Type: M3u8Scte35Behavior (p. 432)
- 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

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

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. 432)
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

**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. A value of "0" writes out the PMT once per segment file.

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

**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

**timedMetadataBehavior**

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

Type: [M3u8TimedMetadataBehavior](#)
Required: False

**M3u8TimedMetadataBehavior (Enum)**

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

- NO_PASSTHROUGH
- PASSTHROUGH

**Mp2CodingMode (Enum)**

The MPEG2 Audio coding mode. Valid values are codingMode10 (for mono) or codingMode20 (for stereo).
CODING_MODE_1_0
CODING_MODE_2_0

Mp2Settings

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

  Type: Mp2CodingMode (p. 434)
  Required: False

bitrate
Average bitrate in bits/second.

  Type: number
  Required: False

sampleRate
Sample rate in Hz.

  Type: number
  Required: False

MsSmoothGroupSettings

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

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

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

  Type: string
  Required: False
segmentationMode

When set to useInputSegmentation, the output segment or fragment points are set by the RAI markers from the input streams.

Type: SmoothGroupSegmentationMode (p. 445)
Required: False

numRetries

Number of retry attempts.

Type: integer
Required: False
Minimum: 0

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

eventStopBehavior

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

Type: SmoothGroupEventStopBehavior (p. 445)
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. 446)
Required: False

destination

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

Type: OutputLocationRef (p. 441)
Required: True

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. 446)
Required: False
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. 445)
- **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 unless those certificates are manually added to the OS trusted keystore.

- **Type**: SmoothGroupCertificateMode (p. 445)
- **Required**: False

inputLossAction

Parameter that control output group behavior on input loss.

- **Type**: InputLossActionForMsSmoothOut (p. 418)
- **Required**: False

sendDelayMs

Outputs that are "output locked" can use this delay. Assign a delay to the output that is "secondary". Do not assign a delay to the "primary" output. The delay means that the primary output will always reach the downstream system before the secondary, which helps ensure that the downstream system always uses the primary output. (If there were no delay, the downstream system might flip-flop between whichever output happens to arrive first.) If the primary fails, the downstream system will switch to the secondary output. When the primary is restarted, the downstream system will switch back to the primary (because once again it is always arriving first)

- **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. 445)*

*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. 446)*

*Required: False*

**MsSmoothOutputSettings**

**nameModifier**

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

*Type: string*

*Required: False*

**NetworkInputServerValidation (Enum)**

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.

*CHECK_CRYPTOGRAPHY_AND_VALIDATE_NAME*

*CHECK_CRYPTOGRAPHY_ONLY*

**NetworkInputSettings**

**hlsInputSettings**

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

*Type: HlsInputSettings (p. 411)*

*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. 438)
*Required:* False

**Output**

**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

**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. 441)
*Required:* True

**audioDescriptionNames**

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

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

**OutputDestination**

**settings**

Destination settings for output; one for each redundant encoder.

*Type:* Array of type OutputDestinationSettings (p. 440)
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

url

A URL specifying a destination

  Type: string
  Required: False

username

username for destination

  Type: string
  Required: False

OutputGroup

outputs

  Type: Array of type Output (p. 439)
  Required: True

outputGroupSettings

Settings associated with the output group.

  Type: OutputGroupSettings (p. 441)
  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
AWS Elemental MediaLive API Reference
Properties

Required: False

OutputGroupSettings

archiveGroupSettings

Type: ArchiveGroupSettings (p. 361)
Required: False

udpGroupSettings

Type: UdpGroupSettings (p. 448)
Required: False

msSmoothGroupSettings

Type: MsSmoothGroupSettings (p. 435)
Required: False

hlsGroupSettings

Type: HlsGroupSettings (p. 406)
Required: False

OutputLocationRef

destinationRefId

Type: string
Required: False

OutputSettings

archiveOutputSettings

Type: ArchiveOutputSettings (p. 361)
Required: False

msSmoothOutputSettings

Type: MsSmoothOutputSettings (p. 438)
Required: False

udpOutputSettings

Type: UdpOutputSettings (p. 449)
Required: False

hlsOutputSettings

Type: HlsOutputSettings (p. 414)
**Required**: False

**PassThroughSettings**

**RemixSettings**

**channelMappings**

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

- **Type**: Array of type `AudioChannelMapping (p. 362)`
- **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

**Scte20Convert608To708 (Enum)**

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.

- **DISABLED**
- **UPCONVERT**
**Scte20PlusEmbeddedDestinationSettings**

**Scte20SourceSettings**

**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. 442)
- **Required:** False

**Scte27DestinationSettings**

**Scte27SourceSettings**

**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 (Enum)**

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

- FOLLOW
- IGNORE

**Scte35AposWebDeliveryAllowedBehavior (Enum)**

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

- FOLLOW
- IGNORE
Scte35SpliceInsert

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. 444)
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. 444)
Required: False

Scte35SpliceInsertNoRegionalBlackoutBehavior (Enum)
When set to ignore, Segment Descriptors with noRegionalBlackoutFlag set to 0 will no longer trigger blackouts or Ad Avail slates

FOLLOW
IGNORE

Scte35SpliceInsertWebDeliveryAllowedBehavior (Enum)
When set to ignore, Segment Descriptors with webDeliveryAllowedFlag set to 0 will no longer trigger blackouts or Ad Avail slates

FOLLOW
IGNORE

Scte35TimeSignalApos

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. 443)*
*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. 443)*
*Required: False*

**SmoothGroupAudioOnlyTimecodeControl (Enum)**

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.

*PASSTHROUGH*
*USE_CONFIGURED_CLOCK*

**SmoothGroupCertificateMode (Enum)**

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 unless those certificates are manually added to the OS trusted keystore.

*SELF_SIGNED*
*VERIFY_AUTHENTICITY*

**SmoothGroupEventIdMode (Enum)**

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.

*NO_EVENT_ID*
*USE_CONFIGURED*
*USE_TIMESTAMP*

**SmoothGroupEventStopBehavior (Enum)**

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

*NONE*
*SEND_EOS*

**SmoothGroupSegmentationMode (Enum)**

When set to useInputSegmentation, the output segment or fragment points are set by the RAI markers from the input streams.
USE_INPUT_SEGMENTATION  
USE_SEGMENT_DURATION

**SmoothGroupSparseTrackType (Enum)**

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

- NONE  
- SCTE_35

**SmoothGroupStreamManifestBehavior (Enum)**

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

- DO_NOT_SEND  
- SEND

**SmoothGroupTimestampOffsetMode (Enum)**

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

- USE_CONFIGURED_OFFSET  
- USE_EVENT_START_DATE

**SmpteTtDestinationSettings**

**StandardHlsSettings**

**m3u8Settings**

Type: **M3u8Settings** (p. 432)

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**

**staticKeyValue**

Static key value as a 32 character hexadecimal string.

Type: string  
Required: True
**keyProviderServer**

The URL of the license server used for protecting content.

Type: InputLocation (p. 418)

Required: False

**TeletextDestinationSettings**

**TeletextSourceSettings**

**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**

**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. 447)

Required: True

**TimecodeConfigSource (Enum)**

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.

EMBEDDED

SYSTEMCLOCK
NEWBASED

**TtmlDestinationSettings**

**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. 448)
Required: False

**TtmlDestinationStyleControl (Enum)**

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.

- PASSTHROUGH
- USE_CONFIGURED

**UdpContainerSettings**

**m2tsSettings**

Type: M2tsSettings (p. 424)
Required: False

**UdpGroupSettings**

**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. 418)
Required: False

**timedMetadataId3Frame**

Indicates ID3 frame that has the timecode.

Type: UdpTimedMetadataId3Frame (p. 449)
Required: False

**timedMetadataId3Period**

Timed Metadata interval in seconds.

Type: integer
Required: False
Minimum: 0
UdpOutputSettings

destination

Destination address and port number for RTP or UDP packets. Can be unicast or multicast RTP or UDP (e.g. rtp://239.10.10.10:5001 or udp://10.100.100.100:5002).

Type: OutputLocationRef (p. 441)
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. 448)
Required: True

fecOutputSettings

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

Type: FecOutputSettings (p. 392)
Required: False

UdpTimedMetadataId3Frame (Enum)

Indicates ID3 frame that has the timecode.

NONE
PRIV
TDRL

VideoCodecSettings

h264Settings

Type: H264Settings (p. 397)
Required: False

VideoDescription

respondToAfd

Indicates how to respond to the AFD values in the input stream. Setting to "respond" causes input video to be clipped, depending on AFD value, input display aspect ratio and output display aspect ratio.
properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>VideoDescriptionRespondToAfd (p. 451)</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
<td></td>
<td></td>
</tr>
<tr>
<td>scalingBehavior</td>
<td></td>
<td>False</td>
<td>When set to &quot;stretchToOutput&quot;, automatically configures the output position to stretch the video to the specified output resolution. This option will override any position value.</td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>True</td>
<td>The name of this VideoDescription. Outputs will use this name to uniquely identify this Description. Description names should be unique within this Live Event.</td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
<td></td>
<td></td>
</tr>
<tr>
<td>width</td>
<td>integer</td>
<td>False</td>
<td>Output video width (in pixels). Leave out to use source video width. If left out, height must also be left out. Display aspect ratio is always preserved by letterboxing or pillarboxing when necessary.</td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sharpness</td>
<td>integer</td>
<td>False</td>
<td>Changes the width of the anti-alias filter kernel used for scaling. Only applies if scaling is being performed and antiAlias is set to true. 0 is the softest setting, 100 the sharpest, and 50 recommended for most content.</td>
</tr>
<tr>
<td>Minimum</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>codecSettings</td>
<td>VideoCodecSettings (p. 449)</td>
<td>True</td>
<td>Video codec settings.</td>
</tr>
<tr>
<td>Required</td>
<td>True</td>
<td></td>
<td></td>
</tr>
<tr>
<td>height</td>
<td>integer</td>
<td>False</td>
<td>Output video height (in pixels). Leave blank to use source video height. If left blank, width must also be unspecified.</td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**VideoDescriptionRespondToAfd (Enum)**

Indicates how to respond to the AFD values in the input stream. Setting to "respond" causes input video to be clipped, depending on AFD value, input display aspect ratio and output display aspect ratio.

- NONE
- PASSTHROUGH
- RESPOND

**VideoDescriptionScalingBehavior (Enum)**

When set to "stretchToOutput", automatically configures the output position to stretch the video to the specified output resolution. This option will override any position value.

- DEFAULT
- STRETCH_TO_OUTPUT

**VideoSelector**

**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. 451)
- **Required**: False

**selectorSettings**

The video selector settings.

- **Type**: VideoSelectorSettings (p. 452)
- **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. 452)
- **Required**: False

**VideoSelectorColorSpace (Enum)**

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

- FOLLOW
- REC_601
- REC_709
Properties

**VideoSelectorColorSpaceUsage (Enum)**

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.

FALLBACK
FORCE

**VideoSelectorPid**

pid

Selects a specific PID from within a video source.

Type: integer
Required: False
Minimum: 0
Maximum: 8191

**VideoSelectorProgramId**

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**

videoSelectorPid

Type: VideoSelectorPid (p. 452)
Required: False

videoSelectorProgramId

Type: VideoSelectorProgramId (p. 452)
Required: False
WebvttDestinationSettings

InputSecurityGroups

URI

/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</td>
<td>An array of Input Security Groups</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 455)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 455)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>None</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 455)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>504</td>
<td>None</td>
<td>Gateway Timeout Error</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 455)</td>
<td>Request limit exceeded on list channel calls to channel service.</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. 455)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 455)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>None</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 455)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>504</td>
<td>None</td>
<td>Gateway Timeout Error</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 455)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
</tbody>
</table>

Schemas

Request Bodies

Example POST

```
{
  "whitelistRules (p. 456)": [
    {
      "cidr (p. 456)": "string"
    }
  ]
}
```

Response Bodies

Example ListInputSecurityGroupsResultModel

```
{
  "inputSecurityGroups (p. 457)": [
    {
      "id (p. 456)": "string",
      "arn (p. 456)": "string",
      "whitelistRules (p. 456)": [
        {
          "cidr (p. 456)": "string"
        }
      ]
    }
  ],
  "nextToken (p. 457)": "string"
}
```

Example CreateInputSecurityGroupResultModel

```
{
}
```
"securityGroup (p. 455)": {
   "id (p. 456)": "string",
   "arn (p. 456)": "string",
   "whitelistRules (p. 456)": [
      {
         "cidr (p. 456)": "string"
      }
   ]
}

Example InvalidRequest

```
{
   "message (p. 457)": "string"
}
```

Example AccessDenied

```
{
   "message (p. 455)": "string"
}
```

Example LimitExceeded

```
{
   "message (p. 457)": "string"
}
```

Example InternalServiceError

```
{
   "message (p. 457)": "string"
}
```

Properties

AccessDenied

message

Type: string
Required: False

CreateInputSecurityGroupResultModel

securityGroup

Type: InputSecurityGroup (p. 456)
Required: False
**InputSecurityGroup**

**id**

The Id of the Input Security Group

  Type: string  
  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. 456)  
  Required: False

**InputSecurityGroupWhitelistRequest**

**whitelistRules**

List of IPv4 CIDR addresses to whitelist

  Type: Array of type InputWhitelistRuleCidr (p. 456)  
  Required: False

**InputWhitelistRule**

**cidr**

The IPv4 CIDR that's whitelisted.

  Type: string  
  Required: False

**InputWhitelistRuleCidr**

**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**

**inputSecurityGroups**

List of input security groups

- **Type:** Array of type InputSecurityGroup (p. 456)
- **Required:** False

**nextToken**

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

**InputSecurityGroups inputSecurityGroupId**

**URI**

/inputSecurityGroups/inputSecurityGroupId

**HTTP Methods**

**GET**

Operation ID: DescribeInputSecurityGroup

Produces a summary of an Input Security Group
AWS Elemental MediaLive API Reference

HTTP Methods

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. 459)</td>
<td>An Input Security Group</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 459)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 460)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>None</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 459)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 459)</td>
<td>The channel you’re requesting to describe does not exist.</td>
</tr>
<tr>
<td>504</td>
<td>None</td>
<td>Gateway Timeout Error</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 459)</td>
<td>Request limit exceeded on list channel calls to channel service.</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. 459)</td>
<td>An Input Security Group</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 459)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 460)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>None</td>
<td>Bad Gateway Error</td>
</tr>
</tbody>
</table>
Schemas

Response Bodies

Example InputSecurityGroup

```json
{
  "id (p. 460)”: "string",
  "arn (p. 460)”: "string",
  "whitelistRules (p. 460)": [
    {  
      "cidr (p. 460)”: "string"
    }
  ]
}
```

Example Empty

```json
{
}
```

Example InvalidRequest

```json
{
  "message (p. 461)": "string"
}
```

Example AccessDenied

```json
{
  "message (p. 460)": "string"
}
```

Example ResourceNotFound

```json
{
  "message (p. 461)": "string"
}
```

Example LimitExceeded

```json
{
}
```
"message (p. 461)": "string"
}

Example InternalServiceError

{
  "message (p. 461)": "string"
}

Properties

AccessDenied

message

  Type: string
  Required: False

Empty

InputSecurityGroup

id

The Id of the Input Security Group

  Type: string
  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. 460)
  Required: False

InputWhitelistRule

cidr

The IPv4 CIDR that's whitelisted.

  Type: string
  Required: False
**Inputs**

**URI**

/inputs

**HTTP Methods**

**GET**

Operation ID: ListInputs

Produce 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. 463)</td>
<td>An array of inputs</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 464)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 464)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>None</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 464)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>504</td>
<td>None</td>
<td>Gateway Timeout Error</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 464)</td>
<td>Request limit exceeded on list channel calls to channel service.</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. 463)</td>
<td>Creation of channel is started.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 464)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 464)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>None</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 464)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>504</td>
<td>None</td>
<td>Gateway Timeout Error</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 464)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
</tbody>
</table>

Schemas

Request Bodies

Example POST

```json
{
  "inputSecurityGroups (p. 465)": [
    "string"
  ]
} 462```
 Response Bodies

Example ListInputsResultModel

```json
{
  "inputs (p. 469)": [
    {
      "attachedChannels (p. 466)": [
        "string"
      ],
      "sources (p. 466)": [
        {
          "passwordParam (p. 467)": "string",
          "url (p. 467)": "string",
          "username (p. 468)": "string"
        }
      ],
      "destinations (p. 466)": [
        {
          "port (p. 467)": "string",
          "ip (p. 467)": "string",
          "url (p. 467)": "string"
        }
      ],
      "name (p. 466)": "string",
      "securityGroups (p. 466)": [
        "string"
      ],
      "id (p. 466)": "string",
      "state (p. 466)": enum,
      "arn (p. 466)": "string",
      "type (p. 467)": enum
    }
  ],
  "nextToken (p. 469)": "string"
}
```

Example CreateInputResultModel

```json
{
  "input (p. 465)": {
    "attachedChannels (p. 466)": [
      "string"
    ],
    "sources (p. 465)": [
      {
        "passwordParam (p. 468)": "string",
        "url (p. 468)": "string",
        "username (p. 468)": "string"
      }
    ],
    "requestId (p. 465)": "string",
    "destinations (p. 465)": [
      {
        "streamName (p. 467)": "string"
      }
    ],
    "name (p. 465)": "string",
    "type (p. 465)": enum
  }
}
```
"sources (p. 466)": [  
  {  
    "passwordParam (p. 467)": "string",  
    "url (p. 467)": "string",  
    "username (p. 468)": "string"  
  },  
],  
"destinations (p. 466)": [  
  {  
    "port (p. 467)": "string",  
    "ip (p. 467)": "string",  
    "url (p. 467)": "string"  
  },  
],  
"name (p. 466)": "string",  
"securityGroups (p. 466)": [  
  "string"  
],  
"id (p. 466)": "string",  
"state (p. 466)": enum,  
"arn (p. 466)": "string",  
"type (p. 467)": enum
}

Example InvalidRequest

{
  "message (p. 469)": "string"
}

Example AccessDenied

{
  "message (p. 464)": "string"
}

Example LimitExceeded

{
  "message (p. 469)": "string"
}

Example InternalServiceError

{
  "message (p. 468)": "string"
}

Properties

AccessDenied

message

Type: string
Required: False

CreateInput

inputSecurityGroups

A list of security groups referenced by IDs to attach to the input.

Type: Array of type string
Required: False

sources

settings required for PULL-type inputs; one per redundancy group Only one of sources and destinations can be specified

Type: Array of type InputSourceRequest (p. 468)
Required: False

requestId

Unique identifier of the request to ensure the request is handled exactly once in case of retries

Type: string
Required: False

destinations

settings required for PUSH-type inputs; one per redundancy group. Only one of sources and destinations can be specified. Note: there are currently no settings required for PUSH-type inputs

Type: Array of type InputDestinationRequest (p. 467)
Required: False

name

Name of the input.

Type: string
Required: False

type

Type: InputType (p. 468)
Required: False

CreateInputResultModel

input

Type: Input (p. 466)
Required: False
### Input

**attachedChannels**

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

**sources**

List of sources of input (PULL-type)

- **Type:** Array of type **InputSource** (p. 467)
- **Required:** False

**destinations**

List of destinations of input (PULL-type)

- **Type:** Array of type **InputDestination** (p. 467)
- **Required:** False

**name**

User-assigned name (mutable)

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

**securityGroups**

List of IDs for all the security groups attached to the input.

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

**id**

generated ID of input (unique for user account, immutable)

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

**state**

- **Type:** **InputState** (p. 468)
- **Required:** False

**arn**

Unique ARN of input (generated, immutable)

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

type

Type: InputType (p. 468)
Required: False

InputDestination

port

port for input

Type: string
Required: False

ip

system-generated static IP address of endpoint. Remains fixed for the lifetime of the input

Type: string
Required: False

url

This represents the endpoint that the customer stream will be pushed to.

Type: string
Required: False

InputDestinationRequest

streamName

A unique name for the location the RTMP stream is being pushed to.

Type: string
Required: False

InputSource

passwordParam

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
username for input source
  Type: string
  Required: False

**InputSourceRequest**

**passwordParam**
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
username for input source
  Type: string
  Required: False

**InputState (Enum)**

- CREATING
- DETACHED
- ATTACHED
- DELETING
- DELETED

**InputType (Enum)**

- UDP_PUSH
- RTP_PUSH
- RTMP_PUSH
- RTMP_PULL
- URL_PULL

**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. 466)
  Required: False

nextToken
  Type: string
  Required: False

Inputs inputId

URI

/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>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Input (p. 471)</td>
<td>Input details</td>
</tr>
</tbody>
</table>
**HTTP Methods**

### Status Code | Response Model | Description
--- | --- | ---
400 | InvalidRequest (p. 471) | This request was invalid.
500 | InternalServiceError (p. 472) | Unexpected internal service error.
502 | None | Bad Gateway Error
403 | AccessDenied (p. 471) | You do not have permission to list channels.
404 | ResourceNotFound (p. 471) | The channel you’re requesting to describe does not exist.
504 | None | Gateway Timeout Error
429 | LimitExceeded (p. 472) | Request limit exceeded on list channel calls to channel service.

### 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. 471)</td>
<td>Successful deletion</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 471)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 472)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>None</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 471)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 471)</td>
<td>The channel you’re requesting to describe does not exist.</td>
</tr>
<tr>
<td>504</td>
<td>None</td>
<td>Gateway Timeout Error</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 472)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 472)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
</tbody>
</table>
Schemas

Response Bodies

Example Input

```json
{
  "attachedChannels (p. 472)": ["string"],
  "sources (p. 472)": [
    {
      "passwordParam (p. 474)": "string",
      "url (p. 474)": "string",
      "username (p. 474)": "string"
    }
  ],
  "destinations (p. 472)": [
    {
      "port (p. 473)": "string",
      "ip (p. 473)": "string",
      "url (p. 474)": "string"
    }
  ],
  "name (p. 473)": "string",
  "securityGroups (p. 473)": [
    "string"
  ],
  "id (p. 473)": "string",
  "state (p. 473)": enum,
  "arn (p. 473)": "string",
  "type (p. 473)": enum
}
```

Example Empty

```json
{}
```

Example InvalidRequest

```json
{
  "message (p. 475)": "string"
}
```

Example AccessDenied

```json
{
  "message (p. 472)": "string"
}
```

Example ResourceNotFound

```json
{
  "message (p. 475)": "string"
}
```
Example ResourceConflict

```json
{
    "message (p. 475)": "string"
}
```

Example LimitExceeded

```json
{
    "message (p. 475)": "string"
}
```

Example InternalServiceError

```json
{
    "message (p. 474)": "string"
}
```

Properties

AccessDenied

message

Type: string
Required: False

Empty

Input

attachedChannels

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

sources

List of sources of input (PULL-type)

Type: Array of type InputSource (p. 474)
Required: False

destinations

List of destinations of input (PULL-type)

Type: Array of type InputDestination (p. 473)
Required: False
name
user-assigned name (mutable)
  Type: string
  Required: False

securityGroups
List of IDs for all the security groups attached to the input.
  Type: Array of type string
  Required: False

id
generated ID of input (unique for user account, immutable)
  Type: string
  Required: False

state
  Type: InputState (p. 474)
  Required: False

arn
Unique ARN of input (generated, immutable)
  Type: string
  Required: False

type
  Type: InputType (p. 474)
  Required: False

InputDestination

port
port for input
  Type: string
  Required: False

ip
system-generated static IP address of endpoint. Remains fixed for the lifetime of the input
  Type: string
  Required: False
url
This represents the endpoint that the customer stream will be pushed to.

Type: string
Required: False

InputSource

passwordParam
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
username for input source

Type: string
Required: False

InputState (Enum)

CREATING
DETACHED
ATTACHED
DELETING
DELETED

InputType (Enum)

UDP_PUSH
RTP_PUSH
RTMP_PUSH
RTMP_PULL
URL_PULL

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