



## CHAPTER 2

# Monitoring APIs

---

This chapter describes the format and content of the Monitoring API messages. The Monitoring API messages consist of the following:

- [MaxErrorModifiedTime](#), page 2-2
- [MaxStreamModifiedTime](#), page 2-2
- [StreamHistory](#), page 2-3
- [StreamListHistory](#), page 2-5
- [TrickModeHistory](#), page 2-7
- [TrickModeListHistory](#), page 2-9
- [PlayServerHistory](#), page 2-11
- [Errors](#), page 2-13
- [Devices](#), page 2-15
- [ServiceGroups](#), page 2-16
- [ContentState](#), page 2-17
- [Version](#), page 2-18
- [ContentRange](#), page 2-19
- [CapacityPlanning](#), page 2-21
- [Trick-mode Event Reconciliation](#), page 2-23



**Note**

Devices, ServiceGroups, ContentState, Version, ContentRange, and Trick-mode Event Reconciliation response messages are not time-sensitive and always return the current state of the requested information.

---

All HTTP request messages for the Monitoring APIs follow the HTTP GET format. All response messages for the Monitoring APIs return an XML document.

# MaxErrorModifiedTime

The MaxErrorModifiedTime request-response message returns the timestamp of the latest error in the database.

## Request

Required: **MaxErrorModifiedTime**

## Request Example

```
http://209.165.201.1/everstream/api.php?messageType=MaxErrorModifiedTime
```

## Response

One of the following HTTP status codes is returned:

- 200 Ok—Request was successful.
- 400 Bad Request—Request parameters were incomplete or invalid.
- 404 Not Found—MessageType was invalid.

The MaxErrorModifiedTimeResponse element is returned in the XML body response.

Table 2-1 describes the XML body elements and attributes returned in the MaxErrorModifiedTimeResponse element.

**Table 2-1** MaxErrorModifiedTimeResponse

Element	Attributes	Data Types	Descriptions
MaxErrorModifiedTime	—	list element	Element that contains the information.
	maxTime	xs:integer	The latest time (since start of UNIX epoch time) that an error occurred in the database. <sup>1</sup>
	curTime	xs:integer	The current time on the CDSM (since the start of UNIX epoch time). <sup>1</sup>

1. UNIX epoch time is 1970-01-01T00:00:00Z.

## Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<MaxErrorModifiedTimeResponse
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../src/xsd/MaxErrorModifiedTime.xsd">
  <MaxErrorModifiedTime maxTime="1233884342" curTime="1233884888" />
</MaxErrorModifiedTimeResponse>
```

# MaxStreamModifiedTime

The MaxStreamModifiedTime request-response message returns the timestamp of the latest stream that was modified in the database.

## Request

Required: **MaxStreamModifiedTime**

**Request Example**

```
http://209.165.201.1/everstream/api.php?messageType=MaxStreamModifiedTime
```

**Response**

One of the following HTTP status codes is returned:

- 200 Ok—Request was successful.
- 400 Bad Request—Request parameters were incomplete or invalid.
- 404 Not Found—MessageType was invalid.

The MaxStreamModifiedTimeResponse element is returned in the XML body response.

Table 2-2 describes the XML body elements and attributes returned in the MaxStreamModifiedTimeResponse element.

**Table 2-2** MaxStreamModifiedTimeResponse

Element	Attributes	Data Types	Descriptions
MaxStreamModifiedTime	—	list element	Element that contains the information.
	maxTime	xs:integer	The latest time (since start of UNIX epoch time) that a stream was modified in the database. <sup>1</sup>
	curTime	xs:integer	The current time on the CDSM (since the start of UNIX epoch time). <sup>1</sup>

1. UNIX epoch time is 1970-01-01T00:00:00Z.

**Response Example**

```
<?xml version="1.0" encoding="UTF-8"?>
<MaxStreamModifiedTimeResponse
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../src/xsd/MaxStreamModifiedTime.xsd">
  <MaxStreamModifiedTime maxTime="1233884552" curTime="1233884873" />
</MaxStreamModifiedTimeResponse>
```

# StreamHistory

The StreamHistory request-response message returns a list of streams that were served during the specified period of time.

**Note**

We recommend using the Errors API call in association with the StreamHistory API call in order to view any error records pertaining to the session ID.

**Request**

Required: **StreamHistory**

Required: **fromDate**

Required: **toDate**

The length of time between `fromDate` and `toDate` must not exceed one hour. The request example uses the `maxRows`, `fromOffset`, and `dateFormat` optional parameters.

### Request Example

```
http://209.165.201.1/everstream/api.php?messageType=StreamHistory&fromDate=1233856800&toDate=1233860399&maxRows=1000000&fromOffset=0&dateFormat=sec
```

### Response

One of the following HTTP status codes are returned:

- 200 Ok—Request was successful.
- 400 Bad Request—Request parameters were incomplete or invalid.
- 404 Not Found—Message type was invalid.

The `StreamsResponse` element is returned in the XML body response.

[Table 2-3](#) describes the XML body elements and attributes returned in the `StreamsResponse` element.

**Table 2-3** *StreamsResponse*

Element	Attributes	Data Types	Descriptions
Stream	—	list element	Element that contains the stream information.
	<code>startTime</code>	<code>xs:integer</code>	Stream start time (since start of UNIX epoch time). <sup>1</sup>
	<code>endTime</code>	<code>xs:integer</code>	Stream end time (since start of UNIX epoch time). <sup>1</sup>
	<code>dbTime</code>	<code>xs:integer</code>	Time (since start of UNIX epoch time) this stream was recorded in the database.
	<code>tsidOutputs</code>	<code>xs:string</code>	TSID <sup>2</sup> output.
	<code>macAddress</code>	<code>xs:string</code>	MAC address of the STB <sup>3</sup> .
	<code>serverIpAddress</code>	<code>xs:string</code>	IP address of the Streamer.
	<code>serviceGroup</code>	<code>xs:integer</code>	Service Group used to send the stream.
	<code>sessionID</code>	<code>xs:string</code>	Session ID associated with this stream.
	<code>streamType</code>	<code>xs:string</code>	Stream type (either user or sys) is used to differentiate between user-requested streams and system streams.
	<code>contentName</code>	<code>xs:string</code>	Name of the content.
	<code>assetIOR</code>	<code>xs:string</code>	IOR <sup>4</sup> for the content.
	<code>bitrate</code>	<code>xs:integer</code>	Rate the content is being streamed, in bits per second.
	<code>assetName</code>	<code>xs:string</code>	Name used in the ADI <sup>5</sup> metadata for a group of content objects that make up one asset.

1. UNIX epoch time is 1970-01-01T00:00:00Z.

2. TSID = Transport Stream ID.

3. STB = set-top box.

4. IOR = interoperable object reference.

5. ADI = asset distribution interface.

### Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<StreamsResponse
```

```

dateFormat="sec"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="../../src/xsd/Streams.xsd">

  <Stream startTime="1233857400" endTime="1233857562" dbTime="1233859791" tsidOutput="0"
  macAddress="EmulatedSess" serverIpAddress="" serviceGroup="22"
  sessionID="EmulatedSession:209.165.201.15:14996:1233857403739" streamType="user"
  contentName="content info unknown: 3857732138715670" assetIOR="" bitrate="" assetName="" />

  <Stream startTime="1233857400" endTime="1233857562" dbTime="1233859791" tsidOutput="0"
  macAddress="EmulatedSess" serverIpAddress="" serviceGroup="22"
  sessionID="EmulatedSession:209.165.201.15:14997:1233857403839" streamType="user"
  contentName="content info unknown: 3857732138715677" assetIOR="" bitrate="" assetName="" />

  <Stream startTime="1233857400" endTime="1233859212" dbTime="1233859818" tsidOutput="0"
  macAddress="EmulatedSess" serverIpAddress="" serviceGroup="22"
  sessionID="EmulatedSession:209.165.201.15:15000:1233857404139" streamType="user"
  contentName="ccLongHD_11.mpg"
  assetIOR="IOR:010000002600000049444c3a6973612e7477632e636f6d2f41737365744d6f64756c652f4173
  7365743a312e3000000001000000000000007c000000010102000e0000003137322e32322e39392e313736001a
  423000000014010f004e55500000001200000000010000000049534150726f7879000000000001000000417272
  6f796f4173736574020000000000000080000000100000004f415401000001400000001313d000100010000
  00000090101000000000" bitrate="15000000" assetName="ArroyoAsset" />
</StreamsResponse>

```

## StreamListHistory

The StreamListHistory API provides the ability to list content playlists for each session that occurred in the specified time period. The StreamListHistory API augments the StreamHistory API by providing information on single-content sessions and multi-content (playlist) sessions. The StreamHistory API provides information for single-content sessions.

The StreamListHistory request-response message returns a list of single content streams and multi-content streams (playlists) for all sessions that occurred during the specified period of time.



### Note

We recommend using the Errors API call in association with the StreamListHistory API call in order to view any error records pertaining to the session ID. See the [“Errors” section on page 2-13](#) for more information.

### Request

Required: **StreamListHistory**

Required: **fromDate**

Required: **toDate**

The length of time between fromDate and toDate must not exceed one hour. The request example uses the maxRows, fromOffset, and dateFormat optional parameters.

### Request Example

```

http://209.165.201.1/everstream/api.php?messageType=StreamListHistory&fromDate=1233856800&
toDate=1233860399&maxRows=1000000&fromOffset=0&dateFormat=sec

```

**Response**

One of the following HTTP status codes are returned:

- 200 Ok—Request was successful.
- 400 Bad Request—Request parameters were incomplete or invalid.
- 404 Not Found—MessageType was invalid.

The StreamsResponse element is returned in the XML body response.

Table 2-4 describes the XML body elements and attributes returned in the StreamsResponse element.

**Table 2-4 StreamsResponse for StreamListHistory**

Element	Attributes	Data Types	Descriptions
Stream	—	list element	Element that contains the stream information.
	startTime	xs:integer	Stream start time (since start of UNIX epoch time). <sup>1</sup>
	endTime	xs:integer	Stream end time (since start of UNIX epoch time). <sup>1</sup>
	dbTime	xs:integer	Time (since start of UNIX epoch time) this stream was recorded in the database.
	tsidOutputs	xs:string	TSID <sup>2</sup> output.
	macAddress	xs:string	MAC address of the STB <sup>3</sup> .
	serverIpAddress	xs:string	IP address of the Streamer.
	serviceGroup	xs:integer	Service Group used to send the stream.
	sessionID	xs:string	Session ID associated with this stream.
	streamType	xs:string	Stream type (either user or sys) is used to differentiate between user-requested streams and system streams.
Content		Subelement	Element that contains information about a content object.
Content	contentName	xs:string	Name of the content.
	bitrate	xs:integer	Rate the content is being streamed, in bits per second.
	assetName	xs:string	Name used in the ADI <sup>4</sup> metadata for a group of content objects that make up one asset.
	contentLength	xs:integer	Duration, in milliseconds, of the content.
	segmentIndex	xs:integer	Identifies the content location in a playlist; for example, a value of 0 is the first content, 1 is the second content, and so on.

1. UNIX epoch time is 1970-01-01T00:00:00Z.

2. TSID = Transport Stream ID.

3. STB = set-top box.

4. ADI = asset distribution interface.

**Response Example**

```
<?xml version="1.0" encoding="UTF-8"?>
<StreamsResponse
  dateFormat="sec"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../../src/xsd/Streams.xsd">
```

```

<Stream startTime="1331582148" endTime="1331592074" dbTime="1331592672" tsidOutput="0"
macAddress="1331582148:1" serverIpAddress="" serviceGroup="1" sessionID="1331582148:117644001"
streamType="user">
  <Content contentName="DK00694_019.mpg" assetName="ArroyoAsset" bitrate="3750000" contentLength="3599212"
segmentIndex="0" />
</Stream>

<Stream startTime="1331582456" endTime="1331592708" dbTime="1331593306" tsidOutput="0"
macAddress="1331582456:6" serverIpAddress="172.22.98.74" serviceGroup="1" sessionID="1331582456:693585001"
streamType="user">
  <Content contentName="StaMain.mpg" assetName="arroyo.tv::AVSC00000000000000402" bitrate="3371250"
contentLength="6334786" segmentIndex="0" />
</Stream>

<Stream startTime="1331582761" endTime="1331592708" dbTime="1331593306" tsidOutput="0"
macAddress="1331582761:2" serverIpAddress="172.22.98.74" serviceGroup="1" sessionID="1331582761:281367002"
streamType="user">
  <Content contentName="StaMain.mpg" assetName="arroyo.tv::AVSC00000000000000402" bitrate="3371250"
contentLength="6334786" segmentIndex="0" />
  <Content contentName="Test_BBQ.mpg" assetName="ArroyoAsset" bitrate="3750000" contentLength="2661405"
segmentIndex="1" />
  <Content contentName="StaMain.mpg" assetName="arroyo.tv::AVSC00000000000000402" bitrate="3371250"
contentLength="6334786" segmentIndex="2" />
</Stream>

<Stream startTime="1331582063" endTime="1331592708" dbTime="1331593306" tsidOutput="0"
macAddress="C0A800F0DB2A" serverIpAddress="172.22.98.74" serviceGroup="1"
sessionID="C0A800F0DB2A::1::3232235818::333" streamType="user">
  <Content contentName="StaMain.mpg" assetName="arroyo.tv::AVSC00000000000000402" bitrate="3371250"
contentLength="6334786" segmentIndex="0" />
</Stream>
</StreamsResponse>

```

## TrickModeHistory

The TrickModeHistory request-response message returns a list of trick-mode actions for all sessions during the specified period of time.

### Request

Required: **TrickModeHistory**

Required: **fromDate**

Required: **toDate**

The length of time between fromDate and toDate must not exceed one hour. The request example uses the maxRows, fromOffset, and dateFormat optional parameters.

### Request Example

```

http://209.165.201.1/everstream/api.php?messageType=TrickModeHistory&fromDate=1235012400&t
oDate=1235015999&maxRows=10000&fromOffset=0&dateFormat=sec

```

### Response

One of the following HTTP status codes are returned:

- 200 Ok—Request was successful.

- 400 Bad Request—Request parameters were incomplete or invalid.
- 404 Not Found—MessageType was invalid.

The TrickModesResponse element is returned in the XML body response.

Table 2-5 describes the XML body elements and attributes returned in the TrickModesResponse.

**Table 2-5** TrickModesResponse

Element	Attributes	Data Types	Descriptions
TrickMode	—	list element	Element that contains the trick-mode information.
	NPT	xs:integer	The current normal play time. NPT <sup>1</sup> starts at 0 at the start of the video, advances in real time in normal play mode, decrements in reverse mode, and is fixed when the video is paused.
	errorText	xs:string	Information on any errors that occurred during trick play.
	eventTime	xs:integer	Time (since start of UNIX epoch time) <sup>2</sup> this trick mode was recorded in the database.
	scale	xs:string	The direction and speed of play.
	sessionID	xs:string	Session ID associated with this stream.

1. NPT = normal play time.
2. UNIX epoch time is 1970-01-01T00:00:00Z.

### Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<TrickModesResponse
  dateFormat="sec"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../../src/xsd/TrickModes.xsd">
  <TrickMode
    NPT="0"
    errorText=""
    eventTime="1235013676"
    scale="0"
    sessionID="C0A800F0DB2A::1235013676:664506::13"
  />
  <TrickMode
    NPT="0"
    errorText=""
    eventTime="1235013676"
    scale="1"
    sessionID="C0A800F0DB2A::1235013676:664506::13"
  />
  <TrickMode
    NPT="0"
    errorText=""
    eventTime="1235014331"
    scale="0"
    sessionID="C0A800F0DB2A::1235013676:664506::13"
  />
</TrickModesResponse>
```



# TrickModeListHistory

The TrickModeListHistory API provides the ability to list content playlists for each session that occurred in the specified time period. The TrickModeListHistory API augments the TrickModeHistory API by providing information on single content session and multi-content (playlist) sessions. The TrickModeHistory API provides information for single-content sessions.

The TrickModeHistory request-response message returns a list of trick-mode actions for single content streams and multi-content streams (playlists) for all sessions that occurred during the specified period of time.

## Request

Required: **TrickModeHistory**

Required: **fromDate**

Required: **toDate**

The length of time between fromDate and toDate must not exceed one hour. The request example uses the maxRows, fromOffset, and dateFormat optional parameters.

## Request Example

```
http://209.165.201.1/everstream/api.php?messageType=TrickModeHistory&fromDate=1235012400&toDate=1235015999&maxRows=100000&fromOffset=0&dateFormat=sec
```

## Response

One of the following HTTP status codes are returned:

- 200 Ok—Request was successful.
- 400 Bad Request—Request parameters were incomplete or invalid.
- 404 Not Found—MessageType was invalid.

The TrickModesResponse element is returned in the XML body response.

Table 2-6 describes the XML body elements and attributes returned in the TrickModesResponse.

**Table 2-6** TrickModesResponse for TrickModeListHistory

Element	Attributes	Data Types	Descriptions
TrickMode	—	list element	Element that contains the trick-mode information.
	NPT	xs:integer	The current normal play time. NPT <sup>1</sup> starts at 0 at the start of the video, advances in real time in normal play mode, decrements in reverse mode, and is fixed when the video is paused.
	errorText	xs:string	Information on any errors that occurred during trick play.
	eventTime	xs:integer	Time (since start of UNIX epoch time) <sup>2</sup> this trick mode was recorded in the database.
	scale	xs:string	The direction and speed of play.
	sessionID	xs:string	Session ID associated with this stream.
	segmentIndex	xs:integer	Identifies the content segment the trick-mode occurred in. If the content object is not part of a playlist, but instead a single content object, the value is zero (0).
	type	xs:integer	LSCP op code.
	mode	xs:integer	LSCP stream mode.
	response	xs:integer	LSCP response code. <b>Note</b> Currently, the response attribute is not supported and always returns zero (0).

1. NPT = normal play time.
2. UNIX epoch time is 1970-01-01T00:00:00Z.

### Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<TrickModesResponse
  dateFormat="sec"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../../src/xsd/TrickModes.xsd">
  <TrickMode NPT="0" errorText="" eventTime="1331582456" scale="0" sessionID="1331582456:693585001"
  segmentIndex="0" type="0" mode="0" response="0" />
  <TrickMode NPT="0" errorText="" eventTime="1331582456" scale="1" sessionID="1331582456:693585001"
  segmentIndex="0" type="160" mode="4" response="0" />
  <TrickMode NPT="0" errorText="" eventTime="1331582456" scale="1" sessionID="1331582456:693585001"
  segmentIndex="0" type="6" mode="4" response="0" />
  <TrickMode NPT="576920" errorText="" eventTime="1331583034" scale="10" sessionID="1331582456:693585001"
  segmentIndex="0" type="6" mode="4" response="0" />
  <TrickMode NPT="614372" errorText="" eventTime="1331583038" scale="0" sessionID="1331582456:693585001"
  segmentIndex="0" type="1" mode="1" response="0" />
  <TrickMode NPT="615791" errorText="" eventTime="1331583046" scale="-10" sessionID="1331582456:693585001"
  segmentIndex="0" type="6" mode="4" response="0" />
  <TrickMode NPT="503550" errorText="" eventTime="1331583057" scale="1" sessionID="1331582456:693585001"
  segmentIndex="0" type="6" mode="4" response="0" />
  <TrickMode NPT="6334785" errorText="" eventTime="1331588888" scale="1" sessionID="1331582456:693585001"
  segmentIndex="0" type="161" mode="4" response="0" />
  <TrickMode NPT="6334785" errorText="" eventTime="1331588888" scale="1" sessionID="1331582456:693585001"
  segmentIndex="0" type="64" mode="7" response="0" />
  <TrickMode NPT="6334785" errorText="" eventTime="1331588888" scale="1" sessionID="1331582456:693585001"
  segmentIndex="0" type="1" mode="1" response="0" />
</TrickModesResponse>
```

```

<TrickMode NPT="10154785" errorText="" eventTime="1331588888" scale="1" sessionID="1331582456:693585001"
segmentIndex="0" type="161" mode="1" response="0" />
<TrickMode NPT="0" errorText="" eventTime="1331592708" scale="0" sessionID="1331582456:693585001"
segmentIndex="0" type="65" mode="0" response="0" />
<TrickMode NPT="0" errorText="" eventTime="1331582761" scale="0" sessionID="1331582761:281367002"
segmentIndex="0" type="0" mode="0" response="0" />
<TrickMode NPT="0" errorText="" eventTime="1331582761" scale="1" sessionID="1331582761:281367002"
segmentIndex="0" type="160" mode="4" response="0" />
<TrickMode NPT="0" errorText="" eventTime="1331582761" scale="1" sessionID="1331582761:281367002"
segmentIndex="0" type="6" mode="4" response="0" />
<TrickMode NPT="320040" errorText="" eventTime="1331583081" scale="0" sessionID="1331582761:281367002"
segmentIndex="0" type="1" mode="1" response="0" />
<TrickMode NPT="320040" errorText="" eventTime="1331583089" scale="1" sessionID="1331582761:281367002"
segmentIndex="0" type="6" mode="4" response="0" />
<TrickMode NPT="329044" errorText="" eventTime="1331583098" scale="10" sessionID="1331582761:281367002"
segmentIndex="0" type="6" mode="4" response="0" />
<TrickMode NPT="864808" errorText="" eventTime="1331583152" scale="1" sessionID="1331582761:281367002"
segmentIndex="0" type="6" mode="4" response="0" />
<TrickMode NPT="6334785" errorText="" eventTime="1331588622" scale="1" sessionID="1331582761:281367002"
segmentIndex="0" type="161" mode="4" response="0" />
<TrickMode NPT="6334785" errorText="" eventTime="1331588622" scale="1" sessionID="1331582761:281367002"
segmentIndex="1" type="160" mode="4" response="0" />
<TrickMode NPT="8996190" errorText="" eventTime="1331591285" scale="1" sessionID="1331582761:281367002"
segmentIndex="1" type="161" mode="4" response="0" />
<TrickMode NPT="8996190" errorText="" eventTime="1331591285" scale="1" sessionID="1331582761:281367002"
segmentIndex="2" type="160" mode="4" response="0" />
<TrickMode NPT="9803463" errorText="" eventTime="1331592094" scale="10" sessionID="1331582761:281367002"
segmentIndex="2" type="6" mode="4" response="0" />
<TrickMode NPT="15329546" errorText="" eventTime="1331592647" scale="10" sessionID="1331582761:281367002"
segmentIndex="2" type="161" mode="4" response="0" />
<TrickMode NPT="15329546" errorText="" eventTime="1331592647" scale="10" sessionID="1331582761:281367002"
segmentIndex="2" type="64" mode="7" response="0" />
<TrickMode NPT="15329546" errorText="" eventTime="1331592647" scale="10" sessionID="1331582761:281367002"
segmentIndex="2" type="1" mode="1" response="0" />
<TrickMode NPT="15939546" errorText="" eventTime="1331592647" scale="10" sessionID="1331582761:281367002"
segmentIndex="2" type="161" mode="1" response="0" />
<TrickMode NPT="0" errorText="" eventTime="1331592708" scale="0" sessionID="1331582761:281367002"
segmentIndex="0" type="65" mode="0" response="0" />
<TrickMode NPT="0" errorText="" eventTime="1331582063" scale="1"
sessionID="C0A800F0DB2A::1::3232235818::333" segmentIndex="0" type="7" mode="3" response="0" />
<TrickMode NPT="0" errorText="" eventTime="1331592708" scale="0"
sessionID="C0A800F0DB2A::1::3232235818::333" segmentIndex="0" type="0" mode="0" response="0" />
</TrickModesResponse>

```

## PlayServerHistory

The PlayServerHistory request-response message returns a list of play servers for each trick-mode action for all sessions during the specified period of time.

### Request

Required: **PlayServerHistory**

Required: **fromDate**

Required: **toDate**

The length of time between fromDate and toDate must not exceed one hour. The request example uses the maxRows, fromOffset, and dateFormat optional parameters.

**Request Example**

```
http://209.165.201.1/everstream/api.php?messageType=PlayServerHistory\&fromDate=1261026000
\&toDate=1261029599\&maxRows=1000000\&fromOffset=0\&dateFormat=sec
```

**Response**

One of the following HTTP status codes are returned:

- 200 Ok—Request was successful.
- 400 Bad Request—Request parameters were incomplete or invalid.
- 404 Not Found—MessageType was invalid.

The PlayServerResponse element is returned in the XML body response.

Table 2-7 describes the XML body elements and attributes returned in the PlayServerResponse.

**Table 2-7** *PlayServerResponse*

Element	Attributes	Data Types	Descriptions
PlayServer	—	list element	Element that contains the trick-mode information.
	NPT	xs:integer	The current normal play time. NPT <sup>1</sup> starts at 0 at the start of the video, advances in real time in normal play mode, decrements in reverse mode, and is fixed when the video is paused.
	errorText	xs:string	Information on any errors that occurred during trick play.
	eventTime	xs:integer	Time (since start of UNIX epoch time) this trick mode was recorded in the database. <sup>2</sup>
	playServerID	xs:integer	Play server ID associated with this trick play.
	scale	xs:string	The direction and speed of play.
	sessionID	xs:string	Session ID associated with this stream.

1. NPT = normal play time.

2. UNIX epoch time is 1970-01-01T00:00:00Z.

**Response Example**

```
<?xml version="1.0" encoding="UTF-8"?>
<PlayServerResponse
  dateFormat="sec"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../src/xsd/PlayServer.xsd">
  <PlayServer NPT="0" errorText="" eventTime="1261026446" playServerID="51" scale="0"
  sessionID="COA800F0DB2A::1261026446:569384::1725" />
  <PlayServer NPT="0" errorText="" eventTime="1261026447" playServerID="52" scale="1"
  sessionID="COA800F0DB2A::1261026446:569384::1725" />
  <PlayServer NPT="0" errorText="" eventTime="1261026751" playServerID="51" scale="0"
  sessionID="COA800F0DB2A::1261026446:569384::1725" />
  <PlayServer NPT="0" errorText="" eventTime="1261026760" playServerID="51" scale="0"
  sessionID="COA800F0DB2A::1261026761:49751::1726" />
  <PlayServer NPT="0" errorText="" eventTime="1261026760" playServerID="52" scale="1"
  sessionID="COA800F0DB2A::1261026761:49751::1726" />
  <PlayServer NPT="16" errorText="" eventTime="1261026760" playServerID="52" scale="1"
  sessionID="COA800F0DB2A::1261026761:49751::1726" />
  <PlayServer NPT="6316" errorText="" eventTime="1261026768" playServerID="52" scale="2"
  sessionID="COA800F0DB2A::1261026761:49751::1726" />
  <PlayServer NPT="0" errorText="" eventTime="1261026824" playServerID="51" scale="0"
  sessionID="COA800F0DB2A::1261026761:49751::1726" />
  <PlayServer NPT="0" errorText="" eventTime="1261026833" playServerID="51" scale="0"
  sessionID="COA800F0DB2A::1261026833:382133::1727" />
  <PlayServer NPT="0" errorText="" eventTime="1261026833" playServerID="52" scale="1"
  sessionID="COA800F0DB2A::1261026833:382133::1727" />
  <PlayServer NPT="14" errorText="" eventTime="1261026833" playServerID="52" scale="1"
  sessionID="COA800F0DB2A::1261026833:382133::1727" />
  <PlayServer NPT="83528" errorText="" eventTime="1261026918" playServerID="52" scale="4"
  sessionID="COA800F0DB2A::1261026833:382133::1727" />
  <PlayServer NPT="217365" errorText="" eventTime="1261026953" playServerID="52"
  scale="8" sessionID="COA800F0DB2A::1261026833:382133::1727" />
  <PlayServer NPT="296029" errorText="" eventTime="1261026964" playServerID="52"
  scale="15" sessionID="COA800F0DB2A::1261026833:382133::1727" />
  <PlayServer NPT="398677" errorText="" eventTime="1261026973" playServerID="52"
  scale="-15" sessionID="COA800F0DB2A::1261026833:382133::1727" />
  <PlayServer NPT="320690" errorText="" eventTime="1261026980" playServerID="52"
  scale="-8" sessionID="COA800F0DB2A::1261026833:382133::1727" />
  <PlayServer NPT="0" errorText="" eventTime="1261027020" playServerID="52" scale="-8"
  sessionID="COA800F0DB2A::1261026833:382133::1727" />
  <PlayServer NPT="0" errorText="" eventTime="1261027087" playServerID="51" scale="0"
  sessionID="COA800F0DB2A::1261026833:382133::1727" />
</PlayServerResponse>
```

## Errors

The Errors request-response message returns a list of stream errors that occurred during the specified period of time.

**Request**

Required: **Errors**

Required: **fromDate**

Required: **toDate**

The length of time between fromDate and toDate must not exceed one hour. The request example uses the maxRows, fromOffset, and dateFormat optional parameters.

**Request Example**

```
http://209.165.201.1/everstream/api.php?messageType=Errors&fromDate=1233874800&toDate=1233878399&maxRows=1000000&fromOffset=0&dateFormat=sec
```

**Response**

One of the following HTTP status codes is returned:

- 200 Ok—Request was successful.
- 400 Bad Request—Request parameters were incomplete or invalid.
- 404 Not Found—MessageType was invalid.

The ErrorsResponse element is returned in the XML body response.

Table 2-8 describes the XML body elements and attributes returned in the ErrorsResponse element.

**Table 2-8** ErrorsResponse

Element	Attributes	Data Types	Descriptions
Error	—	list element	Element that contains the error information.
	errorCode	xs:integer	Stream error code.
	errorText	xs:string	Stream error description.
	errorTime	xs:integer	Time (since start of UNIX epoch time <sup>1</sup> ) this error was recorded in the database.
	macAddress	xs:string	MAC address of the STB.
	serverIpAddress	xs:string	IP address of the Streamer.

1. UNIX epoch time is 1970-01-01T00:00:00Z.

**Response Example**

```
<?xml version="1.0" encoding="UTF-8" ?>
<ErrorsResponse
  dateFormat="sec"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../../src/xsd/Errors.xsd">

  <Error errorCode="5005" errorText="AVS_CMPT_OUT_OF_SVC" errorTime="1233875550"
  macAddress="" serverIpAddress="209.165.201.12" serviceGroup="934282940"
  sourceEventID="1233875550::1::1::75::70::4044::112" sessionId="" />

  <Error errorCode="5005" errorText="AVS_CMPT_OUT_OF_SVC" errorTime="1233875580"
  macAddress="" serverIpAddress="209.165.201.12" serviceGroup="934282940"
  sourceEventID="1233875580::1::1::75::70::4044::113" sessionId="" />

  <Error errorCode="5005" errorText="AVS_CMPT_OUT_OF_SVC" errorTime="1233875610"
  macAddress="" serverIpAddress="209.165.201.12" serviceGroup="934282940"
  sourceEventID="1233875610::1::1::75::70::4044::114" sessionId="" />

  <Error errorCode="5005" errorText="AVS_CMPT_OUT_OF_SVC" errorTime="1233875640"
  macAddress="" serverIpAddress="209.165.201.12" serviceGroup="934282940"
  sourceEventID="1233875640::1::1::75::70::4044::115" sessionId="" />

</ErrorsResponse>
```

# Devices

The Devices request-response message returns a list of the CDS devices.

## Request

Required: **Devices**

## Request Example

```
http://209.165.201.1/everstream/api.php?messageType=Devices
```

## Response

One of the following HTTP status codes is returned:

- 200 Ok—Request was successful.
- 400 Bad Request—Request parameters were incomplete or invalid.
- 404 Not Found—MessageType was invalid.

The DevicesResponse element is returned in the XML body response.

[Table 2-9](#) describes the XML body elements and attributes returned in the DevicesResponse element.

**Table 2-9** *DevicesResponse*

Element	Attributes	Data Types	Descriptions
Device	—	list element	Element that contains the device information.
	description	xs:string	Description of the device.
	hostname	xs:string	Hostname of the device.
	ipAddress	xs:string	IP address of the device.
	type	xs:string	Device type (Streamer, Vault, or ISV <sup>1</sup> ).

1. ISV = Integrated Streamer-Vault.

## Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<DevicesResponse
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../../src/xsd/Devices.xsd">
  <Device description="Vault" hostname="vault316" ipAddress="209.165.201.2" type="Vault" />
  <Device description="Vault" hostname="vault318" ipAddress="209.165.201.4" type="Vault" />
  <Device description="Vault" hostname="vault319" ipAddress="209.165.201.5" type="Vault" />
  <Device description="Streamer" hostname="stm44" ipAddress="209.165.201.6" type="Streamer" />
  <Device description="Streamer" hostname="stm74" ipAddress="209.165.201.8" type="Streamer" />
  <Device description="Streamer" hostname="stm75" ipAddress="209.165.201.9" type="Streamer" />
  <Device description="Streamer" hostname="stm76" ipAddress="209.165.201.11" type="Streamer" />
  <Device description="Streamer" hostname="stm77" ipAddress="209.165.201.12" type="Streamer" />
  <Device description="Streamer" hostname="stm78" ipAddress="209.165.201.14" type="Streamer" />
  <Device description="Streamer" hostname="stm79" ipAddress="209.165.201.16" type="Streamer" />
  <Device description="Streamer" hostname="stm81" ipAddress="209.165.201.19" type="Streamer" />
  <Device description="Streamer" hostname="stm82" ipAddress="209.165.201.20" type="Streamer" />
  <Device description="Streamer" hostname="stm83" ipAddress="209.165.201.21" type="Streamer" />
</DevicesResponse>
```

# ServiceGroups

The ServiceGroups request-response message returns a list of the service groups configured in the CDS.

## Request

Required: **ServiceGroups**

## Request Example

```
http://209.165.201.1/everstream/api.php?messageType=ServiceGroups
```

## Response

One of the following HTTP status codes is returned:

- 200 Ok—Request was successful.
- 400 Bad Request—Request parameters were incomplete or invalid.
- 404 Not Found—MessageType was invalid.

The ServiceGroupsResponse element is returned in the XML body response.

[Table 2-10](#) describes the XML body elements and attributes returned in the ServiceGroupsResponse element.

**Table 2-10** *ServiceGroupsResponse*

Element	Attributes	Data Types	Descriptions
ServiceGroup	—	list element	Element that contains the service group information.
	serviceGroup	xs:integer	Service group configured in the CDS.
	TSIDOutput	xs:integer	The associated TSID out port on the QAM <sup>1</sup> device. <sup>2</sup>
	bandwidth	xs:integer	The bandwidth usage of this service group. <sup>2</sup>
	EdgeDeviceIP	xs:string	The IP address of the QAM device. <sup>2</sup>

1. QAM = Quadrature Amplitude Modulation.

2. Only for Asynchronous Serial Interface (ASI) streaming.

## Response Example

The response example is for a CDS configured with Gigabit Ethernet streaming.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceGroupsResponse
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../src/xsd/ServiceGroups.xsd">
  <ServiceGroup serviceGroup="21">
  <ServiceGroup serviceGroup="22">
  <ServiceGroup serviceGroup="73">
</ServiceGroupsResponse>
```



# ContentState

The ContentState request-response message returns a list of the content in the CDS.

## Request

Required: **ContentState**

## Request Example

```
http://209.165.201.1/everstream/api.php?messageType=ContentState
```

## Response

One of the following HTTP status codes is returned:

- 200 Ok—Request was successful.
- 400 Bad Request—Request parameters were incomplete or invalid.
- 404 Not Found—MessageType was invalid.

The ContentStateResponse element is returned in the XML body response.

[Table 2-11](#) describes the XML body elements and attributes returned in the ContentStateResponse element.

**Table 2-11** ContentStateResponse

Element	Attributes	Data Types	Descriptions
ContentState	—	list element	Element that contains the content information.
	assetIOR	xs:string	IOR for the content.
	contentName	xs:string	Name of the content.
	dateIngested	xs:integer	The time (since start of UNIX epoch time) that the content was ingested into the CDS. <sup>1</sup>
	digestIpAddress	xs:string	The IP address of the Vault that ingested the content.
	filesize	xs:integer	The size, in bytes, of the content file.
	isaAdminState	xs:string	The administrative state of this content. The possible administrative states are: <ul style="list-style-type: none"> <li>• Unprovisioned—Content is loading.</li> <li>• In Service—Content is available for streaming.</li> <li>• Out of Service—Content is not available for streaming.</li> </ul>
	isaOpState	xs:string	The operational state of this content. The possible operational states are: <ul style="list-style-type: none"> <li>• Created—Content is loading.</li> <li>• In Service—Content is available for streaming.</li> <li>• Out of Service—Content is not available for streaming.</li> </ul>
	bitrate	xs:integer	The rate of ingest in bits per second (3750000 = 3.75 Mbps).
	assetName	xs:string	Name used in the ADI metadata for a group of content objects that make up one asset.

1. UNIX epoch time is 1970-01-01T00:00:00Z.

**Response Example**

```
<?xml version="1.0" encoding="UTF-8"?>
<ContentStateResponse
  dateFormat=""
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../src/xsd/ContentState.xsd">
  <ContentState

assetIOR="IOR:010000002600000049444c3a6973612e7477632e636f6d2f41737365744d6f64756c652f4173
7365743a312e3000000001000000000000007c000000010102ff0e0000003137322e32322e39392e313736001a
423000000014010f004e55500000001200000000010000000049534150726f787900000000001000000417272
6f796f4173736574020000000000000080000000100000004f4154010000001400000001313d000100010000
0000000901010000000000" contentName="NBA_SR_MichaelvMagic1991-JPW91GB_2.mpg"
dateIngested="1233682841" digestIpAddress="209.165.201.27" filesize="1385901220"
isaAdminState="InService" isaOpState="InService" bitrate="3750000"
assetName="ArroyoAsset" />
  <ContentState

assetIOR="IOR:010000002600000049444c3a6973612e7477632e636f6d2f41737365744d6f64756c652f4173
7365743a312e3000000001000000000000007c000000010102ff0e0000003137322e32322e39392e313736001a
423000000014010f004e55500000001200000000010000000049534150726f787900000000001000000417272
6f796f4173736574020000000000000080000000100000004f4154010000001400000001313d000100010000
0000000901010000000000" contentName="NBA_SR_MichaelvMagic1991-JPW91GB_4.mpg"
dateIngested="1233682900" digestIpAddress="209.165.201.27" filesize="1385901220"
isaAdminState="InService" isaOpState="InService" bitrate="3750000"
assetName="ArroyoAsset" />
</ContentStateResponse>
```

# Version

The Version request-response message returns the version of the API running on the CDS.

**Request**

Required: **Version**

**Request Example**

```
http://209.165.201.1/everstream/api.php?messageType=Version
```

**Response**

One of the following HTTP status codes is returned.

- 200 Ok—Request was successful.
- 400 Bad Request—Request parameters were incomplete or invalid.
- 404 Not Found—MessageType was invalid.

The APIVersionResponse element is returned in the XML body response.

Table 2-12 describes the XML body attributes returned in the `APIVersionResponse`.

**Table 2-12** *APIVersionResponse*

Attributes	Data Types	Descriptions
major	xs:integer	The integer portion of the API version.
minor	xs:integer	The decimal number portion of the API version.

#### Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<APIVersionResponse
  major="1"
  minor="3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="..\..\src\xsd\APIVersion.xsd"/>
```

## ContentRange

The Content Range request-response message returns a list of content objects contained within the CDS network within a specified range.

#### Request

Required: **ContentRange**

Required: **maxRows**

Optionally, you can include the `fromOffset` parameter. The request example uses the `maxRows` parameter to request the first 500 content objects.

#### Request Example

```
http://209.165.201.1/everstream/api.php?messageType=ContentRange&maxRows=500
```

#### Response

One of the following HTTP status codes is returned.

- 200 Ok—Request was successful.
- 400 Bad Request—Request parameters were incomplete or invalid.
- 404 Not Found—MessageType was invalid.

The `ContentRangeResponse` element is returned in the XML body response.

Table 2-13 describes the XML body attributes returned in the ContentRangeResponse element.

**Table 2-13** ContentRangeResponse

Element	Attributes	Data Types	Descriptions
Content	—	list element	Element that contains the content information.
	Name	xs:integer	Name of the content.
	isaAdminState	xs:string	The administrative state of this content. The possible administrative states are: <ul style="list-style-type: none"> <li>Unprovisioned—Content is loading.</li> <li>In Service—Content is available for streaming.</li> <li>Out of Service—Content is not available for streaming.</li> </ul>
	isaOpState	xs:string	The operational state of this content. The possible operational states are: <ul style="list-style-type: none"> <li>Created—Content is loading.</li> <li>In Service—Content is available for streaming.</li> <li>Out of Service—Content is not available for streaming.</li> </ul>

The list of content objects returned is ordered alphabetically by content name.

### Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ContentRangeResponse
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../src/xsd/ContentRangeResponse.xsd">

  <Content Name="example.com::15825::DHW5_512_A1_01_11.mpg"
    isaAdminState="InService"
    isaOpState="InService"/>
  <Content Name="example.com::15828::DHW5_512_HD_A1_01_11.mpg"
    isaAdminState="InService"
    isaOpState="InService"/>
  <Content Name="example.com::15835::SCR8_803_A1_01_13.mpg"
    isaAdminState="InService"
    isaOpState="InService"/>
  ...

  <Content Name="example.com::16134::SCR8_803_HD_A1_01_13.mpg"
    isaAdminState="InService"
    isaOpState="InService"/>

</ContentStateResponse>
```

In the following example, the request messages uses the name of the last content object returned from the previous request as the value of the fromOffset parameter and specifies that a maximum of 500 content objects be returned using the maxRows parameter:

```
http://209.165.201.1/everstream/api.php?messageType=ContentRange&fromOffset=
example.com::16134::SCR8_803_HD_A1_01_13.mpg&maxRows=500
```

The response provides the next set of content objects, starting alphabetically from the next content object after `example.com::16134::SCR8_803_HD_A1_01_13`, up to a maximum of 500 content objects. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<ContentRangeResponse
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../../src/xsd/ContentRangeResponse.xsd">

  <Content Name="example.com::16135::SCR8_803_HD_A1_01_13.mpg"
    isaAdminState="InService"
    isaOpState="InService"/>
  <Content Name="example.com::16136::SCR8_803_HD_A1_01_13.mpg"
    isaAdminState="InService"
    isaOpState="InService"/>
  <Content Name="example.com::16137::SCR8_803_HD_A1_01_13.mpg"
    isaAdminState="InService"
    isaOpState="InService"/>
  <Content Name="example.com::16138::SCR8_803_HD_A1_01_13.mpg"
    isaAdminState="InService"
    isaOpState="InService"/>

  ...

</ContentStateResponse>
```

## CapacityPlanning

The Capacity Planning Report request-response message returns capacity-related data from Streamers and ISVs (SSVs) to assist in monitoring the capacity of the CDS.

### Request

Required: **CapacityPlanning**

Required: **fromDate**

Required: **toDate**

The length of time between `fromDate` and `toDate` must not exceed one hour. Optional parameters include the `action` and `timeType` parameters. If `action` is set to `SERVICEGROUP`, the `serviceGroup` parameter must be provided. If `action` is set to `SERVER`, the `serverID` parameter must be provided.

### Request Examples

The following request example uses the `action` and `timeType` parameters to request peak bandwidth and peak stream counts per day for the specified time range:

```
http://209.165.201.1/everstream/api.php?messageType=CapacityPlanning&fromDate=1285545600&toDate=1285631999&action=OVERALL&timeType=DAY
```

The following request example uses the `action`, `timeType`, and `serviceGroup` parameters to request peak bandwidth and peak stream counts per hour for the specified time range for all Streamers and ISVs in Service Group 200:

```
http://209.165.201.1/everstream/api.php?messageType=CapacityPlanning&fromDate=1285545600&toDate=1285631999&action=SERVICEGROUP&timeType=HOURL&serviceGroup=200
```

The following request example uses the `action`, `timeType`, and `Server ID` parameters to request peak bandwidth and peak stream counts per week for the specified time range for Streamer 223:

<http://209.165.201.1/everstream/api.php?messageType=CapacityPlanning&fromDate=1285545600&toDate=1285631999&action=SERVERID&timeType=WEEK&serverID=223>

### Response

One of the following HTTP status codes is returned.

- 200 Ok—Request was successful.
- 400 Bad Request—Request parameters were incomplete or invalid.
- 404 Not Found—MessageType was invalid.

The CapacityPlanning element is returned in the XML body response.

Table 2-13 describes the XML body attributes returned in the CapacityPlanning element.

**Table 2-14** CapacityPlanning

Element	Attributes	Data Types	Descriptions
Interval	—	list element	Element that contains the interval information.
	starttime	xs:integer	Stream start time (since start of UNIX epoch time). <sup>1</sup>
	endtime	xs:integer	Stream end time (since start of UNIX epoch time). <sup>1</sup>
	hdstremas	xs:integer	Peek number of high-definition stream objects streamed during the specified time interval.
	sdstreams	xs:integer	Peek number of standard-definition stream objects streamed during the specified time interval.
	bandwdith	xs:integer	Peak bandwidth, in megabits, used for active streams during the specified time interval.
	counter	xs:integer	Peak number of streams during the specified time interval.

1. UNIX epoch time is 1970-01-01T00:00:00Z.

### Response Example

```
<?xml version="1.0" encoding="UTF-8" ?>
<CapacityPlanning
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../../src/xsd/CapacityPlanning.xsd">
  <interval starttime="1285545600" endtime="1285549199" hdstreams="0" sdstreams="7"
  bandwidth="19" counter="7" />
  <interval starttime="1285549200" endtime="1285552799" hdstreams="0" sdstreams="7"
  bandwidth="18" counter="7" />
  <interval starttime="1285552800" endtime="1285556399" hdstreams="0" sdstreams="7"
  bandwidth="19" counter="7" />
  <interval starttime="1285556400" endtime="1285559999" hdstreams="0" sdstreams="7"
  bandwidth="19" counter="7" />
  <interval starttime="1285560000" endtime="1285563599" hdstreams="0" sdstreams="7"
  bandwidth="19" counter="7" />
  <interval starttime="1285563600" endtime="1285567199" hdstreams="0" sdstreams="14"
  bandwidth="38" counter="14" />
  <interval starttime="1285567200" endtime="1285570799" hdstreams="0" sdstreams="7"
  bandwidth="18" counter="7" />
  <interval starttime="1285570800" endtime="1285574399" hdstreams="0" sdstreams="7"
  bandwidth="19" counter="7" />
  <interval starttime="1285574400" endtime="1285577999" hdstreams="0" sdstreams="7"
  bandwidth="19" counter="7" />
  <interval starttime="1285578000" endtime="1285581599" hdstreams="0" sdstreams="7"
  bandwidth="19" counter="7" />
</CapacityPlanning>
```

```

    <interval starttime="1285581600" endtime="1285585199" hdstreams="0" sdstreams="14"
bandwidth="38" counter="14" />
    <interval starttime="1285585200" endtime="1285588799" hdstreams="0" sdstreams="7"
bandwidth="19" counter="7" />
    <interval starttime="1285588800" endtime="1285592399" hdstreams="0" sdstreams="7"
bandwidth="19" counter="7" />
    <interval starttime="1285592400" endtime="1285595999" hdstreams="0" sdstreams="7"
bandwidth="19" counter="7" />
    <interval starttime="1285596000" endtime="1285599599" hdstreams="0" sdstreams="7"
bandwidth="19" counter="7" />
    <interval starttime="1285599600" endtime="1285603199" hdstreams="0" sdstreams="14"
bandwidth="38" counter="14" />
    <interval starttime="1285603200" endtime="1285606799" hdstreams="0" sdstreams="7"
bandwidth="19" counter="7" />
    <interval starttime="1285606800" endtime="1285610399" hdstreams="0" sdstreams="6"
bandwidth="16" counter="6" />
    <interval starttime="1285610400" endtime="1285613999" hdstreams="0" sdstreams="8"
bandwidth="22" counter="8" />
    <interval starttime="1285614000" endtime="1285617599" hdstreams="0" sdstreams="7"
bandwidth="19" counter="7" />
    <interval starttime="1285617600" endtime="1285621199" hdstreams="0" sdstreams="14"
bandwidth="38" counter="14" />
    <interval starttime="1285621200" endtime="1285624799" hdstreams="0" sdstreams="7"
bandwidth="19" counter="7" />
    <interval starttime="1285624800" endtime="1285628399" hdstreams="1" sdstreams="6"
bandwidth="16" counter="6" />
</CapacityPlanning>

```

## Trick-mode Event Reconciliation

The Trick-mode Event Reconciliation request-response message returns the list of trick-mode actions for the specified stream.



### Note

The Trick-mode Event Reconciliation API can only be used on completed streams; that is streams that have been destroyed.

### Request

Required: **GetTrickmodeEventList**

Required: **StreamID**

StreamID is the ID of the stream for which a list of trick-modes events is being requested.

### Request Example

```
http://209.165.201.1/apis/TrickmodeEvents.php?messageType=GetTrickmodeEventsList&StreamID=looney.mpg::6/20/2011|22:59:47.996098::1
```

### Response

If the request is successful, the HTTP status code 200 OK is returned in the header and the TrickmodeEventResponse element is returned in the XML body response.

If no streamID was provided in the request message, the HTTP status code 400 Bad Request is returned in the header and the Error element is returned in the XML body element. The Error element contains the error code 101 and the error message 'Missing Stream ID in request'. For example:

```
<Error code="101" message="Missing Stream ID in request"/>
```

If no `messageType` is provided in the request message, the HTTP status code 400 Bad Request is returned in the header and the Error element is returned in the XML body element. The Error element contains the error code 102 and the error message 'Invalid messageType in request'. For example:

```
<Error code="102" message="Invalid messageType in request"/>
```

Table 2-1 describes the XML body elements and attributes returned in the `TrickmodeEventResponse` element.

**Table 2-15** TrickmodeEventResponse

Element	Attributes	Data Types	Descriptions
TrickmodeEventResponse	—	list element	Element that contains the trick-mode event reconciliation response.
	TrickmodeEventList	Subelement	Element that contains the list of trick-mode actions.
TrickmodeEventList	—	list element	Element that contains the list of trick-mode actions.
	StreamID	xs:string	ID of the stream.
	Size	xs:nonNegative Integer	Size of the stream.
	TrickmodeEvent	Subelement	Element that contains information about a trick-mode action.



Table 2-15 TrickmodeEventResponse (continued)

Element	Attributes	Data Types	Descriptions
TrickmodeEvent	—	list element	Element that contains information about a trick-mode action.
	EventMode	xs:string	Type of trick-mode action. Valid values are: <ul style="list-style-type: none"> <li>• TV_Play</li> <li>• TM_PAUSE</li> <li>• TM_STOP</li> </ul>
	ModeScaleNum	xs:string	Direction and speed of the trick-mode action.
	ModeScaleDenom	xs:string	Divider which is always 1.
	CurrentNPT	xs:string	The current normal play time. NPT <sup>1</sup> starts at 0 at the start of the video, advances in real time in normal play mode, decrements in reverse mode, and is fixed when the video is paused.
	EventTime	xs:string	Time (since start of UNIX epoch time) <sup>2</sup> this trick-mode action was recorded in the database.
	SegmentId	xs:nonNegative Integer	Index of the content segment starting from 1.

1. NPT = normal play time.
2. UNIX epoch time is 1970-01-01T00:00:00Z.

### Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<TrickmodeEventResponse xmlns="http://www.cisco.com/schemas/VCPBU/CDS-TV/R0/ciscowebsvcs">
  <TrickmodeEventList StreamID="looney.mpg::6/20/2011|22:59:47.996098::1" Size="7" >
    <TrickmodeEvent EventMode="TM_PLAY" ModeScaleNum="1" ModeScaleDenom="1" CurrentNPT="0"
      EventTime="1308610790010" SegmentId="1" />
    <TrickmodeEvent EventMode="TM_PAUSE" ModeScaleNum="0" ModeScaleDenom="1"
      CurrentNPT="53012" EventTime="1308610843021" SegmentId="1" />
    <TrickmodeEvent EventMode="TM_PLAY" ModeScaleNum="10" ModeScaleDenom="1"
      CurrentNPT="53546" EventTime="1308610848414" SegmentId="1" />
    <TrickmodeEvent EventMode="TM_PLAY" ModeScaleNum="1" ModeScaleDenom="1"
      CurrentNPT="107704" EventTime="1308610853834" SegmentId="1" />
    <TrickmodeEvent EventMode="TM_PLAY" ModeScaleNum="-10" ModeScaleDenom="1"
      CurrentNPT="109976" EventTime="1308610857116" SegmentId="1" />
    <TrickmodeEvent EventMode="TM_PLAY" ModeScaleNum="1" ModeScaleDenom="1"
      CurrentNPT="38484" EventTime="1308610864404" SegmentId="1" />
    <TrickmodeEvent EventMode="TM_STOP" ModeScaleNum="1" ModeScaleDenom="1"
      CurrentNPT="50514" EventTime="1308610876434" SegmentId="1" />
  </TrickmodeEventList>
</TrickmodeEventResponse>
```

The XML schema file for the Trick-mode Event Reconciliation message response is as follows:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ws="http://www.cisco.com/schemas/VCPBU/CDS-TV/R0/ciscowebsvcs"
  targetNamespace="http://www.cisco.com/schemas/VCPBU/CDS-TV/R0/ciscowebsvcs">

  <xs:element name="TrickmodeEventResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ws:TrickmodeEventList" minOccurs="0"
          maxOccurs="1" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="TrickmodeEventList">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ws:TrickmodeEvent" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="StreamID" type="xs:string" use="required" />
      <xs:attribute name="Size" type="xs:nonNegativeInteger"
        use="required" />
    </xs:complexType>
  </xs:element>

  <xs:element name="TrickmodeEvent">
    <xs:complexType>
      <xs:attribute name="EventMode" type="xs:string" use="required" />
      <xs:attribute name="ModeScaleNum" type="xs:string" use="required" />
      <xs:attribute name="ModeScaleDenom" type="xs:string"
        use="required" />
      <xs:attribute name="CurrentNPT" type="xs:string" use="required" />
      <xs:attribute name="EventTime" type="xs:string" use="required" />
      <xs:attribute name="SegmentId" type="xs:nonNegativeInteger"
        use="required" />
    </xs:complexType>
  </xs:element>
</xs:schema>
```