Barco Device Interface
Media Service API

Barco Device Interface Media Service API

Table of Contents

List of Examples

Glossary

Device Interface	This represents the external interface to a device.
Device Interface API	The API accepted by the device on its external interface
Device Interface Media Service API	The API accepted by the device on its external interface for Media Service
Service	The device provides the following services : DeviceAdminAgent, Media, Layout, MediaStoreAgent
Handler	The code that handles a specific API message for a given service
Agent	The Handler invokes methods on the Agent for the accepted by the device on its external interface
Engine	The Agent invokes methods on the Engine that implement the services on the device
Entity	A Source or a Destination

Introduction

This document contains the description of the device interface API for media services

Chapter 1. Media Service Overview

This chapter provides the version information and a brief description of the requests support by this service

Version

Document Version = 0.0.1
Service Version = x.x.x
Software Version = 4.0.x.x

Media Services Overview

The media service is responsible for enabling the viewing/recording/playback of media
The service provides support for relaying of media streams between endpoints
The service also provides support for lookback capabilities on live sources. The viewer can thus rewind a live source
The messages described here are in conformance with the AgentMessage described in the ref[1](API-AgentMessage.pdf)

Chapter 2. Media Services API

This chapter provides details of the Media Service API

Media Service API to be supported by a Device to start a stream

These requests are typically RPC requests received by a device capable of starting/stopping a stream

- Handling requests to start/stop streams

  The request includes all the information required to start the stream
  If the Start fails, the corresponding state(START_ERROR) is returned in the response
  A Start/Stop of the stream results in Start/Stop Stream requests being sent to the MediaEngine

Example 2.1. StartStreamRequest payload received

```
<StartStreamRequestData>
 <streamNID>c7c002c0-43c0-451f-b087-56818bb4e88f</streamNID>
 <streamdatadoc>
  <!--
   Stream object goes here to be updated with destination IP/Port by the
   MediaAgentImpl
  -->
 </streamdatadoc>
 <mediasourcedatadoc>
  <!-- MediaSource object goes here -->
 </mediasourcedatadoc>
</StartStreamRequestData>
```

streamNID - unique identifier for the stream, will be used for future control operations on the stream
streamdatadoc - complete description of the stream (listed in appendix)
mediasourcedatadoc - complete description of the source (listed in appendix)

Example 2.2. StartStreamResponse payload sent

```
<StartStreamResponseData>
 <StreamStatus roomNID="136cc62f-6537-4cfd-b677-8d09d37738e2"
  streamNID="3bebcbb1-3ead-47b7-b006-a1c7f829b34e">
  <!-- See Appendix for StreamStatus element -->
 </StreamStatus>
</StartStreamResponseData>
```

Example 2.3. StartStreamResponse error payload sent

```
<Error code="2021">
 <Description>Media Source Busy</Description>
</Error>
```

Example 2.4. StopStreamRequest payload received

```
<StopStreamRequestData>
 <streamNID>c7c002c0-43c0-451f-b087-56818bb4e88f</streamNID>
</StopStreamRequestData>
```

streamNID - unique identifier for the stream, used at the time of start

Example 2.5. StopStreamResponse payload sent

```
<StopStreamResponseData>
 <streamNID>c7c002c0-43c0-451f-b087-56818bb4e88f</streamNID>
</StopStreamResponseData>
```

- Maintaining a state machine for each of the MediaStreams which is updated based on

  The notifications received from the MediaEngine
  The notifications received from the upstream relays

- The state machine action handlers are responsible for

  Sending notificatiosn to the peering destination

Example 2.6. StreamStatusEvent payload sent to the peering destination

```
<StreamStatusEventData>
 <StreamStatus roomNID="136cc62f-6537-4cfd-b677-8d09d37738e2"
  streamNID="3bebcbb1-3ead-47b7-b006-a1c7f829b34e">
  <!-- See Appendix for StreamStatus element -->
 </StreamStatus>
</StreamStatusEventData>
```

Presence Events received by Devices

A Entity is a source or a destination These requests are typically received by an entity which has setup/ needs to setup communications with a peer. This setup is done in the context of a "MediaRoom". The "MediaRoom" has a unique identifier referred to as roomNID. All resource allocations on the source/dest entities can be tied/linked to the presence of the MediaRoom

- Presence Event

  The event includes the jid of the entity and its availability value i.e Ready or Unavailable
  If a source is notified that the dest is Unavailable it must release the resources associated with the dest
  If a dest is notified that the source is Unavailable it must update its state
  If a dest is notified that the source is Ready it must resend the StartStreamRequest(s)
  If the SC is notified that a dest is Unavailable it must mark the destination as such
  If the SC is notified that a dest is Ready it must resend the PUTCanvasRequest
  If the source/dest is notified that a MediaRoom is Unavailable it must release the resources associated with it

Example 2.7. MediaPresence Event for Source/Dest received

```
<MediaPresence type="Entity"
  roomNID="136cc62f-6537-4cfd-b677-8d09d37738e2" value="Unavailable/Ready"/>
```

Example 2.8. MediaPresence Event for MediaRoom received

```
<MediaPresence type="MediaRoom"
  jid="xpi@localhost/xpi" value="Unavailable/Ready"/>
```

type - "Entity"/"MediaRoom" indicates whether the presence is for an Entity or a MediaRoom
roomNID - The unique ID of the MediaRoom
jid - The full JID of the Entity User/Device whose presence notification is recevived
value - the state of the entity, could be Ready or Unavailable.

Appendix A. Data Model

The following are sample XMLs that reflect the structure of Objects involved in the MediaRoom service.

MediaRoom Objects

The MediaRoom Objects are listed below. The corresponding xml is available under the svn. The base svn URL is "https://10.51.112.47/svn/ipvsvn/model/trunk/objectXml/MediaRoom.xml" Each of the elements mentioned below is a child element in the MediaRoom document

mediasourcedatadoc : //MediaSource/data
streamdatadoc : //Stream/data
StreamStatus : //Stream/Status

Object Details

Explanation of the use of information in the mediasourcedatadoc and streamdatadoc for starting of streams
Common : streamdatadoc, mediasourcedatadoc

roomNID : Common groupID for streams that are to be controlled as a group

Dest Decoder : streamdatadoc

destNID : Unique Global ID for the destination Port
streamURL : A pre-defined streamURL configured on the source
destUDPStartPort : Determined on the destination based on available UDP network ports
destPortElementID : Unique Local ID for the destination Port
sourceAgentJID : Contact for the source Port
windowId : Reference to the window within a canvas on the destination
canvasId : Reference to the canvas on the destination
startWallClock : Wallclock when the request was initiated
trackStateAtStart : State of the group/track at the time that request was initiated
trackOffsetAtStart : Offset from the start of the group/track at the time the request was initiated

Dest Recorder : streamdatadoc

destFileParentDirNID : The unique global ID of the directory of the recording
destFileUUID : The unique local ID of the recording file
trimLength : The length that the PVR recording that needs to be maintained which is continously trimmed

Source Live : streamdatadoc

relayNID : Unique global ID of the relay Port
sourceNID : Unique global ID of the source Port
mediaSourceNID : Unique instance ID of the mediasource on the central server
destAgentJID : Contact for the destination Port
resolvedSourceIP : The IP address of the source as seen by the destination
destType : The type of the destination i.e. Stream, File, Decoder
streamType : The type of the stream i.e. RTP, V2D, MPEGTS
sourcePortNID : The unique global ID of the source Port on the central server
resolvedDestIP : The IP address of the destination as seen by the source

profileXML : The xml description of the profile of the stream e.g. bandwidth, audio/video etc which is specific to the streamType
destIP : The pre-configured destination IP to be used by the source
callbackPort : The pre-configured destination Port to be used by the source
streamURLType : The type of URL requested, set to HTTP for RTP streams to point to an SDP file on the media server
trickPlay : Indicated whether the destination support playback at multiple speeds i.e. FastForward, Rewind

Source Live : mediasourcedatadoc

sourceNID : Unique global ID of the source on the central server
startOffset : The start offset of the media in case of a playback source
startTimecode : The start timecode of the media in case of a playback source
userRoomMediaSourceNID : This applies for multi-hop relays of multiple playback stream which are bring played in the same group and logically point to the same source. This enables the source to serve both users from the source input
sourceType : The type of the source i.e. Encoder, File
sourcePortNID : The unique global ID of the source Port in the central server
sourcePortElementID : The unique local ID of the source Port on the device
streamType : The type of the stream supported by the source e.g. RTP, V2D, MPEGTS
trackNID : A sub-group ID for the source

Management Server Collaboration : streamdatadoc

mediaDestNID : Unique instance ID for the media destination
profileNID : Unique global ID for the profile being used for the stream
streamOwnerUserJID : The user who initiated the setup of the stream
destStreamURL : The pre-configured streamURL for the destination
sourcePortNID : Unique global ID for the source port on the central server
mediarelayNID : Unique instance ID for the relay on the central server
xAuthAgentJID : Contact for the entity who needs to authorize access before a stream is started
streamGroups : Not used
sourceNetworkAddressRule : The NAT rule being used to resolve the IP addresses for source
destNetworkAddressRule : The NAT rule being used to resolve the IP addresses for destination
publishedSourceIP : The IP address published by the source
publishedDestIP : The IP address published by the destination

Playback, LIVE-PVR : mediasourcedatadoc

lookbackSourceNID : Unique global ID of the lookback source on the central server
lookbackSourceAgentJID : Contact for the lookback source
lookbackSourceFileUUID : Unique local ID of the lookback source file
lookbackSourceFileParentDirNID : Unique globacl ID of the parent directory on the central server
lookbackSourceFilePlayLength : Playlength of the lookback source file
lookbackSourceFileStartTC : Start timecode captured when the lookback source was originally recorded

StreamStatus

The StreamStatus is included in the StartResponse as well as all StreamStatus Updates

Example A.1. The StreamStatus element

```
<StreamStatus roomNID="136cc62f-6537-4cfd-b677-8d09d37738e2"
  streamNID="3bebcbb1-3ead-47b7-b006-a1c7f829b34e">
  <state>6</state>
  <URL>v2d://10.51.50.1/port=6060?bandwidth=10240000?avoption=Both?
  ismulticast=false?enablekbm=false?authscheme=TOKEN?
  authdata=fc654255-b350-432e-a95a-8a9c0d89c7e0
  </URL>
  <lastEventData>
   <!-- Last Event Data goes here -->
  </lastEventData>
  <relayData relayRoomNID="" relayStreamNID="" />
  <MediaTimeline id="">
   <Start WC="" Offset="" TC=""/>
   <Last WC="" Offset="" TC="" TimeScale="" action="" lastState=""
   requestedAction=""/>
  </MediaTimeline>
</StreamStatus>
```

roomNID - unique identifier for the room which includes this stream
streamNID - unique identifier for the stream
state - the state of the stream (Enumerated in appendix)
lastEventData - the last event received by the destination from the upstream source
relayData - the upstream relay roomNID and streamNID
MediaTimeline - the WallClock and Timecode at Start and at the Last update, used for PVR

Simple Examples of MediaAgent Implementation

The simplest example is one where the source can only return one URL. This MediaAgent implementation needs to implement the startStream method An advanced source might inspect the specifications of the stream to decide on how to configure the source and then return the URL. In this case however the source is very simple and can stream at a fixed profile. An example of a profile parameter is the bandwidth to stream at.

```
public Document startStream(DeviceI device, String requestNID,
   String streamNID, Document streamDataDoc,
   Document mediaSourceDataDoc) throws Exception {
  String streamStatusXML = FilePath.readFile(FilePath.streamStatusPath);
  Document streamStatusDoc = XMLUtils.stringToDocument(streamStatusXML);
  XMLUtils.setValueXpath(streamStatusDoc, "//state", String.valueOf(MediaAgentI.WA
  XMLUtils.setValueXpath(streamStatusDoc, "//URL",
   "v2d://10.51.50.1/port=6060?bandwidth=10240000?avoption=Both?ismulticast=false?
  return streamStatusDoc;
}
```

Sample Start Request

This is a sample start request. An encoder needs to look at a simple subset. Most of the remaining elements are related to PVR/Relay functionality All the elements of the streamdatadoc and mediasourcedatadoc have been explained in the previous section.

Example A.2. The Start Request

```
              <?xml version="1.0" encoding="UTF-8"?>
<message id="B9xmI-5218" to="tx1@localhost/tx1"
 from="0c31592d-fbf0-49ae-a7ea-e9e87c7046b1@conference.localhost/managementserver@
 type="groupchat">
 <x>
  <AgentMessage from='xpi@localhost/xpi' to='xpi@localhost/xpi' type='Request' log
   <Request>
    <header>
     <servicename>Media</servicename>
     <requestname>StartStreamRequest</requestname>
     <userJID>none</userJID>
     <requestNID>none</requestNID>
     <clientdata>none</clientdata>
    </header>
    <data>
     <StartStreamRequestData>
      <streamNID>c7c002c0-43c0-451f-b087-56818bb4e88f</streamNID>
      <streamdatadoc .../>
      <mediasourcedatadoc .../>
     </StartStreamRequestData>
    </data>
   </Request>
  </AgentMessage>
 </x>
</message>
```

Example A.3. The streamdatadoc

```
          <streamdatadoc>
  <roomNID>0c31592d-fbf0-49ae-a7ea-e9e87c7046b1</roomNID>
  <mediaSourceNID>3b4ebfe6-50a7-458e-8d3c-db1a09e661b4
  </mediaSourceNID>
  <mediaDestNID>fc6f4255-b350-432e-a95a-8a9c0d89c7e0
  </mediaDestNID>
  <profileNID>default.mediastreamprofilelist.mspl_ead3a5f2-02a7-4e8c-9d35-83
  </profileNID>
  <profileXML .../>
  <relayNID />
  <mediarelayNID />
  <destNID>default.mediastreamrelayportresourcelist.msrprl_b04c0295-dc81-488
  </destNID>
  <destPortNID>default.mediastreamrelayportresourcelist.msrprl_b04c0295-dc81
  </destPortNID>
  <destIP />
  <destUDPStartPort />
  <destAgentJID>managementserver@localhost/managementserver
  </destAgentJID>
  <destFileUUID />
  <destFileParentDirNID />
  <callbackPort />
  <destStreamURL />
  <sourceNID>default.mediastreamioportresourcelist.msioprl_61a8cafa-5d78-458
  </sourceNID>
  <sourcePortNID>default.mediastreamioportresourcelist.msioprl_61a8cafa-5d78
  </sourcePortNID>
  <xAuthAgentJID />

  <streamGroups>
   <Tag Name="track"
    Value="0c31592d-fbf0-49ae-a7ea-e9e87c7046b1.StreamTrack" />
  </streamGroups>

  <streamOwnerUserJID>managementserver@localhost/managementserver
  </streamOwnerUserJID>
  <streamURLType />
  <sourceAgentJID>xpi@localhost/xpi</sourceAgentJID>
  <startWallClock>1354814536060</startWallClock>
  <trimLength />
  <trackStateAtStart>STARTED_CURRENT</trackStateAtStart>
  <trackOffsetAtStart>799</trackOffsetAtStart>
  <sourceNetworkAddressRule />
  <destNetworkAddressRule />
  <publishedSourceIP>10.51.50.1</publishedSourceIP>
  <publishedDestIP>10.51.48.190</publishedDestIP>
  <resolvedSourceIP>10.51.50.1</resolvedSourceIP>
  <resolvedDestIP>10.51.48.190</resolvedDestIP>
  <destPortElementID />
  <destType>OUTPUT_STREAM</destType>
  <streamType>V2D</streamType>
  <streamURL />
  <windowId />
  <canvasId />
  <trickPlay>true</trickPlay>
</streamdatadoc>
```

Example A.4. The profileXML

```
  </Groups>
  <Title>V2D</Title>
  <Description>V2D</Description>
  <StreamType>V2D</StreamType>
  <Priority>1</Priority>
  <StreamActionInfo>
   <AllowPause>true</AllowPause>
   <AllowSkipback>true</AllowSkipback>
   <AllowSkipfwd>true</AllowSkipfwd>
  </StreamActionInfo>
  <Info>
   <V2DStreamConfig>
    <IsMulticast>false</IsMulticast>
    <EnableSRDTranscode>true</EnableSRDTranscode>
    <SRDTranscodeBandwidth />
    <AVOption>Both</AVOption>
   </V2DStreamConfig>
   <V2DEncoderAudioConfig>
    <SampleRate />
    <SampleSize />
    <MonoStereo />
    <AudioEncodeEnable />
   </V2DEncoderAudioConfig>
   <V2DEncoderConnectionParameterConfig>
    <BurstRatio>1</BurstRatio>
    <BurstDuration>50</BurstDuration>
    <Compression>
     <High>6</High>
     <Low>6</Low>
    </Compression>
    <Bandwidth>10000</Bandwidth>
    <Refresh>
     <Min>10</Min>
     <Max>30</Max>
    </Refresh>
    <Slice>
     <Min>0</Min>
     <Max>15</Max>
    </Slice>
    <BlockThreshold>5</BlockThreshold>
    <ColorSampling>4:2:2</ColorSampling>
    <FrameRateDivider>1</FrameRateDivider>
    <IFrameInterval />
   </V2DEncoderConnectionParameterConfig>
   <V2DEncoderConnectionParameterKbmConfig>
    <AllowKMSwitch>true</AllowKMSwitch>
    <KMIdleTime>0</KMIdleTime>
    <EnableKbm>false</EnableKbm>
   </V2DEncoderConnectionParameterKbmConfig>
   <V2DRecorderConfig>
    <FFTracks>1-256</FFTracks>
    <ThumbnailConfig>
     <Enable>true</Enable>
    </ThumbnailConfig>
   </V2DRecorderConfig>
  </Info>
 </V2DMediaStreamProfile>
</profileXML>
```

Example A.5. The mediasourcedatadoc

```
<mediasourcedatadoc>
  <roomNID>0c31592d-fbf0-49ae-a7ea-e9e87c7046b1</roomNID>
  <sourceNID>default.mediastreamioportresourcelist.msioprl_61a8cafa-5d78-458
  </sourceNID>
  <startOffset>0</startOffset>
  <startTimecode>1354814535261</startTimecode>
  <userRoomMediaSourceNID>8bfa1a10-68f2-4943-b60f-779a63fb2423
  </userRoomMediaSourceNID>
  <sourceType>ENCODER</sourceType>
  <sourcePortNID>default.mediastreamioportresourcelist.msioprl_61a8cafa-5d78
  </sourcePortNID>
  <sourcePortElementID>1</sourcePortElementID>
  <sourceAgentJID>xpi@localhost/xpi</sourceAgentJID>
  <streamType>V2D</streamType>
  <streamURL />
  <streamProfile />
  <trackNID>0c31592d-fbf0-49ae-a7ea-e9e87c7046b1.StreamTrack
  </trackNID>
  <streamTrackMediaSourceNID />
  <pvrEnabled />
  <lookbackSourceNID />
  <lookbackSourceAgentJID />
  <lookbackSourceFileUUID />
  <lookbackSourceFileParentDirNID />
  <lookbackSourceFilePlayLength />
  <lookbackSourceFileStartTC />
  <xAuthAgentJID />
  <monitorAgentJID />
</mediasourcedatadoc>
```