REST XML API

REST XML API

Table of Contents

List of Tables

List of Examples

Glossary

REST Message

The REST message module uses HTTP Headers and functions to retrieve, create, update, or delete data on a web services server based on REST architecture

Introduction

Version

Document Version = 1.0
Service Version = 2.32.3
Software Version = 4.2.x

This document contains the description of the REST XML RPC calls

Headers Example

Table 1. Sample Request Headers for every API call

Authorization	Basic c3JjcmVsYXkxQHNyY3BjMTpzcmNyZWxheTE=
Accept Header(No spaces or newlines)	application/vnd.barco.nimbus.Device+xml
Content-Type	application/xml
Resource Header	X-Barco-resource web-client_barco

- Every REST API call in the document uses this as the Base URL : http://{sc_ip_addr}/controller/{ResoucePath}?{QueryParameters}
- Authorization is basic http authorization using base64 encoded username:password
- X-Barco-resource is a custom header which accepts the resource name in the system. The resource name is of the format: 'resource_username' as shown in sample above.
- Accept Header specifies the format in which you expect a response. Some calls in the document use custom Accept Headers as shown in sample above
- Content-Type Header specifies the format in which you POST content to the system as shown in sample above
- Each call in the document specifies the ResourcePath and QueryParameters corresponding to that call.

Chapter 1. Service API

API Calls used in Service

ServiceLogin

- During this process the client's contact Info and state are updated in the Server DB
- The client version number can be obtained from the Version section of the API document corresponding to the client implementation
- ClientTemplateTitle is required to be specified. The server will create a client using the data available for the clientTemplate. Same resource should be used for the custom header X-Barco-resource
- Login API only supports a web client resource added in the system to used for clientTemplateTitle. e.g. web-client
- clientPrivateKey is an optional value in case the resource in the system has a password

Example 1.1. Service Login Request

Table 1.1. Service Login Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Service/Login
Authorization	Basic (Required)

```
<ServiceLogin clientVersion="">
  <clientPrivateKey/>
  <clientTemplateTitle/>
</ServiceLogin>

For E.g.

<ServiceLogin clientVersion="2.31.6">
  <clientTemplateTitle>webclient</clientTemplateTitle>
</ServiceLogin>
```

NOTE: If all is well, the response will contain a Long Polling URL in the response headers. Client is required to keep polling the server with an HTTP GET on this URL 'X-Barco-notification-channel'. This request requires the same set of headers as any other request. Please refer to the Introduction section above for set of required Headers for every request. The server can send asynchronous notifications to the client as response to this request. Every response comes back with a new URL in 'X-Barco-notification-channel' response header. There may or may not be notifications in the notificationlist returned in response for the long polling URL.

Example 1.2. Service Login Response

```
<ServiceLogin>
  <serverDate/>
  <clientPrivateData/>
</ServiceLogin>
```

ServiceLogout

This will make the user presence off-line and remove the user from the system's local cache. This will cause the Long-Polling channel associated with the user to be terminated. Clients that are long-polling using the URL in X-Barco-notification-channel, will see the poll requests failing and should handle them appropriately.

Example 1.3. Service Logout Request

Table 1.2. Service Logout Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Service/Logout
Authorization	Basic (Required)

Example 1.4. Service Logout Response

```
      username: User removed successfully!
```

Audit Event

It is an event to log user actions. Each event is saved as a single line (newline characters are not supported in the request) in a file on the management server. This request is an input to audit report.

This is a list of parameters for an audit event request.

- *type* [Required]
- *name* [Required]
- *level="error/warning/info/debug/trace"* [Optional]

- *target* [Optional]
- *status* [Optional]
- *text* [Required]

Example 1.5. Audit Event Request

Table 1.3. Audit Event Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Service/AuditEvent
Authorization	Basic (Required)

```
   <AuditEvent type='' level='' name='' target='' status=''>
<Description text=''/>
</AuditEvent>
```

Example 1.6. Audit Event Response

```
<AuditEvent/>
```

Get Service Client

This will return the logged in client information

Example 1.7. Get Service Client Request

Table 1.4. Get Service Client Request

Request	HTTP GET
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	whereContactJID, selectFormat
Resource Path	/Service/ServiceClient
Authorization	Basic (Required)

- *whereContactJID=""* [Optional]

Example 1.8. Get Service Client Response - BASIC Format

```
  <GetServiceClient>
 <User NID='' userName='' externalAuthentication=''>
  <UserPrivateData />
  <Groups />
 </User>
 <Asset NID='' title='' state='' presence='' type=''>
  <AssetPrivateKey>
   <KeySchema />
   <KeyData />
  </AssetPrivateKey>
  <AssetPrivateData />
  <Groups />
  <FeatureList>
   <Feature Name=""></Feature>
  </FeatureList>
 </Asset>
 <PortList>
  <Port NID='' title='' streamType='' presence='' state=''
   portType='' >
   <Groups/>
   <FeatureList>
     <Feature name=''/>
   </FeatureList>
  </Port>
 </PortList>
 <Device NID=''/>
 <Permissions>
  <User add='' modify='' delete='' read='' />
  <Device add='' modify='' delete='' read='' />
  <Profile add='' modify='' delete='' read='' />
  <Port add=''  modify='' delete='' read='' viewfrom='' streamto=''/>
  <File modify='' delete='' read='' export='' playback='' />
  <Dir update='' delete='' read='' import='' playback='' record=''/>
  <SessionConfig add='' modify='' delete='' read='' />
  <Policy add='' modify='' delete='' read=''/>
  <System upgrade='' restore='' backup='' report='' sessionmanagement='' />
 </Permissions>
</GetServiceClient>
```

Example 1.9. Get Service Client Response - EXTENDED Format

```
  <GetServiceClient>
<User NID='' userName=''>
 <UserPrivateData />
 <Groups />
</User>
<Asset NID='' title='' state='' presence='' type=''>
 <AssetPrivateKey>
  <KeySchema />
  <KeyData />
 </AssetPrivateKey>
 <AssetPrivateData/>
 <Groups />
 <FeatureList>
  <Feature Name=""></Feature>
 </FeatureList>
</Asset>
<PortList>
 <Port NID='' title='' streamType='' presence='' state=''
  portType='' >
  <Groups/>
  <FeatureList>
    <Feature name=''/>
  </FeatureList>
  <ObserverInfoList>
       <Observer xAuth='' monitor=''>
          <UserNID></UserNID>
          <ResourceNID></ResourceNID>
       </Observer>
     </ObserverInfoList>
     <StreamURL></StreamURL>
 </Port>
</PortList>
<Device NID=''>
  <System>
  <NTPClient enabled='true/false' serverIP='' />
  </System>
```

Example 1.10. Get Service Client Response - EXTENDED Format [Continue]

```
 <Services>
 <HTTPServer enabled='true/false' />
 <SSHServer enabled='true/false' />
 <TelnetServer enabled='true/false' />
 <TFTPServer enabled='true/false' />
 <XMPPServer enabled='true/false' port='' dailyBackup='Yes/No'
  backupTime='' enableExport='' recoveryMode='' />
 <XMPPAgent enabled='true/false' xmppServer='' userName=''
  domain=''>
  <AdminAgent enalbed='true/false' />
  <ConfigAgent enalbed='true/false' />
  <DestinationAgent enalbed='true/false' />
  <SourceAgent enalbed='true/false' />
  <XAuthAgent enalbed='true/false' />
  <PortCtrlAgent enalbed='true/false' />
 </XMPPAgent>
 <StreamingServer enabled='true/false'>
  <TXcntrlPorts from='' to='' />
  <V2DRxPorts from='' to='' />
  <MPEGTSRXPorts from='' to='' />
  <UDPRXPorts from='' to='' />
  <RTPRXPorts from='' to='' />
  <MulticastIP from='' to='' />
 </StreamingServer>
 <DHCPServer enabled='true/false' startIP='' endIP='' publishedDNSServer='' />
 <NTPServer enabled='true/false' serverIP='' />
</Services>
 <Network isPrimaryEnabled='' hostName='' DNSServerIP=''
 DNSName=''>
 <EthernetPortList>
  <EthernetPort  listIndex=''  isPrimary='' portID='' enabled='true/false'
   multicast='true/false' ipAddress='' useDHCP='true/false' netmask=''/>
 </EthernetPortList>
 <NATTable>
  <NAT name='' listIndex='' ip='' />
 </NATTable>
 <RouteTable>
      <Route listIndex='' destination='' netmask='' gateway=''/>
 </RouteTable>
</Network>
<Storage hardDriveEnabled='' NASEnabled=''>
 <HardDiskList>
  <HardDisk name='' mountPoint='' />
 </HardDiskList>
 <NASList>
  <NAS name='' enabled='' server='' readOnly='true/false' />
 </NASList>
</Storage>
```

Example 1.11. Get Service Client Response - EXTENDED Format[Continue]

```
 <AVCodec>
<Port id=''>
 <Displayconfig type='IPVSClientDisplayConfig'>
   <Display configurable="" hres="AUTO" vres="AUTO">
       <Panel id="" enable="" content="">
          <Geometry zIndex="" border="" vpos="0"
                hpos="0" vres="100%" hres="100%"
                aspect="" transparency="">
          </Geometry>
       </Panel>
   </Display>

   <Canvas id="" hres="1280" vres="1024">
      <Background enable="" zIndex="" content="">
         <ImageContent imageURL="" color=""></ImageContent>
         <SessionContent enableConnPanel="" enableBookmarkPanel=""
                         enableCtrlPanel="" enableHeaderPanel=""
                         enablePlayerPanel="" sessionId="">
          </SessionContent>
      </Background>

      <Layout id="LayoutId1" type="" nRow="" nCol="" colWidth=""
              rowHeight="" hpad="" vpad="" stripPos="" nStrip=""
              border="hide" borderStyle="[8px]" header="hide"
              headerStyle="basic" active="true">
      </Layout>

      <Window id="Win1" content="">
         <ConnectionContent MediaStreamIOPortTypeTitle=""
                            MediaStreamIOPortElementID=""
                            sessionId="" connId="" vres=""
                            hres="" srcKbmCtrl="">
         </ConnectionContent>
         <ThumbnailContent MediaStreamIOPortTypeTitle=""
                           MediaStreamIOPortElementID=""
                           imageURL="" refreshInterval="">
         </ThumbnailContent>
         <Properties zIndex="100" border="" vpos="0" hpos="0"
                     vres="100%" hres="100%" aspect=""
                     transparency="0" layoutId="LayoutId1"
                     layoutPos="" visibility="true"
                     maximize="false" fps="15"
                     scale="fitWindow" mute="false"
                     volume="50%" remoteKbmCtrl="false"
                     scrollHPos="0" scrollVPos="0">
         </Properties>
       </Window>
   </Canvas>
 </Displayconfig>
```

Example 1.12. Get Service Client Response - EXTENDED Format[Continue]

```
       <Displayconfig type='XP200DisplayConfig'>
        <DisplayType>Remote</DisplayType>
        <OverlayType>Manual</OverlayType>
        <Override>None</Override>
        <KeepAspect>true</KeepAspect>
        <Aspect>Auto</Aspect>
        <HorizontalTransformation>Scale/Source</HorizontalTransformation>
        <HorizontalScale>100</HorizontalScale>
        <HorizontalResolution>0</HorizontalResolution>
        <VerticalTransformation>Scale/Source</VerticalTransformation>
        <VerticalScale>100</VerticalScale>
        <VerticalResolution>0</VerticalResolution>
        <HorizontalPosition>Location</HorizontalPosition>
        <HorizontalLocation>Left</HorizontalLocation>
        <VerticalPosition>Location</VerticalPosition>
        <VerticalLocation>Top</VerticalLocation>
        <HorizontalOffset>0</HorizontalOffset>
        <VerticalOffset>0</VerticalOffset>
        <Transparency>0</Transparency>
       </Displayconfig>
      </Port>
     </AVCodec>
    </Device>
   </GetServiceClient>
```

Register User

Register user in database. Self Registration on the MS should be enabled to use this API. Use the bootstrap wizard available at http://{ms_ip_address} to enable self registration

This is a list of parameters for RegisterUser request.

- *name* [Required]

- *password* [Required]

Example 1.13. Register User Request

Table 1.5. Register User Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 500, 204, 403
Query Parameters	None
Resource Path	/Service/RegisterUser OR /RegisterUser(deprecated)

Authorization	Basic (Required)

```
<User name='' password=''>
   </User>
```

Chapter 2. Resource API

API Calls used in Resource

Parameter Listing

This is the list of common parameters in Get Requests

- *`startFrom`* [Optional]
- *`countToFetch`* [Optional]

  If the countToFetch is -1 , then it should get entire list of objects .
- *`sortBy=""`* [Optional]
- *`sortOrder="ascending/descending"`* [Optional]
- *`selectFormat="BASIC/EXTENDED"`* [Required]
- *`whereNID='nid1,nid2,nid3'`* [Optional]
- *`whereTagExpr="Name%3D'Location' and Value%3D'Sunnyvale'"`* [Optional]
- *`whereTextSubStr=""`* [Optional]
- *`subscriptionScope=ALLNODES`* [Optional]
- *`subscriptionID`* [Optional]

Device

API calls used for Devices

Get Device

Lists the devices present in the database.

Example 2.1. Get Device Request

Table 2.1. Get Device Request

Request	HTTP GET
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Device+xml"
Status	200, 401, 500
Query Parameters	sortBy, sortOrder, selectFormat, whereNID, whereTitle, whereType, whereTextSubStr, whereTagExpr
Resource Path	/Resource/Device

Authorization	Basic (Required)

- *sortBy="title/type"* [Optional]
- *sortOrder="ascending/descending"* [Optional]
- *whereNID* [Optional]
- *whereTitle* [Optional]
- *whereType="IPVSClient-Device/Proxy-Device/WebClient-Device/DMS-25-Device/DMS-100-Device/DMS-120-Device/DMS-200-Device/ DMS-1000-Device/DMS-1200-Device/DMS-1500-Device/DMS-1600-Device/DMS-1700-Device/ MPEGTS-Rx-Device/MPEGTS-Tx-Device/MS-25-Device/MS-Device/ RTP-Tx-Device/RTP-Rx-Device/ UDP-Tx-Device/UDP-Rx-Device/V2D-Tx-Device/V2D-Rx-Device/V2D-XP100-Device/V2D-XP200-Device/ V2D-XP200S-Device/V2D-XP220-Device/V2D-XP220S-Device/V2D-XPi-Device"* [Optional]
- *whereTextSubStr* [Optional]
- *whereTagExpr="Name%3D'Location' and Value%3D'Sunnyvale'"* [Optional]
- *subscriptionID* [Optional]

Example 2.2. Get Device Response - BASIC Format

```
<Resource from='0' total='0'>
 <Device NID='' title='' type='' description='' presence='' state='' password=''
  unManaged='true/false' unManned='true/false' msStatus='Active/Primary/PrimaryS
  <Groups>
   <Tag name='' value='' />
  </Groups>
  <DevicePrivateData />
  <TemplateNID />
  <PortList>
   <Port NID='' portType='' deviceElementID='' />
  </PortList>
  <Status device='' systemElements='' serviceElements='' hardDisk='' NAS='' netw
  <DeviceList>
   <Device NID='' title='' type='WebClient-Device' description='' presence='' st
  unManaged='true/false' unManned='true/false' msStatus='Active/Primary/PrimaryS
    <Groups>
     <Tag name='' value='' />
    </Groups>
    <DevicePrivateData />
    <TemplateNID />
    <PortList>
     <Port NID='' portType='' deviceElementID='' />
    </PortList>
    <Status device='' systemElements='' serviceElements='' hardDisk='' NAS='' ne
   </Device>
  </DeviceList>
 </Device>
</Resource>
```

AddDevice

Add device in database

This is a list of parameters for add device request.

- *`type`* [Required]
- *`title`* [Required[Does not allow XML restricted special characters]]
- *`description`* [optional]
- *`password`* [Required]
- *`unManned`* [Required]
- *`unManaged`* [Required]
- *`applyTagsToPorts`* [Required]

- *Groups* [optional]
- *DevicePrivateData* [optional]

Example 2.3. Add Device Request

Table 2.2. Add Device Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Device+xml"
Status	201, 401, 500
Query Parameters	None
Resource Path	/Resource/Device
Authorization	Basic (Required)

```
<Device type='' title='' description='' password='' unManaged='true/false' unMa
 <Groups>
  <Tag Name='' Value='' />
 </Groups>
 <DevicePrivateData />
</Device>
```

Example 2.4. Add Device Response

```
   <Resource>
  <!-- Check Device Object Response XML-->
</Resource>
```

DeleteDevice

Delete device from database

Example 2.5. Delete Device Request

Table 2.3. Delete Device Request

Request	HTTP DELETE
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Device+xml"
Status	200, 401, 500
Query Parameters	NID or title

Resource Path	/Resource/Device
Authorization	Basic (Required)

Example 2.6. Delete Device Response

```
<Resource>
   <!-- Check Device Object Response XML-->
</Resource>
```

SetDeviceInfo

Update device description in database

Example 2.7. Set Device Info Request

Table 2.4. Set Device Info Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Device-Info+xml"
Status	200, 401, 500
Query Parameters	NID or title
Resource Path	/Resource/Device
Authorization	Basic (Required)

```
<DeviceInfo description='' password='' applyTagsToPorts='true/false'>
 <Groups>
  <Tag Name='' Value='' />
 </Groups>
 <DevicePrivateData />
</DeviceInfo>
```

Example 2.8. Set Device Info Response

```
<Resource>
 <DeviceInfo>
  <!-- Check Device Object Response XML-->
 </DeviceInfo>
</Resource>
```

Device Object Response XML

Example 2.9. Device Object Response XML

```
<Device NID='' title='' type='' description='' presence='' state='' softwareVers
 <Groups>
  <Tag name='' value='' />
 </Groups>
```

Example 2.10. Device Object responses XML [Continued]

```
<DevicePrivateData />
<PortList>
 <Port NID='' title='' streamType='' presence='' state='' portType=''>
  <Groups />
  <FeatureList>
   <Feature name='' />
  </FeatureList>
  <ObserverInfoList>
       <Observer xAuth='' monitor=''>
          <UserNID></UserNID>
          <ResourceNID></ResourceNID>
       </Observer>
     </ObserverInfoList>
     <StreamURL></StreamURL>
 </Port>
</PortList>
<DeviceConfig NID=''>
 <System>
  <NTPClient enabled='true/false' serverIP='' />
 </System>
 <Services>
  <HTTPServer enabled='true/false' />
  <SSHServer enabled='true/false' />
  <TelnetServer enabled='true/false' />
  <TFTPServer enabled='true/false' />
  <XMPPServer enabled='true/false' port='' dailyBackup='Yes/No' backupTime='' e
  <XMPPAgent enabled='true/false' xmppServer='' userName='' domain=''>
   <AdminAgent enabled='true/false' />
   <ConfigAgent enabled='true/false' />
   <DestinationAgent enabled='true/false' />
   <SourceAgent enabled='true/false' />
   <XAuthAgent enabled='true/false' />
   <PortCtrlAgent enabled='true/false' />
  </XMPPAgent>
  <StreamingServer enabled='true/false'>
   <TXcntrlPorts from='' to='' />
   <V2DRxPorts from='' to='' />
   <MPEGTSRXPorts from='' to='' />
   <UDPRXPorts from='' to='' />
   <RTPRXPorts from='' to='' />
   <MulticastIP from='' to='' />
  </StreamingServer>
  <DHCPServer enabled='true/false' startIP='' endIP='' publishedDNSServer='' />
  <NTPServer enabled='true/false' serverIP='' />
 </Services>
```

Example 2.11. Device Object responses XML [Continued]

```
<Network isPrimaryEnabled='' hostName='' DNSServerIP='' DNSName=''>
 <EthernetPortList>
  <EthernetPort isPrimary='' portID='' enabled='true/false' multicast='true/fa
 </EthernetPortList>
 <NATTable>
  <NAT name='' listIndex='' ip='' />
 </NATTable>
</Network>
<Storage hardDriveEnabled='' NASEnabled=''>
 <HardDiskList>
  <HardDisk name='' mountPoint='' />
 </HardDiskList>
 <NASList>
  <NAS name='' enabled='' server='' readOnly='true/false' />
 </NASList>
</Storage>
   <AVCodec>
 <Port id=''>
  <Displayconfig type='IPVSClientDisplayConfig'>
   <Display configurable="" />
   <Canvas>
    <Background enable='true/false' zIndex='0' content='OS/Image/Session'>
     <ImageContent imageURL="" color="" />
     <SessionContent enableConnPanel="" enableBookmarkPanel="" enableCtrlPanel
    </Background>
    <Window id='userDefinedStr' enable='true/false' content='MediaThumbnail/Me
     <ConnectionContent MediaStreamIOPortTypeTitle="" MediaStreamIOPortElement
     <ThumbnailContent MediaStreamIOPortTypeTitle="" MediaStreamIOPortElementI
     <Geometry zIndex="" border="NONE/THIN/THICK" vpos='ALIGNTOP/ALIGNMIDDLE/A
      aspect='SourceWidth/SourceHeight/Auto/None' transparency="NNN%" />
    </Window>
   </Canvas>
  </Displayconfig>
```

Example 2.12. Device Object responses XML [Continued]

```
    <Displayconfig type='XP200DisplayConfig'>
     <DisplayType>Remote</DisplayType>
     <OverlayType>Manual</OverlayType>
     <Override>None</Override>
     <KeepAspect>true</KeepAspect>
     <Aspect>Auto</Aspect>
     <HorizontalTransformation>Scale/Source</HorizontalTransformation>
     <HorizontalScale>100</HorizontalScale>
     <HorizontalResolution>0</HorizontalResolution>
     <VerticalTransformation>Scale/Source</VerticalTransformation>
     <VerticalScale>100</VerticalScale>
     <VerticalResolution>0</VerticalResolution>
     <HorizontalPosition>Location</HorizontalPosition>
     <HorizontalLocation>Left</HorizontalLocation>
     <VerticalPosition>Location</VerticalPosition>
     <VerticalLocation>Top</VerticalLocation>
     <HorizontalOffset>0</HorizontalOffset>
     <VerticalOffset>0</VerticalOffset>
     <Transparency>0</Transparency>
    </Displayconfig>
   </Port>
  </AVCodec>
 </DeviceConfig>
</Device>
```

Port

API calls used for Ports

Get Port

This is the list of Source and Destination Ports that the Client has access to. Refer to the Appendix for the format of the Port object

Example 2.13. Get Port List Request

Table 2.5. Get Port List Request

Request	HTTP GET
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Port+xml"
Status	200, 401, 500
Query Parameters	startFrom, countToFetch, sortOrder, sortBy, selectFormat, whereNID, whereTextSubStr, whereTagExpr, whereStreamType, wherePermissions, subscriptionID, dstPortTitle,whereState,subscriptionScope,expireTime

Resource Path	/Resource/Port
Authorization	Basic (Required)

- *Parameters are optional*
- *sortBy="title/state"*
- *sortBy="NID/portType/presence"* [Not Supported]
- *selectFormat="BASIC/EXTENDED/EXTENDEDCONNECTIONINFO/ BASICCONNECTIONINFO/BASICURL/EXTENDEDURL"*
- *type="SrcPort/DstPort"*
- *whereStreamType="V2D/RTP/MPEGTS/UDP/RTP-RAW/RTP-PCM/RFB"*
- *wherePermissions="ADD,MODIFY,DELETE,VIEWFROM,STREAMTO"*
- *dstPortTitle* [Optional]
- *whereState="Ready/Busy/Error/NotReady/Offline"* [Optional]
- *expireTime* [Optional. expireTime should be in Seconds format. For permanent URL use INFINITE keyword in expireTime. For perminent URL use INFINITE keyword in expireTime]
- *whereTagExpr="Name%3D'Location' and Value%3D'Sunnyvale'"* [Check for correct URL encoding when using this]

```
For e.g.

To get a list of Source Ports: HTTP GET http:/<msip>/controller/Resource/Por

To get a list of Destination Ports: HTTP GET http:/<msip>/controller/Resourc
```

Example 2.14. NOTE about fetching URLs from the System

For all URLs that the system returns, if the destination information is not provided or is incorrect, in the 'dstPortTitle' parameter, then the URL will be for primary MS IP address. For mediaURL, if destination information is not provided, then logged-in resource will be used as destination. If this logged-in resource happens to be a 'Proxy-Client' the mediaURL may not work. For All URL's, if expireTime is not provided system will use logged in user credentials to generate validation signature. Validity of URL is tied to logged in user session. If user logged out URL is no longer valid If expireTime is provided system will generate expiry based valudation signature. Validity of URL is tied to expireTime. After expiry time URL is no longer valid. For permanent URL's send INFINITE keyword in expireTime.

Example 2.15. Get Port Response - BASIC Format

```
<Resource type="SrcPort/DstPort" from="0" total="50">
  <Port portType="" NID="" title="" streamType="" presence="" state="" portId="">
   <Permissions add='' modify='' delete='' read='' viewfrom='' streamto=''/>
  </Port>
  <Port portType="" NID="" title="" streamType="" presence="" state="" portId="">
   <Permissions add='' modify='' delete='' read='' viewfrom='' streamto=''/>
  </Port>
  <Port portType="" NID="" title="" streamType="" presence="" state="" portId="">
   <Permissions add='' modify='' delete='' read='' viewfrom='' streamto=''/>
  </Port>
</Resource>
```

Example 2.16. Get Port Response - EXTENDED Format

```
<Resource from="0" total="50">
  <Port NID="" title="" streamType="" presence="" state="" portType="" portId="">
   <description />
   <Groups>
    <Tag name="" value="" />
   </Groups>
   <Permissions add='' modify='' delete='' read='' viewfrom='' streamto=''/>
   <Asset NID='' title='' deviceType='' contactJID=''/>
   <MaxInstanceCount/>
  <InstanceList>
           <Instance streamURL="" sessionId="" mode="Dynamic">
               <ProfileXML>
     </ProfileXML>
           </Instance>
         </InstanceList>
  <FeatureList>
                   <Feature name=""/>
            </FeatureList>
 <ObserverInfo>
           <Observer xAuth="" monitor="" userNID="" resourceNID=""/>
       </ObserverInfo>

 </Port>
</Resource>
```

Example 2.17. Get Port Response - BASICCONNECTIONINFO Format

```
<Resource type="SrcPort/DstPort/RelayPort/StorePort" from="0"
 total="0">
 <Port NID='' title='' streamType='' presence='' state='' portType='' portId=''th
  <Asset NID='' title='' deviceType='' contactJID='' />
  <PVRInfo enable='' trimLength='' mediaStreamFileResourceNID='' mediaStreamDirRe
  <Description />
  <Groups>
   <Tag Name='' Value='' />
  </Groups>
  <Permissions add='' modify='' delete='' read='' viewfrom='' streamto=''/>
  <MaxInstanceCount/>
  <FeatureList>
               <Feature name=""/>
           </FeatureList>
```

Example 2.18. Get Port Response - BASICCONNECTIONINFO Format [Continued]

```
           <ObserverInfo>
   <Observer xAuth='' monitor='' userNID='' resourceNID='' />
  </ObserverInfo>
  <ConnectionStatus>
   <PortSessionInfo id='' name=''>
    <PortConnectionInfo id='' name='' state=''  type='VIEW/PLAYBACK/RECORD/PVR'
     srcNID='' srcTitle='' dstNID='' dstTitle='' errorCode='' errorData='' window
   </PortSessionInfo>
  </ConnectionStatus>
  <InstanceList>
   <Instance streamURL='' />
  </InstanceList>
 </Port>
</Resource>
```

Example 2.19. Get Port Response - EXTENDEDCONNECTIONINFO Format

```
<Resource type="SrcPort/DstPort/RelayPort/StorePort" from="0"
 total="0">
 <Port NID='' title='' streamType='' presence='' state='' portType='' portId='' t
  <Asset NID='' title='' deviceType='' contactJID='' />
  <PVRInfo enable='' trimLength='' mediaStreamFileResourceNID='' mediaStreamDirRe
  <Description />
  <Groups>
   <Tag Name='' Value='' />
  </Groups>
  <Permissions add='' modify='' delete='' read='' viewfrom='' streamto=''/>
  <MaxInstanceCount/>
  <FeatureList>
               <Feature name=""/>
           </FeatureList>
           <ObserverInfo>
   <Observer xAuth='' monitor='' userNID='' resourceNID='' />
  </ObserverInfo>
  <ConnectionStatus>
   <PortSessionInfo id='' name=''>
    <PortConnectionInfo id='' name='' state=''  type='VIEW/PLAYBACK/RECORD/PVR'
      srcNID='' srcTitle='' dstNID='' dstTitle='' errorCode='' errorData='' windo
         srcUserJID='' srcUserName='' destUserJID='' destUserName='' thumbnailURL
```

Example 2.20. Get Port Response - EXTENDEDCONNECTIONINFO Format [Continued]

```
      <Profile id='' profileName='' profileNID='' allowKMSwitch='' remoteKM='' av
         maxBandwidth='' lowCompression='' highCompression='' burstRatio=''
      burstDuration='' unicast='' multicast='' />
    </PortConnectionInfo>
   </PortSessionInfo>
  </ConnectionStatus>
  <InstanceList>
   <Instance streamURL='' />
  </InstanceList>
 </Port>
</Resource>
```

Example 2.21. Get Port Response - BASICURL Format

```
<Resource type="SrcPort/DstPort" from="0" total="50">
  <Port portType="" NID="" title="" streamType="" presence="" state="" portId=""
   <Permissions add='' modify='' delete='' read='' viewfrom='' streamto=''/>
  </Port>
</Resource>
```

Example 2.22. Get Port Response - EXTENDEDURL Format

```
<Resource type="SrcPort/DstPort" from="0" total="50">
  <Port portType="" NID="" title="" streamType="" presence="" state="" portId=""
   <description />
   <Groups>
    <Tag name="" value="" />
   </Groups>
   <Permissions add='' modify='' delete='' read='' viewfrom='' streamto=''/>
   <Asset NID='' title='' deviceType='' contactJID=''/>
   <MaxInstanceCount/>
   <FeatureList>
                   <Feature name=""/>
               </FeatureList>
  </Port>
</Resource>
```

Get Port Display Configuration Request

Gets the display configuration for given port.

Example 2.23. Get Port Display Configuration Request

Table 2.6. GetPortDisplayConfig Request

Request	HTTP GET
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Port-DisplayConfig+xml"
Status	200, 401, 500
Query Parameters	whereNID, whereTextSubStr, subscriptionID
Resource Path	/Resource/Port
Authorization	Basic (Required)

Example 2.24. Get Port Display Configuration Response

```
<Resource>
  <PortDisplayConfig NID="" title="">
     <!--Check the appendix A for device specific config-->
        </PortDisplayConfig>
      </Resource>
```

Get Port Display Status Request

Gets the display status for given port.

Example 2.25. Get Port Display Status Request

Table 2.7. GetPortDisplayStatus Request

Request	HTTP GET
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Port-DisplayStatus+xml"
Status	200, 401, 500
Query Parameters	whereNID, whereTextSubStr, subscriptionID
Resource Path	/Resource/Port
Authorization	Basic (Required)

Example 2.26. Get Port Display Status Response

```
<Resource>
 <PortDisplayStatus NID='' title='' >
  <!--Check the appendix A for device specific status-->
 </PortDisplayStatus>
</Resource>
```

Get Port Tags

This is the list of Tags that the Client has access to.

Example 2.27. Get Port Tags Request

Table 2.8. Get Port Tags Request

Request	HTTP GET
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Port-Tags+xml"
Status	200, 401, 500

Query Parameters	sortBy, sortOrder, whereNameTextSubString, whereValueTextSubString, wherePermissions, selectFormat
Resource Path	/Resource/Port
Authorization	Basic (Required)

- *sortBy="Name/Value"*
- *whereNameTextSubString=""* [Optional]
- *whereValueTextSubString* [Optional]
- *wherePermissions="ADD,MODIFY,DELETE,VIWEFROM,STREAMTO"*
- *selectFormat="NameList/NameValueList/NameValueTree"* [Default=NameValueList]

Example 2.28. Get Port Tags Response - NameList Format

```
   <Resource from="0" total="0">
    <PortTags>
  <Tag Name="Location"/>
  <Tag Name="Network"/>
    </PortTags>
   </Resource>
```

Example 2.29. Get Port Tags Response - NameValueList Format

```
  <Resource from="0" total="0">
    <PortTags>
  <Tag Name="Location" Value="Sunnyvale"/>
  <Tag Name="Location" Value="Hyderabad"/>
 </PortTags>
</Resource>
```

Example 2.30. Get Port Tags Response - NameValueTree Format

```
  <Resource from="0" total="0">
    <PortTags>
  <Tag Name="Location">
      <Value>Sunnyvale</Value>
      <Value>Hyderabad</Value>
  </Tag>
 </PortTags>
</Resource>
```

SetPortInfo

Update port title, description, and groups in the database

Example 2.31. Set Port Info Request

Table 2.9. Set Port Info Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Port-Info+xml"
Status	200, 401, 500
Query Parameters	NID or title
Resource Path	/Resource/Port
Authorization	Basic (Required)

```
For e.g.
To set Port info using NID: HTTP POST http:/<msip>/controller/
Resource/Port?NID=default.mediastreamsrcportresourcelist.mssprl_091145e5-a419

To set Port info using title: HTTP POST http:/<msip>/controller/
Resource/Port?title=rtp
```

- *`title`* [Required [Does not allow XML restricted special characters]]
- *`description`* [Optional]
- *`Groups`* [Optional]

```
    <PortInfo title='' description=''>
 <Groups>
  <Tag Name='' Value='' />
 </Groups>
</PortInfo>
```

Example 2.32. Set Port Info Response

```
    <Resource>
  <PortInfo>
     <!-- Check Port Object Response XML-->
  </PortInfo>
 </Resource>
```

SetPortObserverInfo

Update observer information in the database

Example 2.33. Set Port Observer Info Request

Table 2.10. Set Port Observer Info Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Port-ObserverInfo +xml"
Status	200, 401, 500
Query Parameters	NID or title
Resource Path	/Resource/Port
Authorization	Basic (Required)

- *`xAuth(true/false)`* [Optional]
- *`monitor(true/false)`* [Optional]
- *`UserNID`* [Required. if xAuth/monitor present. NID of the User to be set as an Observer]
- *`ResourceNID`* [Required. if xAuth/monitor present. NID of an asset resource in the system]

```
    <PortObserverInfo >
  <Observer xAuth='' monitor=''>
      <UserNID></UserNID>
         <ResourceNID></ResourceNID>
       </Observer>
 </PortObserverInfo>
```

Example 2.34. Set Port Observer Info Response

```
        <Resource>
      <PortObserverInfo>
         <!-- Check Port Object Response XML-->
       </PortObserverInfo>
    </Resource>
```

SetPortPVRInfo

Update the PVR settings in the database

Example 2.35. Set Port PVR Info Request

Table 2.11. Set Port PVR Info Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Port-PVRInfo +xml"
Status	200, 401, 500
Query Parameters	NID or title
Resource Path	/Resource/Port
Authorization	Basic (Required)

- *`enable(true/false)`* [Required]
- *`trimLength`* [Optional - default is 4hrs]
- *`mediaStreamDirResourceNID`* [Required]

```
<PortPVRInfo enable='true/false' trimLength='' mediaStreamDirResourceNID=''
```

Example 2.36. Set Port PVR Info Response

```
        <Resource>
      <PortPVRInfo>
        <!-- Check Port Object Response XML-->
      </PortPVRInfo>
     </Resource>
```

Port Object Response XML

Example 2.37. Port Object Response XML

```
 <Port NID='' title='' streamType='' presence='' state='' portType=''>
<Asset NID='' title='' deviceType='' deviceNID=''/>
<PVRInfo enable='' trimLength='' mediaStreamFileResourceNID='' mediaStreamDir
<Description />
<Groups>
 <Tag Name='' Value='' />
</Groups>
<Permissions add='' modify='' delete='' read='' viewfrom='' streamto=''/>
<MaxInstanceCount/>
<FeatureList>
                <Feature name=""/>
            </FeatureList>
<ObserverInfo>
 <Observer xAuth='' monitor='' userNID='' resourceNID='' />
</ObserverInfo>
<InstanceList>
 <Instance streamURL='' />
</InstanceList>
 </Port>
```

SetPortStreamURL

Update port stream URL in the database

Example 2.38. Set Port StreamURL Request

Table 2.12. Set Port StreamURL Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Port-StreamURL +xml"
Status	200, 401, 500
Query Parameters	NID or title
Resource Path	/Resource/Port
Authorization	Basic (Required)

- *`streamURL`* [Optional]

```
<PortStreamURL streamURL='' />
```

Example 2.39. Set Port StreamURL Response

```
    <Resource>
  <PortStreamURL>
     <!-- Check Port Object Response XML-->
  </PortStreamURL>
 </Resource>
```

Set Port Display Configuration Request

updates the given display configuration.

Tip

Suggest get config from the database update the required parameters and send the set port display configuration request

Example 2.40. Set Port Display Configuration Request

Table 2.13. Set Port Display Configuration Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Port-DisplayConfig+xml"
Status	200, 401, 500
Query Parameters	NID or title
Resource Path	/Resource/Port
Authorization	Basic (Required)

```
<PortDisplayConfig byNID='' byTitle='' >
      <!--Check the appendix for device specific config-->
</PortDisplayConfig>
```

Example 2.41. Set Port Display Configuration Response

```
<PortDisplayConfig>
</PortDisplayConfig>
```

Set Port Window Properties

Sets the unmanned client window Properties. Note: Send all properties or one or combination.

Example 2.42. Set Port Window Properties Request

Table 2.14. Set Port Window Properties Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Port-WindowProperties+xml"
Status	200, 401, 500
Query Parameters	NID
Resource Path	/Resource/Port
Authorization	Basic (Required)

```
<PortWindowProperties bySessionId='' byConnId=''
    byNID='' byCanvasId='' byWindowId='' byLayoutId='' >
 <WindowProperties fps='' maximize='true/false' mute='true/false'
   remoteKbmCtrl='true/false' scale='' visibility='true/false'
   volume='50%' aspect='' border='' hpos='' vpos='' hres=''
   vres='' layoutPos='' scrollHPos='' scrollVPos='' transparency=''/>
</PortWindowProperties>
```

- *`fps`* [optional]
- *`maximize='true/false''`* [optional]
- *`mute='true/false'`* [optional]
- *`remoteKbmCtrl='true/false'`* [optional]
- *`scale`* [optional]
- *`visibility='true/false'`* [optional]
- *`volume`* [optional]
- *`aspect`* [optional]
- *`border`* [optional]
- *`hpos`* [optional]
- *`vpos`* [optional]
- *`hres`* [optional]
- *`vres`* [optional]
- *`layoutPos`* [optional]
- *`scrollHPos`* [optional]
- *`scrollVPos`* [optional]
- *`transparency`* [optional]

Example 2.43. Set Port Window Properties Response

```
   <PortWindowProperties>
</PortWindowProperties>
```

Dir

API calls used for Dir.

Get Dir

This is the list of Media Directories that the Client has access to.

Example 2.44. Get Dir Request

Table 2.15. Get Dir Request

Request	HTTP GET
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Dir+xml"
Status	200, 401, 500
Query Parameters	countToFetch, selectFormat, sortBy, sortOrder, startFrom, whereDateRange, whereNID, whereTagExpr, whereTextSubStr, whereParentDirNID, wherePermissions, whereFilePermissions, subscriptionID, subscriptionScope, expireTime, dstPortTitle
Resource Path	/Resource/Dir

Authorization	Basic (Required)

- *`sortBy="title/state/size/sizeUsed/dateCreated/dateModified"`*
- *`sortBy="NID"`* [Not Supported]
- *`whereNID="DirNID"`* [dirNID Required]
- *`whereParentDirNID="parentDirNID"`* [Optional]
- *`wherePermissions="MODIFY,DELETE,IMPORT,RECORD"`* [Optional]
- *`whereFilePermissions="MODIFY,DELETE,EXPORT,PLAYBACK"`* [Optional]
- *`subscriptionID`* [Optional]
- *`selectFormat="BASIC/SNAPSHOTURL(Deprecated)/UPLOADURL"`* [Optional. default='BASIC']
- *`expireTime`* [Optional. expireTime should be in Seconds format. Used along with UPLOADURL selectFormat. For permanent URL use INFINITE keyword in expireTime]
- *`whereDateRange="From:YYYY-MM-DD HH:MM:SS.MMMM, To:YYYY-MM-DD HH:MM:SS.MMMM, Interval:xx(Y/M/D/H/m/S/s)"`* [Optional]
- *`dstPortTitle`* [Optional]

Example 2.45. Get Dir Response - BASIC Format

```
<GetDir from="0" total="0">
 <Dir NID='' title='' dirType='' state='' size='' sizeUsed='' dateCreated='' dat
  <Groups>
   <Tag Name="" Value="" />
  </Groups>
  <Permissions modify='' delete='' read='' import='' playback='' record='' />
 </Dir>
</GetDir>
```

Example 2.46. Get Dir Response - SNAPSHOTURL Format (Deprecated)

```
<GetDir from="0" total="0">
 <Dir NID='' title='' dirType='' state='' size='' sizeUsed='' dateCreated='' dat
  <Groups>
   <Tag Name="" Value="" />
  </Groups>
  <Permissions modify='' delete='' read='' import='' playback='' record='' />
 </Dir>
</GetDir>
```

Example 2.47. Get Dir Response - UPLOADURL Format

```
<GetDir from="0" total="0">
 <Dir NID='' title='' dirType='' state='' size='' sizeUsed='' dateCreated='' dat
  <Groups>
   <Tag Name="" Value="" />
  </Groups>
  <Permissions modify='' delete='' read='' import='' playback='' record='' />
 </Dir>
</GetDir>
```

Get Dir Tags

This is the list of Tags that the Client has access to.

Example 2.48. Get Dir Tags Request

Table 2.16. Get Dir Tags Request

Request	HTTP GET
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Dir-Tags+xml"
Status	200, 401, 500
Query Parameters	sortBy, sortOrder, whereNameTextSubStr, whereValueTextSubStr, wherePermissions, selectFormat
Resource Path	/Resource/Dir
Authorization	Basic (Required)

- *sortBy="Name/Value"*
- *whereNameTextSubString=""* [Optional]
- *whereValueTextSubString* [Optional]
- *wherePermissions="MODIFY,DELETE,IMPORT,RECORD"* [Optional]
- *selectFormat="NameList/NameValueList/NameValueTree"* [Default=NameValueList]

Example 2.49. Get Dir Tags Response - NameList Format

```
  <DirTags from="0" total="0">
<Tag Name="Location"/>
<Tag Name="Network"/>
  </DirTags>
```

Example 2.50. Get Dir Tags Response - NameValueList Format

```
  <GetDirTags from="0" total="0">
 <Tag Name="Location" Value="Sunnyvale"/>
 <Tag Name="Location" Value="Hyderabad"/>
</GetDirTags>
```

Example 2.51. Get Dir Tags Response - NameValueTree Format

```
  <GetDirTags from="0" total="0">
 <Tag Name="Location">
    <Value>Sunnyvale</Value>
    <Value>Hyderabad</Value>
 </Tag>
</GetDirTags>
```

File

API calls used for File

Get File

This is the list of Files that the Client has access to. If there are Files of type MediaGroup then the associated MediaClip Files will be returned as nested sub-elements of the Media Group File.

Example 2.52. Get File Request

Table 2.17. Get File Request

Request	HTTP GET
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.File+xml"
Status	200, 401, 500
Query Parameters	countToFetch, selectFormat, sortBy, sortByMetaData, sortByTagName, sortOrder, startFrom, whereType, whereMediaGroupID, whereDateRange, whereNID, whereTagExpr, whereTextSubStr, searchTextSubStrIn, whereState, whereMetaDataExpr, whereParentDirNID, whereSessionConfigNID, wherePermissions, subscriptionID, subscriptionScope, expireTime, dstPortTitle, whereDownloadState
Resource Path	/Resource/File

Authorization	Basic (Required)

- *searchTextSubStrIn=TITLE,DESCRIPTION,METADATA,BOOKMARK* [Used along with whereTextSubStr]
- *sortBy="title/state/playlength/dateCreated/size"* [Either sortBy/ sortByMetaData/sortByTagName]
- *sortByMetaData* [Either sortBy/sortByMetaData/sortByTagName]
- *sortByTagName* [Either sortBy/sortByMetaData/sortByTagName]
- *sortBy="NID"* [Not Supported]
- *whereParentDirNID=""* [Optional]
- *whereState="Ready/Busy/Error/NotReady/Offline"* [Optional]
- *whereMetaDataExpr="Name%3D'TestAPI' and Value%3D'RecordTest'"* [Optional]
- *whereType="MediaClip,MediaGroup,MediaImage,ALL"* [Optional Default=ALL]
- *whereMediaGroupID=""* [Optional]
- *whereSessionConfigNID=""* [Optional]
- *wherePermissions="MODIFY,DELETE,PLAYBACK,IMPORT"* [Optional]
- *subscriptionID* [Optional]
- *expireTime* [Optional. expireTime should be in Seconds format. Used along with BASICURL or EXTENDEDURL selectFormat. For permanent URL use INFINITE keyword in expireTime]
- *selectFormat="BASIC/EXTENDED/BASICURL/EXTENDEDURL/ BASICCONNECTIONINFO"* [Optional. Default=BASIC]
- *dstPortTitle* [Optional]
- *whereDateRange="From:YYYY-MM-DD HH:MM:SS.MMMM Z, To:YYYY-MM-DD HH:MM:SS.MMMM Z, Interval:xx(Y/M/D/H/m/S/s)"* [Optional]

  For example from 2014-01-01 00:00:00.0000 +05:30 to 2014-01-02 00:00:00.0000 +05:30 (give value in percent encoded form as below):

  1. whereDateRange="From:2014-01-01%2000%3A00%3A00.000%20%2B05%3A30, To:2014-01-02%2000%3A00%3A00.000%20%2B05%3A30"

  2. whereDateRange="From:2014-01-01%2000%3A00%3A00.000%20%2B05%3A30, Interval:01D"

  3. whereDateRange="To:2014-01-02%2000%3A00%3A00.000%20%2B05%3A30, Interval:01D"
- *whereDownloadState="Ready/Busy/NotReady"* [Optional (filter for cached files)]
- **Refer to Example 2.17 for Note related to URLs**

```
For e.g. Get all the recorded mediaclips:

http://<msip>/controller/Resource/File?whereType=Mediaclip
```

Example 2.53. Sample sortByTagName="Location"

```
<Groups>
   <Tag Name="Location" Value="Sunnyvale"></Tag>
</Groups>
```

Example 2.54. Sample sortByMetaData="TestAPI"

```
<MetaData>
   <Tag Name="TestAPI">RecordTest</Tag>
</MetaData>
```

Example 2.55. Get File Response - EXTENDED Format

```
<Resource from="200" total="400">
  <File NID='' title='' fileType='MediaClip' streamType='' size=''
   playlength='' state='' dateCreated='' dateModified=''
   readOnly='' mediaGroupId='' ffTracks='' bypassAutoStorageRules=''>
   <Groups>
    <Tag Name="" Value="" />
   </Groups>
   <Permissions modify='' delete='' read='' export='' playback='' />
   <MetaData>
    <Tag Name="">Value</Tag>
   </MetaData>
  </File>
  <File NID='' title='' fileType='MediaGroup' streamType='' size=''
   playlength='' presence='' state='' dateCreated='' dateModified=''
   readOnly='' mediaGroupId='' ffTracks='' bypassAutoStorageRules=''>
   <Groups>
    <Tag Name="" Value="" />
   </Groups>
   <Permissions modify='' delete='' read='' export='' playback='' />
   <MetaData>
    <Tag Name="">Value</Tag>
   </MetaData>
   <!-- Files of type MediaGroup will have nested File records -->
   <FileList>
    <File NID='' title='' fileType='MediaClip' streamType='' size=''
    playlength='' presence='' state='' dateCreated='' dateModified=''
    readOnly='' mediaGroupId='' ffTracks='' bypassAutoStorageRules='' >
```

Example 2.56. Get File Response - EXTENDED Format [Continued]

```
     <Groups>
      <Tag Name="" Value="" />
     </Groups>
     <Permissions modify='' delete='' read='' export='' playback='' />
     <MetaData>
      <Tag Name="">Value</Tag>
     </MetaData>
    </File>
   </FileList>
  </File>
</Resource>
```

Example 2.57. Get File Response - BASIC Format

```
<Resource from="200" total="400">
  <File fileType='MediaClip' NID="" title="" mediaGroupId="" streamType=""
      size="" playlength="" presence="" state="" dateCreated=""
      dateModified="" readOnly="" ffTracks='' bypassAutoStorageRules=''>
   <description />
   <Groups>
    <Tag Name="" Value="" />
   </Groups>
   <Permissions modify='' delete='' read='' export='' playback='' />
   <MetaData>
    <Tag Name="">Value</Tag>
   </MetaData>
   <Dir dirNID="" dirTitle="" />
   <Download>
            <Media format="Standard" state="NotReady" stale="" />
            <Media format="Proprietary" state="NotReady" stale="" />
         </Download>
  </File>
</Resource>
```

Example 2.58. Get File Response - BASICURL Format

```
<Resource from="200" total="400">
  <File NID='' title='' fileType='MediaClip' streamType='' size=''
   playlength='' state='' dateCreated='' dateModified=''
   readOnly='' mediaGroupId='' ffTracks='' mediaURL='' thumbnailURL='' bypassAut
   <Groups>
    <Tag Name='' Value='' />
   </Groups>
   <Permissions modify='' delete='' read='' export='' playback='' />
   <MetaData>
    <Tag Name=''>Value</Tag>
   </MetaData>
  </File>
</Resource>
```

Example 2.59. Get File Response - EXTENDEDURL Format

```
<Resource from="200" total="400">
  <File NID='' title='' fileType='' streamType='' size=''
  playlength='' presence='' state='' dateCreated='' dateModified=''
  readOnly='' mediaGroupId='' ffTracks='' mediaURL='' thumbnailURL='' bypassAuto
  <description />
  <Download>
            <Media format="Standard" state="NotReady" stale="" downloadURL=""/>
            <Media format="Proprietary" state="NotReady" stale="" downloadURL=""
         </Download>
  <Groups>
   <Tag Name='' Value='' />
  </Groups>
  <Permissions modify='' delete='' read='' export='' playback='' />
  <MetaData>
   <Tag Name=''>Value</Tag>
  </MetaData>
  <Dir dirNID='' dirTitle='' />
 </File>
</Resource>
```

Example 2.60. Get File Response - BASICCONNECTIONINFO Format

```
<Resource from="200" total="400">
  <File NID='' title='' fileType='' streamType='' size=''
  playlength='' presence='' state='' dateCreated='' dateModified=''
  readOnly='' mediaGroupId='' ffTracks='' mediaURL='' thumbnailURL='' bypassAuto
  <description />
  <Download>
            <Media format="Standard" state="NotReady" stale="" downloadURL=""/>
            <Media format="Proprietary" state="NotReady" stale="" downloadURL=""
         </Download>
  <Groups>
   <Tag Name='' Value='' />
  </Groups>
  <Permissions modify='' delete='' read='' export='' playback='' />
  <MetaData>
   <Tag Name=''>Value</Tag>
  </MetaData>
  <Dir dirNID='' dirTitle='' />
  <ConnectionStatus>
   <FileSessionInfo id='' name=''>
    <FileConnectionInfo id='' name='' state=''  type='VIEW/PLAYBACK/RECORD/PVR'
      srcNID='' srcTitle='' dstNID='' dstTitle='' errorCode='' errorData='' wind
   </FileSessionInfo>
  </ConnectionStatus>
 </File>
</Resource>
```

Get File Tags

This is the list of Tags that the Client has access to.

Example 2.61. Get File Tags Request

Table 2.18. Get File Tags Request

Request	HTTP GET
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.File-Tags+xml"
Status	200, 401, 500
Query Parameters	sortBy, sortOrder, whereNameTextSubString, whereValueTextSubString, wherePermissions, selectFormat
Resource Path	/Resource/File

Authorization	Basic (Required)

- *sortBy="Name/Value"*
- *whereNameTextSubString=""* [Optional]
- *whereValueTextSubString* [Optional]
- *wherePermissions="MODIFY,DELETE,PLAYBACK,EXPORT"* [Optional]
- *selectFormat="NameList/NameValueList/NameValueTree"* [Default=NameValueList]

Example 2.62. Get File Tags Response - NameList Format

```
   <Resource from="0" total="0">
    <FileTags>
  <Tag Name="Location"/>
  <Tag Name="Network"/>
 </FileTags>
   </Resource>
```

Example 2.63. Get File Tags Response - NameValueList Format

```
     <Resource from="0" total="0">
    <FileTags>
  <Tag Name="Location" Value="Sunnyvale"/>
  <Tag Name="Location" Value="Hyderabad"/>
 </FileTags>
</Resource>
```

Example 2.64. Get File Tags Response - NameValueTree Format

```
  <Resource from="0" total="0">
    <FileTags>
  <Tag Name="Location">
      <Value>Sunnyvale</Value>
      <Value>Hyderabad</Value>
  </Tag>
 </FileTags>
</Resource>
```

SetFileMetaData

replace file metadata in database

- *MetaData* [Required]

Example 2.65.

Table 2.19. Set File MetaData Request by NID or title

Request	HTTP POST
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.File-MetaData +xml"
Status	200, 401, 500
Query Parameters	NID or title
Resource Path	/Resource/File
Authorization	Basic (Required)

```
<FileMetaData>
 <MetaData>
  <Tag Name='ID'>12345</Tag>
  <Tag Name='TestCase'>UpdateWithByNID</Tag>
 </MetaData>
</FileMetaData>
```

Example 2.66. Set File MetaData Response

```
  <SetFileMetaData>
    <File NID='' title='' fileType='' streamType='' size=''
playlength='' presence='' state='' dateCreated='' dateModified=''
readOnly='' mediaGroupId='' bypassAutoStorageRules=''>
<description />
<Groups>
 <Tag name='' value='' />
</Groups>
<MetaData>
 <Tag Name=''>Value</Tag>
</MetaData>
<Dir dirNID='' dirTitle='' />
<Download>
          <Media format="" stale="" state=""/>
           </Download>
            <Permissions delete="" modify="" read="" export="" playback=""/>
 </File>
  </SetFileMetaData>
```

Delete File

Delete the given file from the system

- *NID* [Required. NID of the file you want to delete]

Example 2.67. DeleteFile Request

Table 2.20. DeleteFile Request

Request	HTTP DELETE
Accept(No spaces or newlines)	"application/json"
Status	200, 401, 500
Query Parameters	NID, moveToRecycleBin
Resource Path	/Resource/File
Authorization	Basic (Required)

- *`moveToRecycleBin='true/false'`* [if the value is true it will move the file to recycle bin. Otherwise it will delete from the system.Default is True]

Example 2.68. DeleteFile Response

```
 <Resource>
     <File NID="default.mediastreamfileresourcelist.msfrl_72ed6633-4072-45cb-b785
        <Dir NID="default.mediastreamdirresourcelist.msdrl_04e5116b-940c-4acc-aa3
        <Download>
           <Media format="Standard" stale="true" state="Prepared" />
        </Download>
        <Groups>
           <Tag Name="" Value="" />
        </Groups>
        <MetaData>
           <Tag>
              <element Name="ID">
                 <content>12345</content>
              </element>
              <element Name="TestCase">
                 <content>UpdateWithByNID</content>
              </element>
           </Tag>
        </MetaData>
        <Permissions delete="true" export="true" modify="true" playback="true" re
        <description />
     </File>
  </Resource>
```

Prepare File For download

Prepare file For download to client PC. Please subscribe for file updates using a subscriptionID and selectFormat as BASICURL/EXTENDEDURL in GetFile request, to get the file notification with a downloadURL, when media is Ready for download. Any edits to the media may cause the prepared status of the file to become stale, but still return a downloadURL on GetFile. It is the user's discretion whether to download a stale copy of the media or to issue another prepare to update the cached media for download.

- *`byNID`* [fileNID required]
- *`format="Proprietary/Standard/Both"`* [Optinal Default='Proprietary']

Example 2.69. Prepare File For Download Request

Table 2.21. Prepare File For Download Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	Resource/File/PrepareForDownload
Authorization	Basic (Required)

```
<PrepareFileForDownload  byNID=""
          format=""
         />
```

Example 2.70. Prepare File For Download Response

```
    <PrepareFileForDownload>
    <File NID="" title="" fileType="" streamType="" size="" playlength="" med
        <description/>
        <Dir NID="" dirTitle=""/>
        <MetaData>
            <Tag Name=""/>
        </MetaData>
        <Groups>
            <Tag Name="" Value=""/>
        </Groups>
        <Download>
            <Media format="Proprietary" stale="" state="Busy" downloadURL=""/
            <Media format="Standard" stale="" state="Busy" downloadURL=""/>
        </Download>
        <Permissions delete="" modify="" read="" export="" playback=""/>
    </File>
</PrepareFileForDownload>
```

Create Media From Cutlist

Creates Media from existing media using start and end offsets within it.Generates a file notification when the new media creation is complete. Please subscribe for file updates using a subscriptionID in GetFile request for receiving file notifications.

- *byNID* [fileNID of existing media - required]
- *cutList=""StartOffset:EndOffset,StartOffset:EndOffset"* [offsets of existing media in milliseconds. E.g. 0:30000 is 30 seconds of media from start offset. - optional. Skip the value of cutList if you wish to create a copy of parent file]

  Note: Cutlist values are validated against certain rules in the backend and will be rejected as invalid if not in proper order. Cutlist values should be integer values in an ascending order and they should not overlap. (for e.g. 0:30000,20000:40000 is an invalid cutlist range)
- *title* [Title for new media - required]
- *MetaData* [MetaData for new media - Optional]
- *Description* [Description for new media - required]

Example 2.71. Create Media From Cutlist

Table 2.22. Create Media From Cutlist

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	201, 401, 500
Query Parameters	None
Resource Path	Resource/File/CreateMediaFromCutList
Authorization	Basic (Required)

```
<CreateMediaFromCutList byNID="" cutList="StartOffset:EndOffset,StartOffset:E
    <MetaData>
           <Tag Name="">
             <content>Value</content>
          </Tag>
       </MetaData>
       <Description />
   </CreateMediaFromCutList>
```

Example 2.72. Create Media From Cutlist Response

```
     <CreateMediaFromCutList>
    <File NID="" title="" fileType="" streamType="" size="" playlength="" med
        <description/>
        <Dir NID="" dirTitle=""/>
        <MetaData>
            <Tag Name=""/>
        </MetaData>
        <Groups>
            <Tag Name="" Value=""/>
        </Groups>
        <Download>
            <Media format="Proprietary" stale="" state="Busy" downloadURL=""/
            <Media format="Standard" stale="" state="Busy" downloadURL=""/>
        </Download>
        <Permissions delete="" modify="" read="" export="" playback=""/>
    </File>
</CreateMediaFromCutList>
```

Trim File with Cutlist

Trim out sections of an existing mediafile using start and end offsets within it. Generates a file notification when the trimmed media is ready. Please subscribe for file updates using a subscriptionID in GetFile request for receiving file notifications.

- *`byNID`* [fileNID of existing media - required]
- *`cutList="StartOffset:EndOffset,StartOffset:EndOffset"`* [offsets of existing media in milliseconds. E.g. 0:30000 is 30 seconds of media from start offset. - required]

  Note: Cutlist values are validated against certain rules in the backend and will be rejected as invalid if not in proper order. Cutlist values should be integer values in an ascending order and they should not overlap. (for e.g. 0:30000,20000:40000 is an invalid cutlist range)

Example 2.73. Trim File with Cutlist

Table 2.23. Trim File with Cutlist

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	202, 401, 500
Query Parameters	None
Resource Path	Resource/File/Trim

Authorization	Basic (Required)

```
<TrimFile byNID=""
 cutList="StartOffset:EndOffset,StartOffset:EndOffset"></TrimFile>
```

Example 2.74. Trim File with Cutlist Response

```
<TrimFile>
 <File
  NID="default.mediastreamfileresourcelist.msfrl_a9504438-5bb1-4e09-a389-c60
  bypassAutoStorageRules="false" clipID="" dateCreated="1397505139401"
  dateModified="1397505139401" fileType="MediaClip" mediaGroupID=""
  mediaID="480193eb-7930-4d04-8d7a-0af670f2e5dd" playlength="0"
  readOnly="false" sessionConfigNID="" size="" state="Busy" streamType=""
  title="tx4">
  <Dir
   NID="default.mediastreamdirresourcelist.msdrl_4e09075d-cf15-44f8-98c3-ae8
   dirTitle="srcrelay1" />
  <Download>
   <Media format="" stale="" state="NotReady" />
  </Download>
  <Groups>
   <Tag>
    <element Name="Location" Value="Sunnyvale" />
    <element Name="Users" Value="Admin" />
   </Tag>
  </Groups>
  <MetaData>
   <Tag Name="">
    <content>Value</content>
   </Tag>
  </MetaData>
  <Permissions delete="true" export="true" modify="true"
   playback="true" read="true" />
  <description />
 </File>
</TrimFile>
```

Clear Download Cache

Free the disk space which is occupied by cached media file, after it is downloaded.

- *byNID* [fileNID of existing media - required]

Example 2.75. Clear Download Cache

Table 2.24. Clear Download Cache

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	202, 200, 401, 500
Query Parameters	None
Resource Path	Resource/File/ClearDownloadCache
Authorization	Basic (Required)

```
<ClearDownloadCache
      byNID="" >
                </ClearDownloadCache>
```

Example 2.76. Clear Download Cache Response

```
      <ClearDownloadCache>
    <File NID="" title="" fileType="" streamType="" size="" playlength="" med
        <description/>
        <Dir NID="" dirTitle=""/>
        <MetaData>
            <Tag Name=""/>
        </MetaData>
        <Groups>
            <Tag Name="" Value=""/>
        </Groups>
        <Download>
            <Media format="Proprietary" stale="" state="Busy" downloadURL=""/
            <Media format="Standard" stale="" state="Busy" downloadURL=""/>
        </Download>
        <Permissions delete="" modify="" read="" export="" playback=""/>
    </File>
</ClearDownloadCache>
```

Inject Media

Uploads the provided media file to the media store. To get the injectURL, you can use the Get DIR request with format as INJECTURL.

Example 2.77. Inject Media

```
Sample inject URL:

http://<DMS_IP>/mediajs/file/upload?dirID=/data1
         &timestamp=1389123260606
         &apiKey=-1110313941673189477
         &signature=fd154a45722592daa9c63ba4ccf8130e
         &sparam=bc8afb2b6575e6032ddc4eb90c8d0aa2
```

You can upload the file by appending the query parameters to the injectURL and sending HTTP POST with this URL. This should return 200:OK

```
Sample injectURL with appended parameters:

http://<DMS_IP>/mediajs/file/upload?dirID=/data1
         &timestamp=1389123260606
         &apiKey=-1110313941673189477
         &signature=fd154a45722592daa9c63ba4ccf8130e
         &sparam=bc8afb2b6575e6032ddc4eb90c8d0aa2
         &title=new_title
         &patientId=1234
         &patientName=new_patient
         &patientBirthDate=01011980
         &patientGender=male
         &accessionNumber=34534
         &doctorName=doctor_who
```

In case server configuration "Enable HLS Streaming" is true and you do not want to create hls then append the query parameter "options=donotcreatehls" to the injectURL.

```
Sample inject URL if you do not want to create hls:

http://<DMS_IP>/mediajs/file/upload?dirID=/data1
         &timestamp=1389123260606
         &apiKey=-1110313941673189477
         &signature=fd154a45722592daa9c63ba4ccf8130e
         &sparam=bc8afb2b6575e6032ddc4eb90c8d0aa2
         &options=donotcreatehls
```

Download Media

Download a media file from the media store on the system. To get the downloadURL, you can use the Get File request with format as EXTENDEDURL. Send an HTTP GET request using this downloadURL to download and save media to local machine. This should return 200:OK

Example 2.78. Download Media

```
Sample Download URL:
http://<DMS_IP>/mediajs/file/download?mediaID=316581ac-5168-42ae-adc7-e6c6609
        &dirID=/data1
        &timestamp=1389367743953
        &apiKey=5754498016574178712
        &signature=0895c6bd4404d70ab62f6f85104fa0f5
        &format=proprietary
        &uuid=tx1_89e9d445_53c6_4b7f_ae84_4c9b7f9e4905
        &persistence=temporary
```

Note: if downloadURL not present in the response for Get File, may suggest that the File is not prepared for download. Please use Prepare File Request to make the file available for download. This will result in a File Notification which includes the downloadURL for the file.Please subscribe for file updates using a subscriptionID in GetFile request in order to receive any file notifications.

User

API calls for User

Get User

This is the list of Users that the Client has access to.

Example 2.79. Get User Request

Table 2.25. Get User Request

Request	HTTP GET
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.User+xml"
Status	200, 401, 500
Query Parameters	startFrom, countToFetch, sortBy, sortOrder, whereNID, whereUsername, selectFormat, whereTextSubStr, whereTagExpr, subscriptionID

Resource Path	/Resource/User
Authorization	Basic (Required)

- *`sortBy="username/fullName"`* [Optional]
- *`subscriptionID`* [Optional]

Example 2.80. Get User Response - BASIC Format

```
<Resource from="0" total="50">
  <User NID="" username="" email="" fullName="" creationDate=""
     modifiedDate="" presence="" state="" externalAuthentication="">
   <Groups>
    <Tag Name="" Value="" />
   </Groups>
  </User>
  <User NID="" username="" email="" fullName="" creationDate=""
     modifiedDate="" presence="" state="" externalAuthentication="">
   <Groups>
    <Tag Name="" Value="" />
   </Groups>
  </User>
  <User NID="" username="" email="" fullName="" creationDate=""
     modifiedDate="" presence="" state="" externalAuthentication="">
   <Groups>
    <Tag Name="" Value="" />
   </Groups>
  </User>
</Resource>
```

Example 2.81. Get User Response - EXTENDED Format

```
<Resource from="0" total="50">
 <User NID="" username="" email="" fullName="" creationDate=""
    modifiedDate="" presence="" state="" externalAuthentication="">
  <Groups>
   <Tag Name="" Value="" />
  </Groups>
  <UserPrivateData/>
 </User>
</Resource>
```

Get User Tags

This is the list of Tags that the Client has access to.

Example 2.82. Get User Tags Request

Table 2.26. Get User Tags Request

Request	HTTP GET
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.User-Tags+xml"
Status	200, 401, 500
Query Parameters	sortBy, sortOrder, whereNameTextSubString, WhereValueTextSubStr, selectFormat
Resource Path	/Resource/User
Authorization	Basic (Required)

- *sortBy="Name/Value"*
- *whereNameTextSubString=""* [Optional]
- *whereValueTextSubString* [Optional]
- *selectFormat="NameList/NameValueList/NameValueTree"* [Default=NameValueList]

Example 2.83. Get User Tags Response - NameList Format

```
 <Resource from="0" total="0">
  <UserTags>
<Tag Name="Location"/>
<Tag Name="Network"/>
  </UserTags>
 </Resource>
```

Example 2.84. Get User Tags Response - NameValueList Format

```
 <Resource from="0" total="0">
   <UserTags>
 <Tag Name="Location" Value="Sunnyvale"/>
 <Tag Name="Location" Value="Hyderabad"/>
</UserTags>
 </Resource>
```

Example 2.85. Get User Tags Response - NameValueTree Format

```
  <Resource from="0" total="0">
    <UserTags>
  <Tag Name="Location">
      <Value>Sunnyvale</Value>
      <Value>Hyderabad</Value>
  </Tag>
 </UserTags>
</Resource>
```

Set User Password

This is used to set the user password.

Example 2.86. Set User Password Request

Table 2.27. Set User Password Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.User-Password +xml"
Status	200, 401, 500
Query Parameters	NID
Resource Path	/Resource/User
Authorization	Basic (Required)

```
<SetUserPassword oldPassword="" newPassword=""/>
```

Example 2.87. Set User Password Response

```
<Resource>
 <User NID="" username="" externalAuthentication="" email="" fullName="" creation
        <Groups>
            <Tag Name="" Value=""/>
        </Groups>
        <UserPrivateData/>
    </User>
   </Resource>
```

Get Tags

This is the list of Tags that the Client has access to.

Example 2.88. Get Tags Request

Table 2.28. Get Tags Request

Request	HTTP GET
Accept(No spaces or newlines)	"application/vnd.barco.nimbus.Tags+xml"
Status	200, 401, 500
Query Parameters	sortBy, sortOrder, whereNameTextSubString, whereValueTextSubString, wherePermissions, whereObjectType, selectFormat
Resource Path	/Resource/Tags
Authorization	Basic (Required)

- *sortBy="Name/Value"*
- *whereNameTextSubString=""* [Optional]
- *whereValueTextSubString* [Optional]
- *selectFormat="NameList/NameValueList/NameValueTree"* [Default=NameValueList]
- *whereObjectType="Device/Users/MediaRoom/Profiles/Ports/Media/ALL"*

Example 2.89. Get Tags Response - NameList Format

```
 <Resource from="0" total="0">
  <Tags>
<Tag Name="Location"
objectType="Devices/Users/MediaRoom/Profiles/Ports/Media/ALL"/>
<Tag Name="Network"
objectType="Devices/Users/MediaRoom/Profiles/Ports/Media/ALL"/>
  </Tags>
 </Resource>
```

Example 2.90. Get Tags Response - NameValueList Format

```
   <Resource from="0" total="0">
    <Tags>
  <Tag Name="Location" Value="Sunnyvale"
   objectType="Devices/Users/MediaRoom/Profiles/Ports/Media/ALL"/>
  <Tag Name="Location" Value="Hyderabad"
  objectType="Devices/Users/MediaRoom/Profiles/Ports/Media/ALL"/>
 </Tags>
</Resource>
```

Example 2.91. Get Tags Response - NameValueTree Format

```
  <Resource from="0" total="0">
    <Tags>
  <Tag Name="Location"
     objectType="Devices/Users/MediaRoom/Profiles/Ports/Media/ALL">
      <Value>Sunnyvale</Value>
      <Value>Hyderabad</Value>
  </Tag>
 </Tags>
</Resource>
```

Get DisplayProfile

This is the list of DisplayProfile objects that the Client has access to.

Example 2.92. Get DisplayProfile Request

Table 2.29. Get DisplayProfile Request

Request	HTTP GET
Accept	"application/vnd.barco.nimbus.DisplayProfile +xml"
Status	200, 401, 500
Query Parameters	selectFormat, sortBy, sortOrder, countToFetch, startFrom, whereDateRange, whereNID,whereTagExpr,whereTextSubStr
Resource Path	Resource/DisplayProfile

Authorization	Basic (Required)

- *sortBy="title"*
- *whereTextSubStr=""* [Optional]
- *whereTagExpr* [Optional]
- *selectFormat="BASIC/EXTENDED"* [Default=BASIC]

Example 2.93. Get DisplayProfile Response - BASIC Format

```
          <Resource from="1" total="12">
    <DisplayProfile NID="default.displayprofilelist.dpl_1d31d3f3-f0ea-42a1-8f6a-ca
</Resource>
```

Example 2.94. Get DisplayProfile Response - EXTENDED Format

```
    <Resource from="1" total="12">
    <DisplayProfile NID="default.displayprofilelist.dpl_1d31d3f3-f0ea-42a1-8f6a-ca
        <description/>
        <Groups>
            <Tag Name="" Value=""/>
        </Groups>
        <Canvas id="" hres="1920" vres="1080">
            <Background enable="" zIndex="" content="">
                <ImageContent imageURL="" color=""/>
                <SessionContent enableConnPanel="" enableBookmarkPanel="" enableCt
            </Background>
            <Layout id="LayoutId1" type="" nRow="" nCol="" colWidth="" rowHeight="
            <Window id="Win1" content="">
                <ConnectionContent MediaStreamIOPortTypeTitle="" MediaStreamIOPort
                <ThumbnailContent MediaStreamIOPortTypeTitle="" MediaStreamIOPortE
                <Properties zIndex="100" border="" vpos="0%" hpos="0%" vres="100%"
            </Window>
            <Window id="Win2" content="">
                <ConnectionContent MediaStreamIOPortTypeTitle="" MediaStreamIOPort
                <ThumbnailContent MediaStreamIOPortTypeTitle="" MediaStreamIOPortE
                <Properties zIndex="100" border="" vpos="0%" hpos="50%" vres="100%
            </Window>
        </Canvas>
    </DisplayProfile>
</Resource>
```

Chapter 3. Resource Update Notifications

Resource ConnectionUpdate Notification

- *`sessionId`* [session id]
- *`id`* [connection id]
- *`name`* [connection name]
- *`state`* [connection state: Check connectionNotification State listing for enum values]
- *`errorCode`* [optional if there is no error.]
- *`errorData`* [error description. optional if there is no error.]
- *`windowId`* [window id]
- *`canvasId`* [canvas Id]
- *`SrcEntity:id`* [source entity id]
- *`SrcEntity:resourceTitle`* [source title]
- *`SrcEntity:resourceNID`* [source nid]
- *`SrcEntity:userJID`* [source user JID]
- *`SrcEntity:userFullName`* [source user full name]
- *`DstEntity:id`* [destination entity id]
- *`DstEntity:resourceTitle`* [destination title]
- *`DstEntity:resourceNID`* [destination nid]
- *`DstEntity:userJID`* [destination user JID]
- *`DstEntity:userFullName`* [destination user full name]
- *`Profile:id`* [id of profile]
- *`Profile:profileNID`* [Pofile nid]
- *`Profile:profileName`* [Pofile Name]
- *`Profile:allowKMSwitch='true/false'`* [allow key boeard mouse switch]
- *`Profile:remoteKM='true/false'`* [describes remote key boeard mouse allowed or not]
- *`Profile:avOption='Audio/Video/Both'`* [information about audio video options]
- *`Profile:maxBandwidth`* [bandwidth used for this connection]
- *`Profile:lowCompression`* [low compression used for this connection]
- *`Profile:highCompression`* [high compression used for this connection]

- *`Profile:burstRatio`* [burst Ratio used for this connection]
- *`Profile:burstDuration`* [burst duration used for this connection]
- *`Profile:unicast='true/false'`* [information about unicast connection]
- *`Profile:multicast='true/false'`* [information about multicast connection]

Example 3.1. AuthConectionInfo element in AuthorizationRequired Notification

```
   <AuthConnectionInfo sessionId='' id=''  name='' state='' type='VIEW/PLAYBACK/
 <SrcEntity id='' resourceTitle='' resourceNID='' userJID='' userFullName='' />
 <DstEnity id='' resourceTitle='' resourceNID='' userJID=''  userFullName='' />
 <Profile id='' profileName='' profileNID='' allowKMSwitch='' remoteKM='' avOpti
       lowCompression='' highCompression='' burstRatio='' burstDuration='' unica
</AuthConnectionInfo>
```

Example 3.2. PortConectionInfo element in PortConnectionUpdate Notification - BASICCONNECTIONINFO format

```
         <ConnectionStatus>
 <PortSessionInfo id='' name=''>
     <PortConnectionInfo id='' name='' state=''  type='VIEW/PLAYBACK/RECOR
    srcNID='' srcTitle='' dstNID='' dstTitle='' errorCode='' errorData=''
  </PortSessionInfo>
</ConnectionStatus>
```

Example 3.3. PortConectionInfo element in PortConnectionUpdate Notification - EXTENDEDCONNECTIONINFO format

```
         <ConnectionStatus>
 <PortSessionInfo id='' name=''>
     <PortConnectionInfo id='' name='' state=''  type='VIEW/PLAYBACK/RECOR
    srcNID='' srcTitle='' dstNID='' dstTitle='' errorCode='' errorData=''
    srcUserJID='' srcUserName='' destUserJID='' destUserName=''>

    <Profile id='' profileName='' profileNID='' allowKMSwitch='' remoteKM=
       maxBandwidth='' lowCompression='' highCompression='' burstRatio=''
    burstDuration='' unicast='' multicast='' />
  </PortConnectionInfo>
 </PortSessionInfo>
</ConnectionStatus>
```

Example 3.4. FileConnectionInfo element in FileConnectionUpdate Notification - BASICCONNECTIONINFO format

```
        <ConnectionStatus>
 <FileSessionInfo id='' name=''>
  <FileConnectionInfo id='' name='' state=''  type='VIEW/PLAYBACK/RECORD/P
    srcNID='' srcTitle='' dstNID='' dstTitle='' errorCode='' errorData=''
 </FileSessionInfo>
</ConnectionStatus>
```

Resource ConnectionStateUpdate Notification

Check ConnState and Timeline notifications in SessionNotifications for paramter listing

Example 3.5. ConectionState element in ConnectionStateUpdate Notification

```
  <ConnectionState sessionId='' connId='' value='' errorCode='' errorData=''>
 <Timeline sessionId="" connGroupId="" connId="" id="">
  <Start WC="2341234" TC="" Offset="3456" parentTimelineId=""/>
  <Last WC="7444674" TC="" Offset="9678687" TimeScale="0"
  requestedAction="Pause"/>
 </Timeline>
</ConnectionState>
```

Port

This is a list of update notifications corresponding to the port.

Port Update

This is a notification for port update.

Example 3.6. Port Update Notification BASIC

```
 <PortUpdate>
   <Port NID='' title='' streamType='' presence='' state='' portType='' select
</PortUpdate>
```

Example 3.7. Port Update Notification EXTENDED

```
 <PortUpdate>
   <Port NID='' title='' streamType='' presence='' state='' portType='' select
 <Asset NID='' title='' deviceType='' deviceNID='' />
 <description />
 <Groups>
  <Tag name='' value='' />
 </Groups>
 <MaxInstanceCount />
 <FeatureList>
  <Feature name="" />
 </FeatureList>
</Port>
</PortUpdate>
```

Port Connection Update

This is a notification for port connection update.

Example 3.8. Port Connection Update Notification

```
           <PortConnectionUpdate>
              <Port NID='' subscriptionID='' operation='Add/Delete'>
     <PortConnectionInfo ...../>
     <!-- Check PortConnectionInfo element in ConnectionUpdate Section
  </Port>
</PortConnectionUpdate>
```

Port Connection State Update

This is a notification for port connection state update.

Example 3.9. Port Connection State Update Notification

```
<PortConnectionStateUpdate>
 <Port NID='' subscriptionID=''>
     <ConnectionState .../>
  <!-- check ConnectionState element in ConnectionStateUpdate section in Update
 </Port>
</PortConnectionStateUpdate>
```

Port Display Config Update

This is a notification for port Display Config Updates.

Example 3.10. Port DisplayConfigUpdate Notification

```
<PortDisplayConfigUpdate>
 <Port NID='' subscriptionID='' operation='Update'>
   <!--Check the appendix for device specific config-->
 </Port>
   </PortDisplayConfigUpdate>
```

Port Display Status Update

This is a notification for port Display Status Updates.

Example 3.11. Port DisplayStatusUpdate Notification

```
<PortDisplayStatusUpdate>
 <Port NID='' subscriptionID='' operation='Update'>
   <!--Check the appendix for device specific Status-->
 </Port>
   </PortDisplayStatusUpdate>
```

File

This is a list of update notifications corresponding to the File.

File Update

This is a notification for File update.

Example 3.12. File Update Notification BASIC

```
         </FileUpdate>
           <File NID='' title='' fileType='MediaClip' streamType='' size=''
 playlength='' state='' dateCreated='' dateModified=''
 readOnly='' mediaGroupId='' ffTracks='' bypassAutoStorageRules='true/false'>
 <Groups>
  <Tag Name='' Value='' />
 </Groups>
 <Permissions modify='' delete='' read='' export='' playback='' />
 <MetaData>
  <Tag Name=''>Value</Tag>
 </MetaData>
 </File>
</FileUpdate>
```

Example 3.13. File Update Notification EXTENDED

```
          </FileUpdate>
           <File NID='' title='' fileType='MediaClip' streamType='' size=''
 playlength='' state='' dateCreated='' dateModified=''
 readOnly='' mediaGroupId='' ffTracks='' bypassAutoStorageRules='true/false'>
 <Groups>
  <Tag Name='' Value='' />
 </Groups>
 <Permissions modify='' delete='' read='' export='' playback='' />
 <MetaData>
  <Tag Name=''>Value</Tag>
 </MetaData>
 <Download>
           <Media format="Standard" state="NotReady" stale="" />
           <Media format="Proprietary" state="NotReady" stale=""/>
        </Download>
        <Dir dirNID='' dirTitle='' />
 </File>
</FileUpdate>
```

Example 3.14. File Update Notification EXTENDEDURL

```
 </FileUpdate>
           <File NID='' title='' fileType='MediaClip' streamType='' size=''
 playlength='' state='' dateCreated='' dateModified=''
 readOnly='' mediaGroupId='' ffTracks='' bypassAutoStorageRules='true/false'>
 <Groups>
  <Tag Name='' Value='' />
 </Groups>
 <Permissions modify='' delete='' read='' export='' playback='' />
 <MetaData>
  <Tag Name=''>Value</Tag>
 </MetaData>
 <Download>
           <Media format="Standard" state="NotReady" stale="" />
           <Media format="Proprietary" state="NotReady" stale=""/>
        </Download>
        <Dir dirNID='' dirTitle='' />
 </File>
</FileUpdate>
```

File Connection Update

This is a notification for file connection update.

Example 3.15. File Connection Update Notification

```
           <FileConnectionUpdate>
              <File NID='' subscriptionID='' operation='Add/Delete'>
     <FileSessionInfo ...../>
     <!-- Check FileConnectionInfo element in ConnectionUpdate Section
  </File>
</FileConnectionUpdate>
```

File Connection State Update

This is a notification for file connection state update.

Example 3.16. File Connection State Update Notification

```
<FileConnectionStateUpdate>
 <File NID='' subscriptionID=''>
     <ConnectionState .../>
  <!-- check ConnectionState element in ConnectionStateUpdate section in Update
 </File>
</FileConnectionStateUpdate>
```

Chapter 4. Session API

API Calls used in Session

Session Info

API calls for Session Info

Get Session Config

This is the list of runtime Session Configurations that the Client has access to. Refer to the Appendix for the format of the Session object

Example 4.1. Get Session Config Request

Table 4.1. Get Session Config Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

- *`startFrom`*
- *`countToFetch`*
- *`sortBy`*
- *`sortOrder`*
- *`whereNID`*
- *`whereTagExpr`*
- *`whereTextSubStr`*
- *`subscriptionID`*

Example 4.2. Get Session Config Request

```
<GetSessionConfig startFrom="1" countToFetch="100"
 sortBy="" sortOrder="ascending/descending"
 selectFormat="BASIC/EXTENDED"
 whereNID=""
 whereTextSubStr=""
 whereTagExpr=""
 subscriptionID=''/>
```

Example 4.3. Get Session Config Response - BASIC Format

```
<GetSessionConfig from="0" total="50">
  <SessionConfig NID="" title="" dateCreated="" playbackOption="" type="">
   <Groups>
    <Tag Name="" Value="" />
   </Groups>
  </SessionConfig>
  <SessionConfig NID="" title="" dateCreated="" playbackOption="" type="">
   <Groups>
    <Tag Name="" Value="" />
   </Groups>
  </SessionConfig>
</GetSessionConfig>
```

Example 4.4. Get Session Config Response - EXTENDED Format

```
<GetSessionConfig from="0" total="50">
   <SessionConfig NID="" title="">
   <description />
   <Groups>
    <Tag Name="" Value="" />
   </Groups>
   <SessionInfo/>
   </SessionConfig>
</GetSessionConfig>
```

Get Session

[Not Implemented]

This is the list of runtime Sessions that the Client has access to. Refer to the Appendix for the format of the Session object

Example 4.5. Get Session Request

Table 4.2. Get Session Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session

Authorization	Basic (Required)

- *`startFrom`* [Not Supported]
- *`countToFetch`* [Not Supported]
- *`sortBy`* [Not Supported]
- *`sortOrder`* [Not Supported]
- *`whereNID`* [Not Supported]
- *`whereTagExpr`* [Not Supported]
- *`whereTextSubStr`* [Not Supported]

Example 4.6. Get Session Request

```
<GetSession startFrom="1" countToFetch="100"
 sortBy="" sortOrder="ascending/descending"
 selectFormat="BASIC/EXTENDED"
 whereNID=""
 whereTextSubStr=""
 whereTagExpr=""
 subscriptionID=''/>
```

Example 4.7. Get Session Response - BASIC Format[Not Implemented]

```
<GetSession from="0" total="50">
  <Session id='' name='' type='' state=''
   timelineId='' recordTimelineId=''>
   <Properties sessionConfigNID='' ownerUserJID=''
    dateCreated='' playlength='' />
  </Session>
  <Session id='' name='' type='' state=''
   timelineId='' recordTimelineId=''>
   <Properties sessionConfigNID='' ownerUserJID=''
    dateCreated='' playlength='' />
  </Session>
  <Session id='' name='' type='' state=''
   timelineId='' recordTimelineId=''>
   <Properties sessionConfigNID='' ownerUserJID=''
    dateCreated='' playlength='' />
  </Session>
</GetSession>
```

Example 4.8. Get Session Response - BASIC Format[Currently Implemented]

```
<GetSession from="0" total="50">
  <Session id='' name='' type='' state=''
   timelineId='' recordTimelineId=''>
   <Properties sessionConfigNID='' ownerUserJID=''
    dateCreated=''/>
  </Session>
  <Session id='' name='' type='' state=''
   timelineId='' recordTimelineId=''>
   <Properties sessionConfigNID='' ownerUserJID=''
    dateCreated=''/>
  </Session>
  <Session id='' name='' type='' state=''
   timelineId='' recordTimelineId=''>
   <Properties sessionConfigNID='' ownerUserJID=''
    dateCreated=''/>
  </Session>
</GetSession>
```

Example 4.9. Get Session Response - EXTENDED Format[Not Implemented]

```
<GetSession from="0" total="50">
  <Session id='' name='' type='' state=''
   timelineId='' recordTimelineId=''>
   <Properties sessionConfigNID='' ownerUserJID=''
    dateCreated='' playlength='' />
   <Groups>
    <Tag Name="" Value="" />
   </Groups>
   <MetaData>
    <Tag Name=''></Tag>
   </MetaData>
   <TimelineList>
    <Timeline id=''>
     <Start WC='' Offset='' TC='' parentTimelineId='' />
     <Last WC='' Offset='' TC='' TimeScale='' requestedAction=''/>
    </Timeline>
   </TimelineList>
   <GrpFileList>
    <GrpFile type='SyncPly' id='' resourceNID='' resourceTitle=''
     dateCreated='' recSessionConfigId='' recSessionId='' syncEntityId=''>
            <MetaData>
               <Tag Name=""></Tag>
            </MetaData>
         </GrpFile>
    <GrpFile type='SyncRec' id='' resourceNID='' resourceTitle='' >
            <MetaData>
               <Tag Name=""></Tag>
            </MetaData>
       </GrpFile>
   </GrpFileList>
   <DirList>
    <Dir id='' resourceNID='' resourceTitle='' />
   </DirList>
   <ProfileList>
    <Profile id='' name='' profileNID='' ffTracks=''/>
   </ProfileList>
```

Example 4.10. Get Session Response - EXTENDED Format[Not Implemented] (Continued)

```
<EntityList>
 <Entity type='SrcPort' id='' name='' streamType='' presence='' gc='' state='
  <User type='' userNID='' username='' userJID='' />
  <Port type='' resourceNID='' resourceTitle='' trickPlay=''/>
  <PVR enable='' trimLength='' connId='' recEntityId=''
   plyEntityId='' ffTracks='' />
 </Entity>
 <Entity type='DstPort' id='' name='' streamType='' presence='' gc='' state='
  <User type='' userNID='' username='' userJID='' />
  <Port type='' resourceNID='' resourceTitle='' trickPlay=''/>
 </Entity>
 <Entity type='RlyPort' id='' name='' streamType='' presence='' gc='' state='
  <User type='' userNID='' username='' userJID='' />
  <Port type='' resourceNID='' resourceTitle='' trickPlay=''/>
 </Entity>
 <Entity type='PlyFile' id='' name='' streamType='' presence='' gc='' state='
  <User type='' userNID='' username='' userJID='' />
  <File type='' resourceNID='' resourceTitle='' dirId=''
   grpFileId='' startOffset='' playLength='' />
  <Port type='' resourceNID='' resourceTitle='' trickPlay=''/>
 </Entity>
 <Entity type='RecFile' id='' name='' streamType='' presence='' gc='' state='
  <User type='' userNID='' username='' userJID='' />
  <File type='' resourceNID='' resourceTitle='' dirId=''
   grpFileId='' key="" action="" keyMetaDataKeyName="">
            <MetaData>
               <Tag Name=""></Tag>
            </MetaData>
        </File>
  <Port type='' resourceNID='' resourceTitle='' trickPlay=''/>
 </Entity>
 <Entity type='PVRFile' id='' name='' streamType='' presence='' gc='' state='
  <User type='' userNID='' username='' userJID='' />
  <File type='' resourceNID='' resourceTitle='' dirId=''
   trimLength='' />
  <Port type='' resourceNID='' resourceTitle='' trickPlay=''/>
 </Entity>
</EntityList>
```

Example 4.11. Get Session Response - EXTENDED Format[Not Implemented] (Continued)

```
   <ConnectionList>
    <Connection type='VIEW/PLAYBACK' id='' gid='' name=''
     mode='AUTO/ONDEMAND' srcEntityId='' dstEntityId='' profileId=''
     authEntityId='' streamURL='' state='' timelineId='' />
    <Connection type='RECORD' id='' gid='' name=''
     mode='AUTO/ONDEMAND' srcEntityId='' dstEntityId='' profileId=''
     authEntityId='' streamURL='' state='' timelineId='' />
    <Connection type='PVR' id='' gid='' name='' mode='AUTO/ONDEMAND'
     srcEntityId='' dstEntityId='' profileId='' authEntityId='' streamURL=''
     state='' timelineId='' />
   </ConnectionList>
  </Session>
</GetSession>
```

Example 4.12. Get Session Response - EXTENDED Format[Currently Implemented]

```
<GetSession from="0" total="50">
  <Session id='' name='' type='' state=''
   timelineId='' recordTimelineId=''>
   <Properties sessionConfigNID='' ownerUserJID=''
    dateCreated=''/>
   <Groups>
    <Tag Name="" Value="" />
   </Groups>
   <MetaData>
    <Tag Name=''></Tag>
   </MetaData>
   <TimelineList>
    <Timeline id=''>
     <Start WC='' Offset='' TC='' parentTimelineId='' />
     <Last WC='' Offset='' TC='' TimeScale='' requestedAction=''/>
    </Timeline>
   </TimelineList>
   <GrpFileList>
    <GrpFile type='SyncPly' id='' resourceNID='' resourceTitle=''
     dateCreated='' recSessionConfigId='' recSessionId='' syncEntityId='' >
            <MetaData>
               <Tag Name=""></Tag>
            </MetaData>
       </GrpFile>
    <GrpFile type='SyncRec' id='' resourceNID='' resourceTitle='' >
            <MetaData>
               <Tag Name=""></Tag>
            </MetaData>
       </GrpFile>
   </GrpFileList>
```

Example 4.13. Get Session Response - EXTENDED Format[Currently Implemented] (Continued)

```
   <EntityList>
    <Entity type='SrcPort' id='' name='' streamType='' presence='' gc='' state='
     <Port type='' resourceNID='' resourceTitle='' trickPlay=''/>
     <PVR enable='' trimLength='' connId='' ffTracks=''/>
    </Entity>
    <Entity type='DstPort' id='' name='' streamType='' presence='' gc='' state='
     <Port type='' resourceNID='' resourceTitle='' trickPlay=''/>
    </Entity>
    <Entity type='PlyFile' id='' name='' streamType='' presence='' gc='' state='
     <File type='' resourceNID='' resourceTitle='' dirId=''
      grpFileId='' startOffset='' playLength=''/>
    </Entity>
    <Entity type='RecFile' id='' name='' streamType='' presence='' gc='' state='
     <File type='' resourceNID='' resourceTitle='' dirId=''
      grpFileId='' key="" action="" keyMetaDataKeyName="">
               <MetaData>
                  <Tag Name=""></Tag>
               </MetaData>
           </File>
    </Entity>
   </EntityList>
```

Example 4.14. Get Session Response - EXTENDED Format[Currently Implemented] (Continued)

```
   <ConnectionList>
    <Connection id='' srcEntityId='' dstEntityId='' streamURL=''
                state='' timelineId='' errorCode='' errorData=''>
     <ConnSeg srcResourceTitle='' dstResourceTitle='' profileTitle=''
              state='' errorCode='' errorData='' srcResourceNID=''
              dstResourceNID='' profileNID=''/>
    </Connection>
   </ConnectionList>
  </Session>
</GetSession>
```

Session Requests

Parameter Listing

This is the list of Parameters in Session Setup Requests

1. Common Parameters
   - *`sessionId`* [Required]
   - *`sessionConfigId`* [Optional]
2. GrpFile Parameters [Either resourceNID or resourceTitle Required]
   - *`type="SyncRec/SyncPly"`* [Required]
   - *`id`* [Required]
   - *`resourceNID`* [Required]
   - *`resourceTitle`* [Required]
3. Entity Parameters
   - *`type="SrcPort/DstPort/PlyFile/RecFile/PVRFile"`* [Required]
   - *`id`* [Required]
   - Port Parameters [Either resourceNID or resourceTitle Required]
     - *`resourceNID`* [Required]
     - *`resourceTitle`* [Required]
   - PVR Parameters [TBD]
   - File Parameters [Either resourceNID or resourceTitle Required]
     - *`resourceNID`* [Required]
     - *`resourceTitle`* [Required]
     - *`grpFileId`* [Optional]
     - *`key="Title/MetaData"`* [Optional]
     - *`action="Append/Overwrite"`* [Optional]
     - *`keyMetaDataKeyName`* [Optional]
     - *`MetaData element`* [Optional]
4. Connection Parameters
   - *`id`* [Required]
   - *`srcEntityId`* [Required]
   - *`dstEntityId`* [Required]

StartSession

- Note on Groups:

The Groups tag is used to set the Group of the recordings created as a result of this session

- Note on GrpFile:

  GrpFile is referenced by Files in the SessionRequest to indicate their corresponding media group file information
  GrpFile.resourceTitle with (type=SyncRec) is used to set the title of the GroupFile for the recordings created as a result of this session

- Note on EntityList:

  This is the list of sources,destinations in the session
  Entity where type=SrcPort is a live source referened by its title or NID
  Entity.PVR.enable where type=SrcPort indicates that this should have PVR
  Entity where type=DstPort is a destination that can receive from a live or playback source refernced by its title or NID
  Entity where type=RecFile is a recording file to be created with the specified title
  Entity.File.grpFileId indicates that this recording will be group under the GrpFile element with the same id
  Entity where type=PlyFile is a recording file to be played back referenced by the specified NID

- Note on ConnectionList:

  This is the list of connections in the session
  Connection.srcEntityId references the source entity with the matching id
  Connection.dstEntityId references the destination entity with the matching id
  Connection.id will set the id for the connection between the source and destination
  Connection.type will set the type of connection. Possible Connections are 'VIEW/PLAYBACK/RECORD/PVR'

Example 4.15. Basic Start Recording Session Request

Table 4.3. Basic Start Recording Session Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session

Authorization	Basic (Required)

```
<StartSession sessionId="recordSessionID">
 <Session id="recordSessionConfigNID">
  <Properties sessionConfigNID="" ownerUserJID=""
    dateCreated="" playlength=""/>
    <Groups>
        <Tag Name="Location" Value="Sunnyvale"/>
    </Groups>
    <MetaData>
        <Tag Name=""/>
    </MetaData>
    <GrpFileList>
        <GrpFile type="SyncRec" id="gfId" resourceNID=""
         resourceTitle="GroupFileTitle">
        <MetaData>
           <Tag Name=""></Tag>
        </MetaData>
     </GrpFile>
    </GrpFileList>
    <DirList>
       <Dir id="dirId" resourceNID= "" resourceTitle=""/>
    </DirList>
    Dir List is Optional element.
    <ProfileList>
        <Profile id="profileid"
     profileNID="" name="" ffTracks=""/>
    </ProfileList>
     Profile List is optional element
```

Example 4.16. Basic Start Recording Session Request (Continued)

```
<EntityList>
  <Entity type="SrcPort" id="src1">
      <Port resourceNID="" resourceTitle="tx1" trickPlay=""/>
      <PVR enable="true" trimLength="" startTC="" SessionId=""
       recEntityId="" plyEntityId="" connID="" ffTracks=""/>
  </Entity>
  <Entity type="DstPort" id="dst1">
      <Port resourceNID="" resourceTitle="playerpc" trickPlay="false"/>
  </Entity>
  <Entity type="RecFile" id="recordd1">
      <File resourceNID="" resourceTitle="title1" grpFileId="gfId"
      key="" action="" keyMetaDataKeyName="">
       <MetaData>
          <Tag Name=""></Tag>
       </MetaData>
   </File>
  </Entity>
  <Entity type="SrcPort" id="src2">
      <Port resourceNID="" resourceTitle="rtp_tx1" trickPlay=""/>
      <PVR enable="true" trimLength="" startTC="" SessionId=""
       recEntityId="" plyEntityId="" connID="" ffTracks=""/>
  </Entity>
  <Entity type="DstPort" id="dst2">
      <Port resourceNID="" resourceTitle="playerpc" trickPlay="true"/>
  </Entity>
  <Entity type="RecFile" id="recordd2">
      <File resourceNID="" resourceTitle="title2" grpFileId="gfId"
      dirId=" key="" action="" keyMetaDataKeyName="">
       <MetaData>
          <Tag Name=""></Tag>
       </MetaData>
   </File>
  </Entity>
</EntityList>
```

Example 4.17. Basic Start Recording Session Request (Continued)

```
    <ConnectionList>
        <Connection id="strmc1" srcEntityId="src1" dstEntityId="dst1" profileId
           <Connection id="recordc1" srcEntityId="src1" dstEntityId="recordd1"
           <Connection id="strmc2" srcEntityId="src2" dstEntityId="dst2" canvas
           <Connection id="recordc2" srcEntityId="src2" dstEntityId="recordd2"
    </ConnectionList>
   </Session>
</StartSession>
```

Example 4.18. Advance Start Recording Session Request [Not Implemented]

```
<StartSession sessionId="recordSessionID" >
   <Session id="recordSessionConfigNID"
     name="" state="" timelineId="">
    <Properties sessionConfigNID="" ownerUserJID=""
                dateCreated="" playlength=""/>
    <Groups>
        <Tag Name="Location" Value="Sunnyvale"/>
    </Groups>
    <MetaData>
        <Tag Name=""/>
    </MetaData>
    <TimelineList>
       <Timeline id="">
           <Start WC="" Offset="" TC="" parentTimelineId=""/>
           <Last WC="" Offset="" TC="" TimeScale="" requestedAction=""/>
       </Timeline>
    </TimelineList>
    <GrpFileList>
      <GrpFile type="SyncRec" id="gfId" resourceNID=""
                 resourceTitle="GroupFileTitle">
        <MetaData>
           <Tag Name=""></Tag>
        </MetaData>
     </GrpFile>
    </GrpFileList>
    <DirList/>
    <ProfileList>
    </ProfileList>
    <EntityList>
      <Entity type="SrcPort" id="src1" name="" streamType="" presence="" gc=""
          <User type="" userNID="" username="" userJID=""/>
          <Port type="" resourceNID="" resourceTitle="tx1" trickPlay=""/>
          <PVR enable="" trimLength="" startTC="" sessionId="" connId="" recEnt
      </Entity>
```

Example 4.19. Advance Start Recording Session Request [Not Implemented] (Continued)

```
            <Entity type="DstPort" id="dst1" name="" streamType="" presence="" gc=""
                <User type="" userNID='' username='' userJID=""/>
                <Port type="" resourceNID="" resourceTitle="playerpc" trickPlay='fal
            </Entity>
            <Entity type="RecFile" id="recordd1" name="" streamType="" presence="" g
                <User type="" userNID='' username='' userJID=""/>
                <File type="" resourceNID="" resourceTitle="title1" dirId=""
                grpFileId="gfId"/>
                <Port type="" resourceNID="" resourceTitle="" trickPlay=""/>
            </Entity>
            <Entity type="SrcPort" id="src2" name="" streamType="" presence="" gc=""
                <User type="" userNID='' username='' userJID=""/>
                <Port type="" resourceNID="" resourceTitle="rtp_tx1" trickPlay=""/>
                <PVR enable="" trimLength="" startTC="" sessionId="" connId="" recEn
            </Entity>
            <Entity type="DstPort" id="dst2" name="" streamType="" presence="" gc=""
                <User type="" userNID='' username='' userJID=""/>
                <Port type="" resourceNID="" resourceTitle="playerpc" trickPlay="tru
            </Entity>
            <Entity type="RecFile" id="recordd2" name="" streamType="" presence="" g
                <User type="" userNID='' username='' userJID=""/>
                <File type="" resourceNID="" resourceTitle="title2" dirId=""
                grpFileId="gfId" key="" action="" keyMetaDataKeyName="">
                 <MetaData>
                    <Tag Name=""></Tag>
                 </MetaData>
            </File>
                <Port type="" resourceNID="" resourceTitle="" trickPlay=""/>
            </Entity>
        </EntityList>
```

Example 4.20. Advance Start Recording Session Request [Not Implemented] (Continued)

```
   <ConnectionList>
      <Connection type="STREAM" id="strmc1" gid="" name=""
      mode="AUTO/ONDEMAND" srcEntityId="src1" dstEntityId="dst1"
      profileId="unknown" authEntityId="" streamURL="" state=""
      timelineId="" canvasId='' windowId=''/>
      <Connection type="STREAM" id="recordc1" gid="" name=""
       mode="AUTO/ONDEMAND" srcEntityId="src1" dstEntityId="recordd1"
       profileId="unknown" authEntityId="" streamURL=""
       state="" timelineId="" canvasId='' windowId=''/>
      <Connection type="STREAM" id="strmc2" gid="" name=""
       mode="AUTO/ONDEMAND" srcEntityId="src2" dstEntityId="dst2"
       profileId="unknown" authEntityId="" streamURL=""
       state="" timelineId="" canvasId='' windowId=''/>
      <Connection type="STREAM" id="recordc2" gid="" name=""
      mode="AUTO/ONDEMAND" srcEntityId="src2" dstEntityId="recordd2"
      profileId="unknown" authEntityId="" streamURL=""
      state="" timelineId="" canvasId='' windowId=''/>
   </ConnectionList>
  </Session>
</StartSession>
```

Example 4.21. Basic Start Playback Session Request

```
<StartSession sessionId="playbackSessionID" sessionConfigNID="none">
  <Session id="sessionConfigNID"
       name="" state="" timelineId="">
   <GrpFileList>
       <GrpFile type="SyncPly" id="groupFileID"
       resourceNID="default.mediastreamfileresourcelist.
       msfrl_081a2b85-b62a-425f-a11a-7a7b6c3bc681"
       />
   </GrpFileList>
   <EntityList>
       <Entity type="DstPort" id="clientd1" name=""
           streamType="" presence="" gc="" state="" errorCode="" errorData="">
           <Port type="" resourceNID="" resourceTitle="playerpc" trickPlay="tru
       </Entity>
       <Entity type="PlyFile" id="recordd1" name=""
            streamType="" presence="" gc="" state="" errorCode="" errorData="">
           <File type=""
           resourceNID="default.mediastreamfileresourcelist.
           msfrl_cf058887-bf44-4728-a8c6-eddfa831a03e"
           resourceTitle="" grpFileId=""/>
       </Entity>
       <Entity type="DstPort" id="clientd2" name=""
            streamType="" presence="" gc="" state="" errorCode="" errorData="">
           <Port type="" resourceNID="" resourceTitle="playerpc" trickPlay="fal
       </Entity>
       <Entity type="PlyFile" id="recordd2" name=""
           streamType="" presence="" gc="" state="" errorCode="" errorData="">
           <File type=""
           resourceNID="default.mediastreamfileresourcelist.
           msfrl_c13887ae-73f4-4029-9d37-c4a366156d17"
           resourceTitle="" grpFileId="" />
       </Entity>
   </EntityList>
   <ConnectionList>
    <Connection id="strmc1" srcEntityId="recordd1" dstEntityId="clientd1" canva
    <Connection id="strmc2" srcEntityId="recordd2" dstEntityId="clientd2" canva
   </ConnectionList>
  </Session>
</StartSession>
```

Example 4.22. Advance Start Playback Session Request [Not Implemented]

```
<StartSession sessionId="playbackSessionID" sessionConfigNID="none">
 <Session id="sessionConfigNID" name=""
  state="" timelineId="">
  <Properties sessionConfigNID="" ownerUserJID=""
              dateCreated="" playlength=""/>
  <Groups>
      <Tag Name="" Value=""/>
  </Groups>
  <MetaData>
      <Tag Name=""/>
  </MetaData>
  <TimelineList>
      <Timeline id="">
          <Start WC="" Offset="" TC="" parentTimelineId=""/>
          <Last WC="" Offset="" TC="" TimeScale="" requestedAction=""/>
      </Timeline>
  </TimelineList>
  <GrpFileList>
      <GrpFile type="SyncPly" id="groupFileID"
      resourceNID="default.mediastreamfileresourcelist.
      msfrl_081a2b85-b62a-425f-a11a-7a7b6c3bc681"
      resourceTitle="" dateCreated="" recsessionConfigNID=""
      recSessionId="recordSessionID"
      syncEntityId="" recSessionConfigNID="recordSessionConfigNID"/>
  </GrpFileList>
  <DirList/>
  <ProfileList>
      <Profile id="" name="" profileNID="" ffTracks=""/>
  </ProfileList>
```

Example 4.23. Advance Start Playback Session Request [Not Implemented] (Continued)

```
<EntityList>
  <Entity type="DstPort" id="clientd1" name="" streamType="" presence="" gc="
      <User type="" userNID='' username='' userJID=""/>
      <Port type="" resourceNID="" resourceTitle="playerpc" trickPlay="true"/
  </Entity>
  <Entity type="PlyFile" id="recordd1" name="" streamType="" presence="" gc="
      <User type="" userNID='' username='' userJID=""/>
      <File type=""
      resourceNID="default.mediastreamfileresourcelist.
      msfrl_cf058887-bf44-4728-a8c6-eddfa831a03e"
      resourceTitle="" dirId="" grpFileId="" startOffset="" playLength=""/>
      <Port type="" resourceNID="" resourceTitle="" trickPlay=""/>
  </Entity>
  <Entity type="DstPort" id="clientd2" name="" streamType="" presence="" gc="
      <User type="" userNID='' username='' userJID=""/>
      <Port type="" resourceNID="" resourceTitle="playerpc" trickPlay="true"/
  </Entity>
  <Entity type="PlyFile" id="recordd2" name="" streamType="" presence="" gc="
      <User type="" userNID='' username=''userJID=""/>
      <File type=""
      resourceNID="default.mediastreamfileresourcelist.
      msfrl_c13887ae-73f4-4029-9d37-c4a366156d17"
      resourceTitle="" dirId="" grpFileId="" startOffset=""
      playLength=""/>
      <Port type="" resourceNID="" resourceTitle="" trickPlay=""/>
  </Entity>
</EntityList>
```

Example 4.24. Advance Start Playback Session Request [Not Implemented] (Continued)

```
<ConnectionList>
      <Connection type="STREAM" id="strmc1" gid="" name=""
      mode="AUTO/ONDEMAND" srcEntityId="recordd1" dstEntityId="clientd1"
      profileId="unknown" authEntityId="" streamURL="" state="" timelineId=""
       canvasId='' windowId=''/>
      <Connection type="STREAM" id="strmc2" gid="" name=""
      mode="AUTO/ONDEMAND" srcEntityId="recordd2" dstEntityId="clientd2"
      profileId="unknown" authEntityId="" streamURL="" state="" timelineId=""
      canvasId='' windowId=''/>
  </ConnectionList>
 </Session>
</StartSession>
```

StopSession

Used to terminate Sessions.

Example 4.25. StopSession Request

Table 4.4. StopSession Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<StopSession sessionId=""/>
```

Session Connection Operations

API calls for Session Connection Operations

View

Used to setup connections between Live sources and Playback destinations.

Use the SrcPort and DstPort to specify the resources to connect. Use either the name or id field returned in the GetPortList command. Use the srcEntityId or dstEntityId attribute specs in order to specify a connection to an existing src/dst. Use the canvasId and windowId to specify the canvas/window to be used for the connection.
Specify an optional Profile to further constrain the max available profile based on the admin settings. Again use the attribute form to reuse an existing profile.
Specify a PVR buffer length (in minutes) to request a buffered relay in the path of the connection. Use custom routing to override the default location of relays and PVR buffers.

Example 4.26. Basic View Request

A new session is always started. Returns an error if sessionId already exists. If no sessionId is specified a sessionId is generated by the system

Table 4.5. Basic View Request by Title

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500

Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<View sessionId="session1" sessionName="" id="strmc1" name=""
srcEntityId="" dstEntityId="" canvasId="" windowId="" restartSession="true">
    <SrcPort id="" portNID="" portTitle="tx1" pvrEnable="false"/>
    <DstPort id="" portNID="" portTitle="srcpc1"/>
</View>
```

Example 4.27. Basic View Request by Title

A new session is always started. If no sessionId is specified a sessionId is generated by the system

```
      For E.g. to connect a SrcPort "tx1" to a DstPort "xp2001_CHANNEL2"
      <View >
   <SrcPort portTitle="tx1"/>
   <DstPort portTitle="xp2001_Channel2"/>
</View>
```

Example 4.28. Basic View Request by NID

A new session is always started. Returns an error if sessionId already exists. If no sessionId is specified a sessionId is generated by the system

```
<View sessionId="session1" sessionName="" id="strmc1" name=""
srcEntityId="" dstEntityId="" canvasId="" windowId="" restartSession="true">
    <SrcPort id="" portNID="default.mediastreamsrcportresourcelist.
    mssprl_e973aef8-93f1-4388-a378-a77ddc364b2c" portTitle="" pvrEnable="false"/
    <DstPort id="" portNID="default.mediastreamdstportresourcelist.
    msdprl_1d9c4ced-5168-4d4d-bf7e-8b0011f676da" portTitle=""/>
</View>
```

Example 4.29. View Request with Profile Title

```
<View sessionId='mySession1234' id='myConn1234' name='' sessionName=''>
  <SrcPort id="" portNID="" pvrEnable='true' pvrBufferLength='30'/>
  <DstPort id="" portNID=""/>
  <Profile id="" profileTitle="V2D_LAN"/>
</View>
```

Example 4.30. View Request with Profile NID

```
<View sessionId='mySession1234' id='myConn1234' name='' sessionName=''>
  <SrcPort id="" portNID="" pvrEnable='true' pvrBufferLength='30'/>
  <DstPort id="" portNID=""/>
  <Profile id="" profileNID="nid1"/>
</View>
```

Example 4.31. View Request with PVR

[Not Implemented]

```
<View sessionId='mySession1234' id='myConn1234' name='' sessionName=''>
  <SrcPort id="" portNID="" pvrEnable='true' pvrBufferLength='30'/>
  <DstPort id="" portNID=""/>
  <Profile id="" profileNID=""/>
</View>
```

Example 4.32. View Request to add a connection from an existing Src to a new Dst

```
<View sessionId="recordSessionID" sessionName="" id="vstrm1" name=""
   srcEntityId="src1" dstEntityId="" canvasId="" windowId=""
   restartSession="">
    <SrcPort id="" portNID="" portTitle="" pvrEnable=""/>
    <DstPort id="vdst1" portNID="" portTitle="srcpc1"/>
</View>
```

Example 4.33. View Request to add a connection from an existing Src to a new Dst with profileid

```
<View sessionId="recordSessionID" sessionName="" id="vstrm1" name=""
   srcEntityId="src1" dstEntityId="" canvasId="" windowId=""
   restartSession="" profileid="profileId1">
    <SrcPort id="" portNID="" portTitle="" pvrEnable=""/>
    <DstPort id="vdst1" portNID="" portTitle="srcpc1"/>
</View>
```

Example 4.34. View Request to add a connection from an existing Src to existing Dst

```
<View sessionId="recordSessionID" sessionName="" id="vstrm2" name=""
  srcEntityId="src1" dstEntityId="vdst1" canvasId="" windowId=""
  restartSession="false">
 <SrcPort id="" portNID="" portTitle="" pvrEnable=""/>
 <DstPort id="" portNID="" portTitle=""/>
</View>
```

Example 4.35. View Request Response

The response to the view request will consist of the updated Session element

- *sessionId : id of session (Default UUID)* [Optional]
- *id : id of connection (Default UUID)* [Optional]
- *name : name of the Connection (Default conn_UUID)* [Optional]
- *sessionName : name of the session. Default is username and session start Time(srcuser1 Jun 25 00:15)* [Optional]
- *dstEntityId : Destination entity id in session* [Either entity id or portTitle Required]
- *srcEntityId : source entity id in session* [Either entity id or portTitle Required]
- *portTitle : Source/Destination resource Title (Matches to Entity name in the session doc)* [Either entity id or portTitle Required]
- *profileTitle : profile title to be used in session* [Optional]
- *profileNID : profile nid to be used in session* [Optional]
- *profileId : profile Id in session* [Optional]

```
<Session gc="true" id="Session_29BAD04A-DB32-466B-95EF-40D6729D6DFD"
  name="srcuser1 Jun 25 00:15"
  recordTimelineId="505f1b62-9a8e-4008-8c22-2d683ed40dfa.RecordTrack"
  state="Started"
  timelineId="505f1b62-9a8e-4008-8c22-2d683ed40dfa.StreamTrack"
  type="LIVE">
  <Properties dateCreated="1277424909504"
  ownerUserJID="srcuser1@localhost/pc1" playlength="" sessionConfigNID=""/>
  <TimelineList>
       <Timeline connGroupId="" connId=""
id="505f1b62-9a8e-4008-8c22-2d683ed40dfa.StreamTrack"
sessionId="Session_29BAD04A-DB32-466B-95EF-40D6729D6DFD">
     <Start Offset="0" WC="1277424909504" TC="" parentTimelineId=""/>
     <Last Offset="0" TimeScale="1" WC="1277424909504" TC="" requestedAction=""/>
     </Timeline>
     <Timeline connGroupId="" connId=""
      id="505f1b62-9a8e-4008-8c22-2d683ed40dfa.RecordTrack"
         sessionId="Session_29BAD04A-DB32-466B-95EF-40D6729D6DFD">
    <Start Offset="0" WC="1277424909504" TC="" parentTimelineId=""/>
    <Last Offset="0" TimeScale="1" WC="1277424909504" TC="" requestedAction=""/>
  </Timeline>
  </TimelineList>
  <GrpFileList/>
  <DirList/>
  <ProfileList>
    <Profile id="3980fb57-5fd7-4398-be26-2ca9056b0d63"
    profileNID="default.mediastreamprofilelist.mspl_e1cb45a9-f433-40b2-8156-258c0
    name="V2D_LAN" ffTracks=""/>
  </ProfileList>
```

Example 4.36. View Request Response (Continued)

```
  <EntityList>
    <Entity gc="true" id="c5d9ac1f-fb26-4f35-9626-a82e00aad228"
     name="tx1" presence="available" streamType="V2D" type="SrcPort" state="" err
      <Port resourceTitle="tx1"
       resourceNID="default.mediastreamsrcportresourcelist.
       mssprl_9df8a5a4-bad1-488a-a3da-d0edbfeae3f2" trickPlay=""/>
    </Entity>
    <Entity gc="true" id="72a3d1d7-6aa3-481c-ae32-02e7a0a9929a"
      name="pc1" presence="available" streamType="MULTI" type="DstPort" state=""
       <Port resourceTitle="pc1"
        resourceNID="default.mediastreamdstportresourcelist.
        msdprl_ea1024f3-b1a4-4b06-8395-ec6961b61637" trickPlay="true"/>
    </Entity>
  </EntityList>
  <ConnectionList>
       <Connection dstEntityId="72a3d1d7-6aa3-481c-ae32-02e7a0a9929a"
              id="Conn_4A12955D-E857-4540-BC45-B951AE9F4A68" name="tx1"
              srcEntityId="c5d9ac1f-fb26-4f35-9626-a82e00aad228" state="STREAMING
              streamURL="v2d://10.1.5.169/port=6060?bandwidth=10240000?
              avoption=Video?ismulticast=false?enablekbm=false"
              type="VIEW">
           <ConnSeg dstResourceNID="default.mediastreamrelayportresourcelist.
               msrprl_e3349f7e-5552-400e-8264-be7b3884b228"
              dstResourceTitle="srcrelay1"
              profileNID="profileNID"
              profileTitle="V2D_LAN"
              srcResourceNID="default.mediastreamsrcportresourcelist.
              mssprl_9df8a5a4-bad1-488a-a3da-d0edbfeae3f2"
              srcResourceTitle="tx1" state="STREAMING" errorCode="" errorData=""
            <ConnSeg
            dstResourceNID=""
               dstResourceTitle="padmajapc"
               profileNID="default.mediastreamprofilelist.
               mspl_b337c45e-5bea-4574-9909-9a623b9cc22d"
               profileTitle="V2D_LAN"
               srcResourceNID="" srcResourceTitle="srcrelay1"
               state="STREAMING" errorCode="" errorData="" />
        </Connection>
    <Groups>
      <Tag Name="Location" Value="Sunnyvale"/>
    </Groups>
</Session>
```

Example 4.37. View Request Response [Not Implemented]

```
<Session id="session_config_7A6C6DC1-A43B-4D0C-A5BC-F52FE1E2C39E"
name="session_config"
recordTimelineId="4d758c18-8602-46fb-aa0a-d7b96dda88b2.RecordTrack"
state="Started"
timelineId="4d758c18-8602-46fb-aa0a-d7b96dda88b2.StreamTrack" type="LIVE">
<Properties dateCreated="1277427023666" ownerUserJID="srcuser1@localhost/pc1"
 playlength=""
 sessionConfigNID="default.sessionconfiglist.
 scl_8f5fcbea-fed9-4399-8b52-cf1e4d158500" />
<TimelineList>
 <Timeline connGroupId="" connId=""
  id="4d758c18-8602-46fb-aa0a-d7b96dda88b2.StreamTrack"
  sessionId="session_config_7A6C6DC1-A43B-4D0C-A5BC-F52FE1E2C39E">
  <Start Offset="0" WC="1277427023666" TC="" parentTimelineId="" />
  <Last Offset="0" TimeScale="1" TC="" WC="1277427023666"
   requestedAction="" />
 </Timeline>
 <Timeline connGroupId="" connId=""
  id="4d758c18-8602-46fb-aa0a-d7b96dda88b2.RecordTrack"
  sessionId="session_config_7A6C6DC1-A43B-4D0C-A5BC-F52FE1E2C39E">
  <Start Offset="0" WC="1277427023666" TC="" parentTimelineId="" />
  <Last Offset="0" TimeScale="1" TC="" WC="1277427023666"
   requestedAction="" />
       </Timeline>
</TimelineList>
<GrpFileList />
<DirList />
<ProfileList />
```

Example 4.38. View Request Response [Not Implemented] (Continued)

```
 <EntityList>
<Entity id="SrcPort_CE1AE1E4-A3F6-4659-9837-F952577801CB" name="tx1"
 presence="available" streamType="V2D" type="SrcPort" gc="" state="" errorCode=""
  <User resourceNID="" resourceTitle="" userJID="" /> (Not implemented)
  <Port netIp="" netPort=""
     resourceNID="default.mediastreamsrcportresourcelist.
     mssprl_9df8a5a4-bad1-488a-a3da-d0edbfeae3f2"
 resourceTitle="tx1" trickPlay=""/>
  <PVR enable="" plyEntityId="" connId="" startTC="" sessionId="" recEntityId=""
 trimLength="" ffTracks="" />(Not implemented)
</Entity>
<Entity id="DstPort_A4890CD8-D221-439D-841F-2CF352F1A4CC" name="pc1"
 presence="available" streamType="MULTI" type="DstPort" gc="" state="" errorCode=
  <User resourceNID="" resourceTitle="" userJID="" />(Not implemented)
  <Port netIp="" netPort=""
 resourceNID="default.mediastreamdstportresourcelist.
 msdprl_ea1024f3-b1a4-4b06-8395-ec6961b61637"
 resourceTitle="pc1" trickPlay="true"/>
</Entity>
 </EntityList>
```

Example 4.39. View Request Response [Not Implemented] (Continued)

```
 <ConnectionList>
  <Connection authEntityId=""
 dstEntityId="DstPort_A4890CD8-D221-439D-841F-2CF352F1A4CC"
 gid="ConnGrp_086607B9-5BBA-45A1-BE8A-7F9F727D4665"
 id="Conn_6D39B2DF-D5A8-47FB-B4EB-197B7725B01F" mode="" name="tx1"
 profileId="" srcEntityId="SrcPort_CE1AE1E4-A3F6-4659-9837-F952577801CB"
 state="WAITING_FOR_SOURCE-DEST_CONTACTED" streamType="V2D" streamURL=""
 timelineId="" type="VIEW" canvasId='' windowId=''>
 (Not Implemented   gid=""  mode=""  profileId=""
 streamType="V2D" timelineId="")
 <ConnSeg
  dstResourceNID="default.mediastreamdstportresourcelist.
  msdprl_ea1024f3-b1a4-4b06-8395-ec6961b61637"
  dstResourceTitle="pc1"
  profileNID="default.mediastreamprofilelist.
  mspl_b337c45e-5bea-4574-9909-9a623b9cc22d"
  profileTitle="V2D_LAN"
  srcResourceNID="default.mediastreamrelayportresourcelist.
  msrprl_e3349f7e-5552-400e-8264-be7b3884b228"
  srcResourceTitle="srcrelay1" state="WAITING_FOR_SOURCE-DEST_CONTACTED"
  errorCode="" errorData="" />
  </Connection>

  </ConnectionList>
  <Groups>
 <Tag Name="Location" Value="Sunnyvale" />
  </Groups>
</Session>
```

Playback

Used to setup connections between Recorded media and Playback destinations.

Use the SrcFile/GrpFile and DstPort to specify the resources to connect. Identify the file by specifying the resourceNID or the dir, filename and recording date Optional sessionId can be specified Use the srcEntityId or dstEntityId attribute specs in order to specify a connection to an existing src/dst. Use the canvasId and windowId to specify the canvas/window to be used for the connection.
Specify an optional Media Store Port to override the default based on the admin settings.
Specify a SyncMediaId shared among media that should be played back "in Sync".

Support for Playback by Title/fileDate NOT implemented.

Example 4.40. Basic Playback Request by Title

[Not Implemented]

Table 4.6. Basic Playback Request by Title

Request	HTTP POST

Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<Playback>
  <SrcFile id="" fileTitle="" fileDate=""/>
  <DstPort id="" portTitle=""/>
</Playback>
```

Example 4.41. Basic Playback Request by NID

A new session is always started. Returns an error if sessionId already exists. If no sessionId is specified a sessionId is generated by the system

```
<Playback sessionId="playbackAPISessionID" id="strmc1" srcEntityId=""
dstEntityId="" restartSession="" canvasId='' windowId=''>
    <GrpFile id="grpId1" fileNID="" fileTitle=""/>
    <SrcFile id="" fileNID="nid1" fileTitle=""
    grpFileId="grpId1"/>
    <DstPort id="" portNID="" portTitle="srcpc1"/>
</Playback>
```

Example 4.42. Basic Playback Request with file name/date spec

[Not Implemented]

```
<Playback sessionId="" connGroupId="" id="" canvasId='' windowId=''>
  <SrcFile id="" fileTitle="" fileDate=""/>
  <DstPort id="" portNID=""/>
</Playback>
```

Example 4.43. Advanced Playback Request including grpFileId and StorePort spec

[Not Implemented]

```
<Playback sessionId="" connGroupId="" id="" canvasId='' windowId=''>
  <SrcFile id="" fileNID="" fileTitle="" fileDate="" grpFileId="">
    <StorePort id="" portNID=""/>
  </SrcFile>
  <DstPort id="" portNID=""/>
</Playback>
```

Example 4.44. Basic Playback Request by group file NID

A new session is always started with connections for all the clip files under the groupfile specified by the NID. Any existing session with the given sessionId is terminated. If no sessionId is specified a sessionId is generated by the system

```
<Playback sessionId="playbackAPISessionID" id="strmc1"
    srcEntityId="" dstEntityId="" restartSession="" canvasId='' windowId=''>
    <GrpFile id="grpId1"
     fileNID="default.mediastreamfileresourcelist.msfrl_3ed1f399-5026-43e8-9a8
     fileTitle=""/>
    <SrcFile id="" fileNID="" fileTitle="" grpFileId="grpId1"/>
    <DstPort id="" portNID="" portTitle="srcpc1"/>
</Playback>
```

Example 4.45. Basic Playback Request by srcEntityId

Start a playback connection to an existing playback source in an existing session

```
<Playback sessionId="playbackSessionID" id="vstrm1"
     srcEntityId="recordd1" dstEntityId="" restartSession="" canvasId='' windo
     <GrpFile id="grpId1" fileNID="" fileTitle=""/>
     <SrcFile id="" fileNID="" fileTitle="" grpFileId="grpId1"/>
     <DstPort id="vdst1" portNID="" portTitle="srcpc1"/>
</Playback>
```

Example 4.46. Basic Playback Request by srcEntityId and dstEntityId

Start a playback connection to an existing playback source in an existing session

```
     <Playback sessionId="playbackSessionID" id="vstrm2"
          srcEntityId="recordd1" dstEntityId="vdst1" restartSession="" canvasId=''
          <GrpFile id="grpId1" fileNID="" fileTitle=""/>
          <SrcFile id="" fileNID="" fileTitle="" grpFileId="grpId1"/>
          <DstPort id="" portNID="" portTitle=""/>
      </Playback>
```

Example 4.47. Playback Request Response

The response is similiar to the view request response will consist of the updated Session element

Play

This is the equivalent of issuing "Resume" with new speed.

timeScale
Playback Speed. Supported Speed 1, 2, 5, 10, 60

nowOffset
An Offset relative to NOW.

startOffset
An Offset relative to the START Time of the current Scope.

liveOffset
An Offset relative to the LIVE Time of the current Scope.

mediaTime
A Time reference to the TC attached to the media in the Media File

mediaOffset
An Offset relative to the start of the Media File. This provides a sequential index into a Media File irrespective of the TCs of the media in it.

mediaId
The ID of the media associated with the mediaOffset/mediaTime. The session/connGrp/conn offset is found using the mediaId + mediaOffset/mediaTime This is used to resolve ambiguity when there is more than one media in the scope

mediaConnId
The connId associated with the mediaOffset/mediaTime. The session/connGrp/conn offset is found using the connId + mediaOffset/mediaTime This is used to resolve ambiguity when the same media is in more than one connection in the scope

bookmarkId
The ID of the bookmark associated with the mediaOffset/mediaTime. The session/connGrp/conn offset is found using the bookmarkId + mediaOffset/mediaTime This is used to resolve ambiguity when there is more than one media in the scope

Example 4.48. Play Request

Table 4.7. Play Request

Request	HTTP POST

Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<Play sessionId="" timeScale=''/>
```

Example 4.49. Play Response

```
<Timeline sessionId="xyz123" connGroupId="" connId="" id="">
 <Start WC="1252016072510" TC="" Offset="0" parentTimelineId=""/>
 <Last WC="1252016160137" TC="" Offset="87627" TimeScale="1"
 requestedAction="ResumeFrom"/>
</Timeline>
```

Example 4.50. Play Request w/ nowOffset

```
<Play sessionId="" nowOffset=""/>
```

Example 4.51. Play Request w/ startOffset

```
<Play sessionId="" startOffset=""/>
```

Example 4.52. Play Request w/ liveOffset

```
<Play sessionId="" liveOffset=""/>
```

Example 4.53. Play Request w/ mediaOffset

```
<Play sessionId="" mediaOffset="" mediaId="" mediaConnId=""/>
```

Example 4.54. Play Request w/ mediaTime

```
<Play sessionId="" mediaTime="" mediaId="" mediaConnId=""/>
```

Example 4.55. Play Request w/ Bookmark

```
<Play sessionId="" bookmarkId="" mediaConnId=""/>
```

Record

Used to setup connections for Recording media.

Use the SrcPort and DstFile elements to specify the resources to connect. Identify the file by specifying the resourceNID or the dir, filename and recording date
Specify an optional Media Store Port to override the default based on the admin settings.
Specify a SyncMediaId shared among media that should be recorded "in Sync".

- *sessionId : id of session (Default UUID)* [Optional]
- *id : id of connection (Default UUID)* [Optional]
- *name : name of the Connection (Default conn_UUID)* [Optional]
- *sessionName : name of the session. Default is username and session start Time(srcuser1 Jun 25 00:15)* [Optional]
- *srcEntityId : source entity id in session* [Either entity id or portTitle Required]
- *portTitle : Source resource Title (Matches to Entity name in the session doc)* [Either entity id or portTitle Required]
- *DstFile id : id of rec entity* [Either entity id or portTitle Required]
- *DstFile fileTitle :recording file title* [Required if key=Title or new recording otherwise optional [Does not allow XML restricted special characters]]
- *DstFile Key :will be used to find out file existed or not. Values could be Title/MetaData* [optional]
- *DstFile action : what actions we can perform on Append/ Overwrite(Default is Error on Exist if don't specify any thing) (Overwriten[not implemented])* [optional]

- *DstFile keyMetaDataTagName : used along with key attribute. If key=Metadata then have to specify keyMetaDataTagName* [Required if key=MetaData otherwise optional]

- *DstFile MetaData element* [optional]

Example 4.56. Basic Record Request

Table 4.8. Basic Record Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<Record sessionId="" sessionName=""  id="" name="">
    <SrcPort id="" portNID="" portTitle="tx1"/>
    <DstFile id="" fileTitle="RecordingTitle">
        <MetaData>
            <Tag Name="ID">1299792256</Tag>
            <Tag Name="TestCase">BasicRecording</Tag>
        </MetaData>
    </DstFile>
</Record>

<Record>
    <SrcPort portTitle="tx1"/>
    <DstFile fileTitle="Recording1"/>
</Record>
```

Example 4.57. Basic Record Request with file name/dir spec

```
<Record sessionId="" sessionName=""  id="" name="" >
     <SrcPort id="" portNID="" portTitle="tx1"/>
     <DstFile id="" fileTitle="RecordingTitle" dirNID="dirNID1">
         <MetaData>
             <Tag Name="ID">1299792256</Tag>
             <Tag Name="TestCase">BasicRecordingWithDirNID</Tag>
         </MetaData>
     </DstFile>
 </Record>
```

Example 4.58. Record from an existing Src to file

```
<Record sessionId="id1" id="conn1" name="" srcEntityId="srcentityId1"
     <DstFile id="" fileTitle="" dirNID="">
         <MetaData>
             <Tag Name="ID">1299792256</Tag>
             <Tag Name="TestCase">RecordExistingSource</Tag>
         </MetaData>
     </DstFile>
 </Record>
```

Example 4.59. Append to Recording Based on Title

```
<Record sessionId="" id="" name="" >
     <SrcPort id="" portNID="" portTitle="tx1"/>
     <DstFile id="" fileTitle="fileTitle1"  key='Title'  action='Appe
 </Record>
```

Example 4.60. Append to Recording Based on MetaData

```
<Record sessionId="" id="" name="" srcEntityId="">
     <SrcPort id="" portNID="" portTitle="tx1"/>
     <DstFile id="" fileTitle="fileTitle1"
         key='MetaData'  action='Append' keyMetaDataTagName='ID'>
         <MetaData>
             <Tag Name="ID">1299792256</Tag>
             <Tag Name="TestCase">AppendToRecording</Tag>
         </MetaData>
     </DstFile>
 </Record>
```

Example 4.61. Record Request with Profile NID

```
<Record sessionId="" sessionName=""  id="" name="" >
     <Profile id="" profileNID="NID1" profileTitle=""/>
     <SrcPort id="" portNID="" portTitle="tx1"/>
     <DstFile id="" fileTitle="RecordingTitle" dirNID="dirNID1">
         <MetaData>
             <Tag Name="ID">1299792256</Tag>
             <Tag Name="TestCase">BasicRecordingWithDirNID</Tag>
         </MetaData>
     </DstFile>
 </Record>
```

Example 4.62. Record Request with Profile Title

```
<Record sessionId="" sessionName=""  id="" name="" >
     <Profile id="" profileNID="" profileTitle="V2D_LAN"/>
     <SrcPort id="" portNID="" portTitle="tx1"/>
     <DstFile id="" fileTitle="" dirNID="dirNID1">
         <MetaData>
             <Tag Name="ID">1299792256</Tag>
             <Tag Name="TestCase">BasicRecordingWithDirNID</Tag>
         </MetaData>
     </DstFile>
 </Record>
```

Example 4.63. Record from an existing Src with profile id

```
<Record sessionId="" sessionName=""  id="" name="" profileId="profile
     <Profile id="" profileNID="" profileTitle=""/>
     <SrcPort id="" portNID="" portTitle="tx1"/>
     <DstFile id="" fileTitle="" dirNID="dirNID1">
         <MetaData>
             <Tag Name="ID">1299792256</Tag>
             <Tag Name="TestCase">BasicRecordingWithDirNID</Tag>
         </MetaData>
     </DstFile>
 </Record>
```

Example 4.64. Advanced Record Request including grpFileId and StorePort spec

[Not Implemented]

```
<Record sessionId="" connGroupId="" id="">
  <SrcPort id="" portNID=""/>
  <DstFile id="" grpFileId="" fileTitle="" dirTitle="" dirNID="">
    <StorePort id="" portNID=""/>
  </DstFile>
</Record>
```

Example 4.65. Record Request Response

```
<Session name="srcuser1 Mar 10 21:24" timelineId="c29ccf26-74c4-4e5a-9d3e-f2818
 recordTimelineId="c29ccf26-74c4-4e5a-9d3e-f2818e54cd1d.RecordTrack"
 type="LIVE" gc="true" id="set_file_metada_test_1299792256" state="Started">
   <Properties sessionConfigNID="" ownerUserJID="srcuser1@localhost/srcpc1" dat
   <TimelineList/>
   <GrpFileList/>
   <DirList/>
   <ProfileList>
     <Profile id="3980fb57-5fd7-4398-be26-2ca9056b0d63"
     profileNID="default.mediastreamprofilelist.mspl_e1cb45a9-f433-40b2-8156-25
     name="V2D_LAN"/>
   </ProfileList>
   <EntityList>
   <Entity gc="false" id="021cf69d-b6e4-4f5e-88e2-56930830d243" type="SrcPort"
     <Port resourceNID="default.mediastreamsrcportresourcelist.mssprl_e22023ce-
   </Entity>
   <Entity gc="false" id="47bdcfe0-b7a1-4e73-a1b9-01318f99e654" type="RecFile"
      <File resourceTitle="set_file_metada_test_1299792256" grpFileId="" key=""
           clipId="47bdcfe0-b7a1-4e73-a1b9-01318f99e654" resourceNID="default.m
           playLength="0" mediaGroupId="">
         <MetaData>
             <Tag Name="ID">1299792256</Tag>
             <Tag Name="TestCase">AddMediaStreamFile</Tag>
         </MetaData>
      </File>
    </Entity>
   </EntityList>
```

Example 4.66. Record Request Response [continued]

```
  <ConnectionList>
   <Connection dstEntityId="47bdcfe0-b7a1-4e73-a1b9-01318f99e654"
      id="set_file_metada_test_1299792256" srcEntityId="021cf69d-b6e4-4f5e-88e2
       type="RECORD" name="set_file_metada_test_1299792256" canvasId="set_file_
       windowId="set_file_metada_test_1299792256" state="WAITING_FOR_SOURCE-DES
   </ConnectionList>
   <Groups>
      <Tag Name="Location" Value="Sunnyvale"/>
   </Groups>
</Session>
```

ResumeRecord

Example 4.67. ResumeRecord Request

Table 4.9. ResumeRecord Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<ResumeRecord sessionId='xyz123'/>
```

Example 4.68. ResumeRecord Response

```
<Timeline sessionId="xyz123" connGroupId="" connId="" id="">
 <Start WC="1252016072510" TC="" Offset="0" parentTimelineId=""/>
 <Last WC="1252016479279" TC="" Offset="406769" TimeScale="1"
 requestedAction="Resume"/>
</Timeline>
```

SwitchSrc

Used to switch the src for an existing connections.

Use the SrcEntityId to specify the source to switch to.

Example 4.69. SwitchSrc Request

Table 4.10. SwitchSrc Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session

Authorization	Basic (Required)

```
<SwitchSrc sessionId="recordSessionID" connId="strmc2" newSrcEntityId="src1"/
```

Example 4.70. SwitchSrc Response

```
<Connection id="strmc2" srcEntityId="src1" dstEntityId="myDst1234"/>
```

PauseRecord

Example 4.71. PauseRecord Request

Table 4.11. PauseRecord Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<PauseRecord sessionId='xyz123'/>
```

Example 4.72. PauseRecord Response

```
<Timeline sessionId="xyz123" connGroupId="" connId="" id="">
 <Start WC="1252016072510" TC="" Offset="0" parentTimelineId=""/>
 <Last WC="1252016452472" TC="" Offset="379962" TimeScale="0"
 requestedAction="Pause"/>
</Timeline>
```

Pause

Example 4.73. Pause Request

Table 4.12. Pause Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<Pause sessionId='xyz123'/>
```

Example 4.74. Pause Response

```
<Timeline sessionId="xyz123" connGroupId="" connId="" id="">
 <Start WC="1252016072510" TC="" Offset="0" parentTimelineId=""/>
 <Last WC="1252016119868" TC="" Offset="47358" TimeScale="0" requestedAction="Pa
</Timeline>
```

SkipBack

The "skipOffset" is always a "nowOffset". This is the equivalent of issuing "Resume" with a negative "nowOffset"

Example 4.75. SkipBack Request

Table 4.13. SkipBack Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session

Authorization	Basic (Required)

```
<SkipBack sessionId='xyz123' skipOffset='2000' timeScale='' />
```

Example 4.76. SkipBack Response

```
<Timeline sessionId="xyz123" connGroupId="" connId="" id="">
 <Start WC="1252016072510" TC="" Offset="0" parentTimelineId=""/>
 <Last WC="1252016400432" TC="" Offset="325922" TimeScale="1"
 requestedAction="SkipBack"/>
</Timeline>
```

SkipFwd

The "skipOffset" is always a "nowOffset". This is the equivalent of issuing "Resume" with a positive "nowOffset"

Example 4.77. SkipFwd Request

Table 4.14. SkipFwd Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<SkipFwd sessionId='xyz123' skipOffset='1000' timeScale='' />
```

Example 4.78. SkipFwd Response

```
<Timeline sessionId="xyz123" connGroupId="" connId="" id="">
 <Start WC="1252016072510" TC="" Offset="0" parentTimelineId=""/>
 <Last WC="1252016423002" TC="" Offset="349492" TimeScale="1"
 requestedAction="SkipFwd"/>
</Timeline>
```

Catchup

This is the equivalent of issuing "Resume" with a liveOffset=0

Example 4.79. Catchup Request

Table 4.15. Catchup Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<Catchup sessionId=""/>
```

Example 4.80. Catchup Response

```
<Timeline sessionId="xyz123" connGroupId="" connId="" id="">
 <Start WC="1252016072510" TC="" Offset="0" parentTimelineId=""/>
 <Last WC="1252016160137" TC="" Offset="87627" TimeScale="1"
 requestedAction="Catchup"/>
</Timeline>
```

Session Participant API

API calls for Session Participants

Invite

[Not Implemented]

Issue an Invite to explicitly invite another user to a session

Example 4.81. Simple Invite by userId or jid

Table 4.16. Simple Invite by userId or jid

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500

Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<Invite sessionId="" id="" userId="" jid=""/>
```

Example 4.82. Invite by userId(who) and/or deviceId(from where)

```
<Invite sessionId="" id="" userId="" deviceId=""/>
```

Join

In response to the Join - the system will send back an "MUC invite" to an MUC room. The client should respond to that by doing a standard "XMPP MultiUserChat Join" to enter the MUC room.

- Join is once per session with or without invite.
- An invite is required to join "Invitation Only" Sessions (the default)
- Sessions that have been "Published" can be joined without an explicit "Invite"

Example 4.83. Join without invite Request

Table 4.17. Join without invite Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<Join sessionId=""/>
```

Example 4.84. Join without invite Response

```
<Session .../>
```

Example 4.85. Join without invite Response - MUC Room Invite Message

```
<Session .../>
```

Example 4.86. Join with invite

```
<Join sessionId="" inviteId=""/>
```

Kick

[Not Implemented]

Issue a Kick to explicitly force a user to leave a session

Example 4.87. Kick

Table 4.18. Kick

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<Kick sessionId="" id="" userName="" jid="" userId="" deviceId=""/>
```

Bookmark Operations

API calls for Bookmarks

AddBookmark

The location of the bookmark is derived by the App Server based on the "Time" of the specified Connection (or ConnGroup). All the fields in the RequestXML are optional. If an 'id' is not specified - it will be generated by the system. Bookmarks can be added without a title or comment and then updated with all the info later.

Example 4.88. AddBookmark Request

Table 4.19. AddBookmark Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<Bookmark whereFileNID="" whereSessionInfo="" title="" TC="" offset="">
 <Description />
 <MetaData>
  <Tag Name=""></Tag>
 </MetaData>
</Bookmark>
```

- *Either whereSessionInfo or whereFileNID Required*

  whereSessionInfo is key:value pairs e.g. "@whereSessionInfo": "sessionId:,connGroupId:,connId:,entityId

  whereFileNID is the NID of the file to add bookmark to.

Example 4.89. AddBookmark Response

```
<AddBookmark NID="" title="">
   <Bookmark TC="" offset="" owner=""
    dateCreated="" fileNID="">
    <Description />
    <MetaData>
     <Tag Name=""></Tag>
    </MetaData>
   </Bookmark>
</AddBookmark>
```

DeleteBookmark

Example 4.90. DeleteBookmark Request

Table 4.20. DeleteBookmark Request

Request	HTTP POST

Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<DeleteBookmark whereNID="" whereTCRange="" whereDateRange=""
 whereFileNID="" whereSessionInfo=""/>
```

- *whereNID or whereSessionInfo or whereFileNID* Required

  whereNID is the NID of the bookmark to delete.

  whereTCRange or whereDateRange format: "From:2014-11-17 00:00:00.0000,To:2014-11-19 15:00:00.0000"

Example 4.91. DeleteBookmark Response

```
<DeleteBookmark total="">
 <Bookmark NID=""/>
 <Bookmark NID=""/>
 <Bookmark NID=""/>
</DeleteBookmark>
```

GetBookmark

Example 4.92. GetBookmark Request

Table 4.21. GetBookmark Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

Example 4.93. GetBookmark Request

```
<GetBookmark startFrom="" countToFetch="" sortOrder=""
 sortBy="ascending/descending"
 selectFormat="BASIC/EXTENDED"
 whereNID=""
 whereTextSubStr=""
 whereTCRange=""
 whereFileNID=""
 whereSessionInfo=""/>
```

- *Either whereSessionInfo or whereFileNID Required*

  whereSessionInfo is key:value pairs e.g. "@whereSessionInfo": "sessionId:,connGroupId:,connId:,entityId"

  whereFileNID is the NID of the file which contains bookmarks.

Example 4.94. GetBookmark Response - BASIC Format

```
 <GetBookmark from="" total="">
<Bookmark NID="" title="" TC="" offset="" owner=""
    dateCreated="" fileNID=""/>
<Bookmark NID="" title="" TC="" offset="" owner=""
    dateCreated="" fileNID=""/>
<Bookmark NID="" title="" TC="" offset="" owner=""
    dateCreated="" fileNID=""/>
 </GetBookmark>
```

Example 4.95. GetBookmark Response - EXTENDED Format

```
<GetBookmark from="" total="">
<Bookmark NID="" title="" TC="" offset="" owner=""
    dateCreated="" fileNID="">
 <Description />
 <MetaData>
  <Tag Name=""></Tag>
 </MetaData>
</Bookmark>
</GetBookmark>
```

NewBookmark

Return new bookmark for given session or file nid by filling the time code and offset values. (Note: this call don't add bookmark in database. To add bookmark in database user has to use AddBookmark along with this). []

Example 4.96. NewBookmark Request

Table 4.22. NewBookmark Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session
Authorization	Basic (Required)

```
<NewBookmark whereFileNID=""  whereSessionInfo="sessionId:,connGroupId:,connId
```

- *Either whereSessionInfo or whereFileNID Required*

Example 4.97. NewBookmark Response

```
<NewBookmark>
 <Bookmark title="" TC="" offset="">
  <Description />
  <MetaData>
     <Tag Name=""></Tag>
  </MetaData>
 </Bookmark>
   </NewBookmark>
```

UpdateBookmark

Example 4.98. UpdateBookmark Request

Table 4.23. UpdateBookmark Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None

Resource Path	/Session
Authorization	Basic (Required)

```
<UpdateBookmark whereNID="">
  <Bookmark title="" TC="" offset="">
   <Description />
   <MetaData>
    <Tag Name=""></Tag>
   </MetaData>
  </Bookmark>
</UpdateBookmark>
```

- *whereNID* Required

Example 4.99. UpdateBookmark Response

```
<UpdateBookmark>
 <Bookmark NID="" title="" TC="" offset=""
     owner="" dateCreated="" fileNID="">
  <Description />
  <MetaData>
   <Tag Name=""></Tag>
  </MetaData>
 </Bookmark>
</UpdateBookmark>
```

Authorization

External Authorization

Authorize Request

Auth user can reply back with his decision allow/deny/revoke using Authorize request

Example 4.100. Authorize Request

Table 4.24. Authorize Request

Request	HTTP POST
Accept(No spaces or newlines)	"application/xml"
Status	200, 401, 500
Query Parameters	None
Resource Path	/Session

Authorization	Basic (Required)

```
<Authorize sessionId='' connId='' action='ALLOW/DENY/REVOKE' reason='' />
```

Example 4.101. Authorize Response

```
   <Authorize>
  <AuthConnectionInfo ...../>
   <!-- Check authConnectionInfo element in ConnectionUpdate Section of Update
</Authorize>
```

Chapter 5. Session Notifications

EntityPresence Notification

Every time an entity joins or leaves the room a presence update is sent into the room

Example 5.1. EntityPresence Notification

```
<EntityPresence entityId="" sessionId="" jid="" value=""/>
```

ConnState Notification

Every time a connection state changes - a notification is sent into the room. If value is Error - then code and msg are valid The list of possible connections states are mentioned below

Example 5.2. ConnState Notification: CONNECTED

```
<ConnState sessionId="" connId="" value='<see below>'>
  <Timeline/>
</ConnState>

The list of possible connections states are mentioned below
[desc not present in the notification]
The connection states are also reflected in the Connection
element within the Session

<ConnStateList>
  <ConnState value="CREATED"  desc="Stream has been created"/>
  <ConnState value="AUTHORIZING"  desc="Waiting for authorization"/>
  <ConnState value="CHECKING_AVAILABILITY"
             desc="Waiting for the source/dest resource to acquire a contactJID"
  <ConnState value="WAITING_FOR_DEST"
             desc="Waiting for destination to join the room"/>
  <ConnState value="WAITING_FOR_SOURCE-DEST_CONTACTED"
             desc="Destination notified of stream"/>
  <ConnState value="WAITING_FOR_SOURCE-DEST_ACKNOWLEDGED"
             desc="Destination waiting for source"/>
  <ConnState value="WAITING_FOR_STREAMURL"
             desc="Destination waiting for streamURL"/>
  <ConnState value="WAITING_FOR_STREAMSTART"
             desc="Destination waiting for stream started notification"/>
  <ConnState value="STREAMING"  desc="Stream started"/>
  <ConnState value="STREAM_PAUSED"  desc="Stream paused"/>
  <ConnState value="DENIED_XAUTH"  desc="Authorization denied"/>
  <ConnState value="REVOKED_XAUTH"  desc="Authorization revoked"/>
  <ConnState value="END_OF_MEDIA"  desc="End of Media"/>
  <ConnState value="SETUP_ERROR"  desc="Error while setup of stream"/>
  <ConnState value="START_ERROR"  desc="Error when starting the stream"/>
  <ConnState value="SOURCE_ERROR"  desc="Source setup error"/>
  <ConnState state="MEDIA_ERROR"  desc="Media error"/>
</ConnStateList>

The Timeline element is same as the Session Timeline mentioned earlier,
but now specific to the scope of the Connection
```

Example 5.3. ConnState Notification: Error

```
<ConnState sessionId="" connId="" value='ERROR' errorCode='NNNN' errorData='Erro
```

SessionState Notification

Every time a Session state changes - a notification is sent into the room. If value is Error - then code and msg are valid The possible values in the SessionState Notification are CREATED and DELETED

Example 5.4. SessionState Notification

```
  <SessionState sessionId="" value="" />
```

Timeline Notification

A Timeline is a collection of 3 attributes that serves as a the basic unit for sharing Time/Offset information among Entities. It allows for Time/Offset exchange without being subject to the delays in message transmission/processing. Using it, one Entity can derive/track the Current Offset of another entity (to the extent that their WCs are in Sync).

Timelines exist Hierarchically for:

- the session
- all connection groups
- all connections
- all entities.

A Basic Timeline correlates:

- lastWC = the WC when the last action that affected this Timeline happened
- lastOffset = the Offset of this Entity w.r.t. its START at that time
- TimeScale = the rate at which the Offset has been changing since then - reflects the lastAction: eg: Play = 1, Stop = 0, Rewind = -1, 15xFF = 15

To track fully Time/Offset, the Timline includes Info about the Start of the Entity.

- StartWC = The WC at the time this Enitity was Started. This is used to derive the Entity's LIVE Time at any moment.
- StartOffset = The difference between the 0 offset of this Entity and the 0 Offset of it's current Parent Entity This is used to translate a parent Entity Offset to a child Entity Offset.

- PVR actions essentially specify a new "target" Timeline for the scope they are issued at.
- These then result in timeline change requests for entities that make up that scope
- When the entites achieve this timeline, they send timeline update notifications into the session.
- If all goes well, these will match the requested timeline.
- The other event that causes Timeline Notifications is when Connection grouping/hierarchy changes.

If the difference between the Start and Last WC is equal to the difference between the Start and Last Offset then you are "Live/CaughtUp"

Example 5.5. Timeline Notification

```
<Timeline sessionId="" connGroupId="" connId="" id="">
  <Start WC="2341234" TC="" Offset="3456" parentTimelineId=""/>
  <Last WC="7444674" TC="" Offset="9678687" TimeScale="0"
   requestedAction="Pause"/>
</Timeline>
```

Appendix A. Data Model

RX Display Config/Status Parameter Listing

- *Background enable="true/false"* [Required]
- *Background zIndex=""* [Required]
- *Background content="0xRRGGBB/LOCALVIDEO"* [Required]
- *Window id=""* [Required]
- *Window enable="true/false"* [Required]
- *Window zIndex=""* [Required]
- *Window content="SPLASH/REMOTEVIDEO"* [Required]
- *vpos="ALIGNTOP/ALIGNMIDDLE/ALIGNBOTTOM/NNNpx"* [Required]
- *hpos="ALIGNLEFT/ALIGNMIDDLE/ALIGNRIGHT/NNNpx"* [Required]
- *hres="NNNpx"* [Required]
- *vres="NNNpx"* [Required]
- *aspect="SourceWidth/SourceHeight/Auto/None"* [Required]
- *transparency="NNNpx"* [Required]

RX DisplayConfig

```
<Canvas>
  <Background zIndex="0" content="LOCALVIDEO" enable="true"/>
  <Window id="id" zIndex="1" content="REMOTEVIDEO" enable="true">
    <Geometry vpos="ALIGNTOP"
        hpos="ALIGNLEFT"
        vres="75%"
        hres="75%"
        aspect="Auto"
        transparency="20%"/>
   </Window>
    </Canvas>
```

RX DisplayStatus

```
<Canvas hres="1280px" vres="1024px">
   <Display id="" hres="1280px" vres="1024px" lut="15 [VESA 1280x1024@60 Hz]"/>
    <Background zIndex="0" content="LOCALVIDEO" enable="true"/>
    <Window id="" enable="true" zIndex="1" content="REMOTEVIDEO">
      <Geometry vpos="0px" hpos="0px"
         vres="768px" hres="960px"
         transparency="20%"/>
      <Source hres="960px" vres="768px"
         hscale="75%" vscale="75%"
         lut="Server Splash Screen 1280x1024@60.0"/>
   </Window>
</Canvas>
```

Remote Operated Client Display Config/Status Parameter Listing

- *Background enable="true/false"* [Required]
- *Background zIndex=""* [Required]
- *Background content='OS/Image/Session'* [Required]
- *ImageContent imageURL=""* [Required]
- *ImageContent color=""* [Required]
- *SessionContent enableConnPanel='true/false'* [Required]
- *SessionContent enableBookmarkPanel='true/false'* [Required]
- *SessionContent enableCtrlPanel='true/false'* [Required]
- *SessionContent enableHeaderPanel='true/false'* [Required]
- *SessionContent enablePlayerPanel='true/false'* [Required]
- *SessionContent sessionId=''* [Required]
- *SessionContent content="0xRRGGBB/LOCALVIDEO"* [Required]
- *Layout id=""* [Required]
- *Layout type="Custom/AutoGrid/AutoStrip"* [Required]
- *Layout nRow=""* [Required if type='AutoGrid']
- *Layout nCol=""* [Required if type='AutoGrid']
- *Layout colWidth=""* [Required if type='AutoGrid/AutoStrip']
- *Layout rowHeight=""* [Required if type='AutoGrid/AutoStrip']
- *Layout hpad=""* [Required if type='AutoGrid/AutoStrip']

- *`Layout vpad=""`* [Required if type='AutoGrid/AutoStrip']
- *`Layout stripPos=""`* [Required if type='AutoStrip']
- *`Layout nStrip=""`* [Required if type='AutoStrip']
- *`Layout border="show/hide/autohide"`* [Required if type='Custom']
- *`Layout borderStyle="[L:T:R:B]/[css]"`* [Required if type='Custom']
- *`Layout header="hide/show/sutohide"`* [Required if type='Custom']
- *`Layout headerStyle="basic/extended/[css]"`* [Required if type='Custom']
- *`Layout active="true/false"`* [Required]
- *`Window id=""`* [Required]
- *`Window`* [Required]
- *`Window content='MediaThumbnail/MediaStream'`* [Required]
- *`Window zIndex=""`* [Required]
- *`Window content="SPLASH/REMOTEVIDEO"`* [Required]
- *`ConnectionContent MediaStreamIOPortTypeTitle=""`* [Required]
- *`ConnectionContent MediaStreamIOPortElementID=''`* [Required]
- *`ConnectionContent sessionId=''`* [Required]
- *`ConnectionContent connId=''`* [Required]
- *`ConnectionContent srcKbmCtrl=''`* [Not Supported]
- *`ThumbnailContent MediaStreamIOPortTypeTitle=""`* [Required]
- *`ThumbnailContent MediaStreamIOPortElementID=''`* [Required]
- *`ThumbnailContent imageURL=""`* [Required]
- *`ThumbnailContent refreshInterval=''`* [Required]
- *`Properties fps`* [Required]
- *`Properties maximize='true/false''`* [Required]
- *`Properties mute='true/false'`* [Required]
- *`Properties remoteKbmCtrl='true/false'`* [Required]
- *`Properties scale`* [Required]
- *`Properties visibility='true/false'`* [Required]
- *`Properties volume`* [Required]
- *`Properties aspect`* [Required]

- *Properties border* [Required]
- *Properties hpos* [Required]
- *Properties vpos* [Required]
- *Properties hres* [Required]
- *Properties vres* [Required]
- *Properties layoutPos* [Required]
- *Properties scrollHPos* [Required]
- *Properties scrollVPos* [Required]
- *Properties transparency* [Required]

Remote Operated Client DisplayConfig

```
<Display configurable="" hres="AUTO" vres="AUTO">
    <Panel id="" enable="" content="">
       <Geometry zIndex="" border="" vpos="0"
             hpos="0" vres="100%" hres="100%"
             aspect="" transparency="">
       </Geometry>
    </Panel>
</Display>

<Canvas id="" hres="1280" vres="1024">
   <Background enable="" zIndex="" content="">
      <ImageContent imageURL="" color=""></ImageContent>
      <SessionContent enableConnPanel="" enableBookmarkPanel=""
                      enableCtrlPanel="" enableHeaderPanel=""
                      enablePlayerPanel="" sessionId="">
       </SessionContent>
   </Background>

   <Layout id="LayoutId1" type="" nRow="" nCol="" colWidth=""
           rowHeight="" hpad="" vpad="" stripPos="" nStrip=""
           border="hide" borderStyle="[8px]" header="hide"
           headerStyle="basic" active="true">
   </Layout>

   <Window id="Win1" content="">
      <ConnectionContent MediaStreamIOPortTypeTitle=""
                         MediaStreamIOPortElementID=""
                         sessionId="" connId="" vres=""
                         hres="" srcKbmCtrl="">
      </ConnectionContent>
      <ThumbnailContent MediaStreamIOPortTypeTitle=""
```

```
                        MediaStreamIOPortElementID=""
                        imageURL="" refreshInterval="">
      </ThumbnailContent>
      <Properties zIndex="100" border="" vpos="0" hpos="0"
                  vres="100%" hres="100%" aspect=""
                  transparency="0" layoutId="LayoutId1"
                  layoutPos="" visibility="true"
                  maximize="false" fps="15"
                  scale="fitWindow" mute="false"
                  volume="50%" remoteKbmCtrl="false"
                  scrollHPos="0" scrollVPos="0">
      </Properties>
    </Window>
</Canvas>
```

Remote Operated Client DisplayStatus

```
<Display configurable="" hres="AUTO" vres="AUTO">
    <Panel id="" enable="" content="">
       <Geometry zIndex="" border="" vpos="0"
             hpos="0" vres="100%" hres="100%"
             aspect="" transparency="">
       </Geometry>
    </Panel>
</Display>

<Canvas id="" hres="1280" vres="1024">
   <Background enable="" zIndex="" content="">
      <ImageContent imageURL="" color=""></ImageContent>
      <SessionContent enableConnPanel="" enableBookmarkPanel=""
                      enableCtrlPanel="" enableHeaderPanel=""
                      enablePlayerPanel="" sessionId="">
       </SessionContent>
   </Background>

   <Layout id="LayoutId1" type="" nRow="" nCol="" colWidth=""
           rowHeight="" hpad="" vpad="" stripPos="" nStrip=""
           border="hide" borderStyle="[8px]" header="hide"
           headerStyle="basic" active="true">
   </Layout>

   <Window id="Win1" content="">
      <ConnectionContent MediaStreamIOPortTypeTitle=""
                         MediaStreamIOPortElementID=""
                         sessionId="" connId="" vres=""
                         hres="" srcKbmCtrl="">
      </ConnectionContent>
      <ThumbnailContent MediaStreamIOPortTypeTitle=""
                        MediaStreamIOPortElementID=""
                        imageURL="" refreshInterval="">
      </ThumbnailContent>
      <Properties zIndex="100" border="" vpos="0" hpos="0"
                  vres="100%" hres="100%" aspect=""
                  transparency="0" layoutId="LayoutId1"
                  layoutPos="" visibility="true"
                  maximize="false" fps="15"
                  scale="fitWindow" mute="false"
                  volume="50%" remoteKbmCtrl="false"
                  scrollHPos="0" scrollVPos="0">
      </Properties>
    </Window>
</Canvas>
```

Appendix B. REST API - Basic Tutorial

The following tutorial walks you through a basic set of calls used in the API and how to execute them.

REST Tester

Prerequisites for the tutorial:

- Login to admin UI
- Add user: user1
- Add device(webclient): webclient

We will use the BarcoRESTdiagnostics tool for demonstrating the use of the API.

BarcoRESTdiagnostics: http://<msip>/clients/gui/player/resttester/BarcoRESTdiagnostics.html

Important fields in the tester are:

- IPaddress: Enter your Management Server IP Address
- User: Enter the username for an existing user.
- Password: Enter password for above user
- Resource: Resource is the web client resource existing in the MS
- Service version: Service version is the current service version of your MS.
- Login: Link to login the user with above details
- Logout: Link for user logout
- URL: Enter REST URL here http:<msip>/controller/<resource_path>?<query_params>
- Accept Header: The format in which you expect your response from server (e.g. application/json)
- HTTP Method: Allows to choose the method type for your REST request.
- SUBMIT: Used to submit your REST requests
- Request: Hidden box, which is visible when request method is POST. Add your request body here for POST/PUT requests.
- Response: Displays the response for your REST request.
- Notifications: Displays the notifications being sent by server.

Service Login

Open BarcoRESTdiagnostics.html in a browser window.

Following are the details to be filled out in the REST Tester page for executing the Service Login request. Click on the 'Login' link after filling in the below details.

- IPaddress: MS ip address
- User: Enter the username for an existing user.
- Password: Enter password for above user
- Resource: Resource is the web client resource existing in the MS
- Service version: Service version is the current service version of your MS.

Service Login Response is available in the Response text field and a Long Polling channel is established for the User. **To see more details about how a Long polling notification channel is setup, please refer to the ServiceLogin API call in this document.**

GetPort - Sources

Fetching the list of source ports from the system.

Following are the details to be filled out in the REST Tester page for executing the GetPort request

- URL: http://10.51.51.72/controller/Resource/Port?type=SrcPort
- METHOD: HTTP GET
- Accept Header: application/xml

Upon hitting SUBMIT, a response will appear in the Response text field. From the response, please note down title for one of the source ports. For e.g. "tx1"

GetPort - Destination

Fetching the list of destination ports from the system.

Following are the details to be filled out in the REST Tester page for executing the GetPort request

- URL: http://10.51.51.72/controller/Resource/Port?type=DstPort
- METHOD: HTTP GET
- Accept Header: application/xml

Upon hitting SUBMIT, a response will appear in the Response text field. From the response, please note down title for one of the destination ports. For e.g. "rx1"

View

View a source stream on the destination.

Following are the details to be filled out in the REST Tester page for executing the View request. Enter the request body in the Request field which appears on selection of the HTTP POST method from drop down. Hit SUBMIT to send the request.

- URL: http://10.51.51.72/controller/Session
- METHOD: HTTP POST
- Accept Header: application/xml
- REQUEST BODY:

<View> <DstPort portTitle="rx1" /> <SrcPort portTitle="tx1" /> </View>

From the response, please note down session ID. Should be found as an attribute "@id" at the bottom of Session response

StopSession

Stop the session started by the View call

Following are the details to be filled out in the REST Tester page for executing the Stop Session request. Enter the request body in the Request field which appears on selection of the HTTP POST method from drop down. Hit SUBMIT to send the request.

- URL: http://10.51.51.72/controller/Session
- METHOD: HTTP POST
- Accept Header: application/xml
- REQUEST BODY:

<StopSession sessionId="" />

Please use the sessionId from the View response

Record

Start a recording session

Following are the details to be filled out in the REST Tester page for executing the Record request Enter the request body in the Request field which appears on selection of the HTTP POST method from drop down. Hit SUBMIT to send the request.

- URL: http://10.51.51.72/controller/Session
- METHOD: HTTP POST
- Accept Header: application/xml
- REQUEST BODY:
<Record> <DstFile fileTitle="tx1record" /> <SrcPort portTitle="tx1" /> </Record>

Please use the sessionId from the Record response and run the StopSession(see above) to stop the recording

GetFile - Recordings

Fetch the recording files

Following are the details to be filled out in the REST Tester page for executing the GetFile request. Hit SUBMIT to send the request.

- URL: http://10.51.51.72/controller/Resource/File
- METHOD: HTTP GET
- Accept Header: application/xml

Please note down the fileNID for one of the files from the GetFile response. e.g. "default.mediastreamfileresourcelist.msfrl_eb2bbd33-285b-4c1a-babd-20e7bb19df07"

Playback

Start a playback session

Following are the details to be filled out in the REST Tester page for executing the Playback request Enter the request body in the Request field which appears on selection of the HTTP POST method from drop down. Hit SUBMIT to send the request.

- URL: http://10.51.51.72/controller/Session
- METHOD: HTTP POST
- Accept Header: application/xml
- REQUEST BODY:
<Playback> <DstPort portTitle="rx1" /> <SrcFile fileNID="default.mediastreamfileresourcelist.msfrl_eb2bbd33-285b-4c1a-babd-20e7bb19df07" /> </Playback>

Please use the sessionId from the Playback response and run the StopSession(see above) to stop the recording

View live source using mediaURL

View Live source From SDK Player

Following actions requires REST Tester and sdkPlayer **(BarcoSDK/player/src/barco/BarcoPlayer.html)**. The URL for the sdkplayer will be **http://<webserver_ip>/<basePath>/BarcoSDK/player/src/barco/BarcoPlayer.html**

1. Enter http://<MSIP>/controller/Resource/Port?type=SrcPort&selectFormat=BASICURL&dstPortTitle=web-client_barco in the URL field.
2. destPortTitle should be the resource_username using in above field 2 & 4 e.g. web-client_barco
3. Select the HTTP Method as GET
4. Hit SUBMIT.
5. Should see a list of ports that user is allowed to see in the Response field.
6. Copy this mediaURL from response and paste in sample sdk player test box
7. Keep the BarcoRESTdiagnostics loggedin for LIVE to work.
8. Sample Response:
<Resource total="1" from="1"> <Port NID="default.mediastreamsrcportresourcelist.mssprl_c3dfb990-6b80-4408-8047-e54f05360496" streamType="RTP" portType="StreamSrc" streamingURL="http://10.51.51.18/mediaroom/index.m3u8?dirID=/
data1&amp;mediaID=46bfae8c-53aa-409c-8a7a-80cf7cc0099e&amp;apiKey=6319348664898058382&amp;timestamp=IN
hls="true" portId="1" presence="unavailable" mediaURL="http://10.51.51.65/clients/player?requestName=View&amp;SrcPort.portNID=default.mediastreamsrcportresourcelist.mssprl_aeba7237-d972-4372-a8f9-237ff55923a6&amp;userJID=mediaserver-localhost/
mediaserver_mediaserver&amp;timestamp=1397687388320&amp;apiKey=-7192030090146228263&amp;signature=63e3
facf-42b9-aec5-46d9fc1898b6" state="Unmanaged" title="axis"> <Groups> <Tag Name="Location" Value="Sunnyvale" /> <Tag Name="Department" Value="Engineering" /> </Groups> <Permissions add="false" modify="false" delete="false" streamto="false" viewfrom="true" read="true" /> </Port> </Resource>

Playback recorded file using mediaURL

Playback recorded file From SDK Player

Following actions requires REST Tester and sdkPlayer **(BarcoSDK/player/src/barco/BarcoPlayer.html)**. The URL for the sdkplayer will be **http://<webserver_ip>/<basePath>/BarcoSDK/player/src/barco/BarcoPlayer.html**

1. Enter http://<MSIP>/controller/Resource/File?selectFormat=BASICURL&dstPortTitle=web-client_barco in the URL field.
2. destPortTitle should be the resource_username using in above field 2 & 4 e.g. web-client_barco
3. Select the HTTP Method as GET
4. Hit SUBMIT.
5. Should see a list of files that user is allowed to see in the Response field.
6. Copy this mediaURL from response and paste in sample sdk player test box
7. Keep the BarcoRESTdiagnostics loggedin for LIVE to work.
8. Sample Response:
<Resource total="1" from="1"> <File NID="default.mediastreamfileresourcelist.msfrl_b42ffc9b-5660-4b08-a4fe-ecebba1108b3" streamingURL="http://10.51.51.18/mediaroom/index.m3u8?dirID=/data1&amp;apiKey=6319348664898058382&amp;mediaID=be8de800-2074-4036-81c9-b9d877f95f92&amp;timestamp=INFINITE&amp;signature=64eebc2385c5d8849fbf3b526577a730"

clipID="fea9d85b-6ce2-474a-a321-8abe789e650e" mediaID="be8de800-2074-4036-81c9-b9d877f95f92" size="181621315" readOnly="false" bypassAutoStorageRules="false" streamType="RTP" thumbnailURL="http://10.51.51.18/mediaportal/thumbnail?dirID=/data1&amp;timestamp=1392986268212&amp;apiKey=6319348664898058382&amp;signature=36d263de5f892444f7f4b7
b9d877f95f92" hls="true" dateCreated="1392894023752" fileType="MediaClip" dateModified="1392894023752" playlength="357327" mediaURL="http://10.51.51.18/clients/player?requestName=Playback&amp;SrcFile.fileNID=default.mediastreamfileresourcelist.msfrl_b42ffc9b-5660-4b08-a4fe-ecebba1108b3&amp;userJID=testuser-localhost/wc1_testuser&amp;timestamp=1392986268212&amp;apiKey=-6388481005313874188&amp;signature=b5839419473cf6
title="axis" state="Ready" sessionConfigNID="" mediaGroupID=""> <Permissions playback="true" export="false" modify="false" delete="false" read="true" /> <Groups> <Tag Name="Location" Value="Sunnyvale" /> </Groups> <MetaData>__EMPTY_STRING_</MetaData> <Download> <Media format="" state="NotReady" stale="false" /> </Download> </File> </Resource>

Download Media

Prepare File for download

We first subscribe the user for notifications and then prepare file for download.

1. Enter http://<MSIP>/controller/Resource/File?selectFormat=EXTENDEDURL&subscriptionID=1234 in the URL field.
2. Select the HTTP Method as GET
3. Hit SUBMIT.
4. Should see a list of files that user is allowed to see in the Response field.
5. Enter http://<MSIP>/controller/Resource/File/PrepareForDownload in the URL field.
6. Select the HTTP Method as POST
7. In the Request text area, add your request body. E.g. <PrepareFileForDownload byNID="default.mediastreamfileresourcelist.msfrl_959ecf85-022b-4146-8641-6ef1ad64f0f2" format="Standard" />
byNID is the NID for the RTP file you want to download **(get this from the GetFile Response). check the permissions element in the getFile response for verifying the permissions on the file. Should have export permissions for the file to allow download.** e.g. <Permissions delete="true" export="true" modify="false" delete="false" read="true" /> format is Proprietary (Barco Native format) or Standard (Zip with mp4)
8. Hit SUBMIT.
9. Will get a downloadURL in file response in the Notifications field, once the file is ready for download.
10. Use any browser to download media using downloadURL

Create Media using Cutpoints

Create Media using Cutpoints

1. Enter http://<MSIP>/controller/Resource/File/CreateMediaFromCutList in the URL field.
2. Select the HTTP Method as POST
3. In the Request text area, add your request body. E.g. <CreateMediaFromCutList byNID="default.mediastreamfileresourcelist.msfrl_959ecf85-022b-4146-8641-6ef1ad64f0f2" cutList="0:30000,40000:60000,70000:120000" title="newFileTitle"> <MetaData> <Tag Name=""> <content>Value</content> </Tag> </MetaData> <Description /> </CreateMediaFromCutList>
byNID is the NID for the RTP file you want to download **(get this from the GetFile Response)**. cutList is a comma separated list of start:end offsets in the media in milliseconds. E.g. 0:30000 is 30 seconds of media from start offset. Title is the name for the new media being created.
4. Hit SUBMIT.

5. Will get a 201 response for request accepted.
8. Will get a file notification in the Notifications field, when the created media is ready

API

Following is a list of pointers to the API call documentation over REST. Note that the request bodies mentioned as sample are not minimal. Some of the elements/attributes could be optional.

Table B.1.

ServiceLogin	Refer Example 1.1 Service Login Request
GetPort - Sources	Refer Example 2.13 Get Port Request
GetPort - Destination	Refer Example 2.13 Get Port Request
View	Refer Example 4.27 Basic View Request by Title
StopSession	Refer Example 4.25 Stop Session Request
Record	Refer Example 4.50 Basic Record Request
GetFile - MediaClips	Refer Example 2.49 Get File Request
Playback	Refer Example 4.40 Basic Playback Request

NOTE: BarcoRESTdiagnotics tool is qualified for Google Chrome and IE-9. When loading the page in IE-9 please ensure you are NOT running the page in compatibility view mode. The BarcoRESTdiagnostics.html is a pure JavaScript implementation using XMLHttpRequests and does not use a JQuery or any other 3rd party library. To debug or view the communication logs use the browsers development tools.

Release Notes

Summary of changes from 4.0 to 4.1

Added Requests

```
Prepare File: prepare a file to download for Proprietary/Standard/Both formats
Create Media From Cutlist: Creates Media from existing media using start and end
Trim File with Cutlist: Trim out sections of an existing media file using start
Clear Download Cache: Remove cache memory , which is occupied while downloading
Download Media: Download the provided media file to the media store, you can use
Inject Media: Uploads the provided media file to the media store
```

Updated Requests

```
GetFile:Added new selectFormat BASICCONNECTIONINFO which will return connection
```

```
GetDir: Added new selectFormat UPLOADURL which will return upload URL. Which is
```

Added Notifications

```
FileUpdate: Added file Add/Update/Delete notifications
FileConnectionUpdate: Added connection Add/Delete notification
FileConnectionStateUpdate: Added file connection state update notification
```

Summary of changes from 4.1 to 4.2

Added Requests

```
Get DisplayProfile: The list of DisplayProfile objects that the Client has acces
```

Updated Requests

```
GetFile:Added new query parameter whereDownloadState
GetDir: Added new query parameter dstPortTitle
DeleteFile: Added new query parameter moveToRecycleBin
GetDevice: Updated to allow filter for whereType="Proxy-Device/WebClient-Device"
```

```
Updated Accept header to "application/xml" for Session and Bookmark API's (Start
```

```
Create Media From Cutlist: Cutlist values are validated against certain rules in
```

```
Inject Media: Added support to enable HLS streaming
Play: Slowmotion is not supported
```