

XMPP API

XMPP API

Table of Contents

Glossary	1
Introduction	iii
1. Service Overview	4
Version	4
The Central Server Entity	4
The Client Entities	4
The Service Messaging	5
2. Service Message Overview	6
Service Request	6
Service Response	6
Service Error	6
Service Notification	7
3. Generic Service Client API	8
Startup Operations	8
XMPP Login	8
ServiceLogin	8
Get Service Client	9
4. Session Object API	15
Parameter Listing	15
Resource Info Operations	15
Get Port Tags	15
Get Port	17
Get Profile	21
Get Dir Tags	22
Get Dir	23
Get File Tags	26
Get File	28
Prepare File For Download	34
Clear Download Cache	35
Create Media From Cutlist	36
Trim File with Cutlist	38
Inject Media	39
Download Media	41
Get User Tags	41
Get User	42
Get Tags	43
Get DisplayProfile	45
Session Info Operations	46
Get Session	46
Get Session Config	52
Get Invite	54
5. Session Setup API	55
Session Requests	55
Parameter Listing	55
StartSession	56
StopSession	66
Publish	66
Session Connection Operations	66
View	66
Playback	74
Record	77

PostSnapshot	82
ViewSnapshot	82
ViewThumbnail	82
SwitchSrc	83
SwitchDst	83
SwitchProfile	83
DeleteConn	84
6. Session Participant API	85
Session Participant Operations	85
Invite	85
Kick	85
Join	85
Leave	86
7. Session Controller API	87
Scope for Session Control Operations	87
Session Scope	87
ConnGroup Scope	87
Connection Scope	87
Session MediaStream Operations	87
Start	88
Stop	88
Session PVR Operations	88
Time for PVR Operations	88
TimeSpec for PVR Operations	88
Action for PVR Operations	91
Connection Scope	95
Bookmark Operations	96
Parameter Listing	96
NewBookmark	96
AddBookmark	97
UpdateBookmark	98
DeleteBookmark	98
GetBookmark	99
8. Session Notifications	101
EntityPresence Notification	101
ConnState Notification	101
SessionState Notification	103
Timeline Notification	103
9. Windowing API	105
Parameter Listing	105
Get Operations	105
Get Port Display Configuration Request	105
Get Port Display Status Request	105
Set Operations	106
Set Port Display Configuration Request	106
Set port command Parameter Listing	106
Set Port Window Properties	107
10. Resource Update Notifications	109
Resource ConnectionUpdate Notification	109
Resource ConnectionStateUpdate Notification	111
Port	111
Port Update	111
Port Connection Update	112
Port Connection State Update	112

Port Display Config Update	113
Port Display Status Update	113
File	113
File Update	113
File Connection Update	115
File Connection State Update	115
11. External Authorization	117
AuthorizationRequired Message	117
Authorize Request	117
12. Admin API	118
SetFileMetaData	118
DeleteFileRequest	119
Set User Password	121
Device Object	121
AddDevice	121
SetDeviceInfo	122
Get Device	123
DeleteDevice	129
Device Object Response XML	130
Port Object	132
Parameter Listing Of Set API	133
SetPortInfo	133
SetPortStreamURL	133
SetPortObserverInfo	134
SetPortPVRInfo	134
Port Object Response XML	135
13. Report API	136
Audit Event	136
14. Frequently Asked Questions	137
A. Data Model	138
Resource Objects	138
Resource Attributes	138
Object Info	139
Session Objects	140
DisplayProfileInfo	141
RX Display Config/Status Parameter Listing	142
RX DisplayConfig	143
RX DisplayStatus	143
Remote Operated Client Display Config/Status Parameter Listing	143
Remote Operated Client DisplayConfig	145
Remote Operated Client DisplayStatus	146
XMPP API Tutorial	147
Prerequisites	147
Tutorial	148
Release Notes	149
Summary of changes from 3.10 to 3.12	149
Summary of changes from 3.12 to 3.14	150
Summary of changes from 3.14 to 3.16	150
Summary of changes from 3.16 to 3.18	150
Summary of changes from 4.0 to 4.1	151
Summary of changes from 4.1 to 4.2	151

List of Examples

2.1. Request	6
2.2. Response	6
2.3. MediaRoom Service returns Error	7
2.4. Notification	7
3.1. Service Login Request	8
3.2. Service Login Response	9
3.3. Get Service Client Request	9
3.4. Get Service Client Response - BASIC Format	10
3.5. Get Service Client Response - EXTENDED Format	11
3.6. Get Service Client Response - EXTENDED Format [Continue]	12
3.7. Get Service Client Response - EXTENDED Format[Continue]	13
3.8. Get Service Client Response - EXTENDED Format[Continue]	14
4.1. Get Port Tags Request	16
4.2. Get Port Tags Response - NameList Format	16
4.3. Get Port Tags Response - NameValueList Format	16
4.4. Get Port Tags Response - NameValueTree Format	16
4.5. Get PortList Request	17
4.6. Get Port Response - BASIC Format	17
4.7. Get Port Response - EXTENDED Format	18
4.8. Get Port Response - BASICCONNECTIONINFO Format	19
4.9. Get Port Response - EXTENDEDCONNECTIONINFO Format	20
4.10. Get Port Response - BASICURL Format	20
4.11. Get Port Response - EXTENDEDURL Format	21
4.12. Get Profile Request	21
4.13. Get Profile Response - BASIC Format	21
4.14. Get Profile Response - EXTENDED Format	22
4.15. Get Dir Tags Request	22
4.16. Get Dir Tags Response - NameList Format	22
4.17. Get Dir Tags Response - NameValueList Format	23
4.18. Get Dir Tags Response - NameValueTree Format	23
4.19. Get Dir Request	24
4.20. Get Dir Response - BASIC Format	25
4.21. Get Dir Response - EXTENDED Format	25
4.22. Get Dir Response - INJECTURL Format	26
4.23. Get File Tags Request	27
4.24. Get File Tags Response - NameList Format	27
4.25. Get File Tags Response - NameValueList Format	27
4.26. Get File Tags Response - NameValueTree Format	27
4.27. Get File Request	29
4.28. Sample sortByTagName="Location"	30
4.29. Sample sortByMetaData="TestAPI"	30
4.30. Get File Response - BASIC Format	31
4.31. Get File Response - EXTENDED Format	32
4.32. Get File Response - BASICURL Format	32
4.33. Get File Response - EXTENDEDURL Format	33
4.34. Get File Response - BASICCONNECTIONINFO Format	34
4.35. Prepare File For Download Request	35
4.36. Prepare File for Download Response	35
4.37. Clear Download Cache	36
4.38. Clear Download Cache Response	36
4.39. Create Media From Cutlist Request	37

4.40. Create Media From Cutlist Response	37
4.41. Trim File with Cutlist	38
4.42. Trim File with Cutlist Response	39
4.43. Inject Media	40
4.44. Download Media	41
4.45. Get User Tags Request	41
4.46. Get User Tags Response - NameList Format	42
4.47. Get User Tags Response - NameValueList Format	42
4.48. Get User Tags Response - NameValueTree Format	42
4.49. Get User Request	42
4.50. Get User Response - BASIC Format	43
4.51. Get User Response - EXTENDED Format	43
4.52. Get Tags Request	44
4.53. Get Tags Response - NameList Format	44
4.54. Get Tags Response - NameValueList Format	44
4.55. Get Tags Response - NameValueTree Format	45
4.56. Get DisplayProfile Request	45
4.57. Get DisplayProfile Response - BASIC Format	45
4.58. Get DisplayProfile Response - EXTENDED Format	46
4.59. Get Session Request	46
4.60. Get Session Response - BASIC Format[Not Implemented]	47
4.61. Get Session Response - BASIC Format[Currently Implemented]	47
4.62. Get Session Response - EXTENDED Format[Not Implemented]	48
4.63. Get Session Response - EXTENDED Format[Not Implemented] (Continued)	49
4.64. Get Session Response - EXTENDED Format[Not Implemented] (Continued)	50
4.65. Get Session Response - EXTENDED Format[Currently Implemented]	51
4.66. Get Session Response - EXTENDED Format[Currently Implemented] (Continued)	52
4.67. Get Session Response - EXTENDED Format[Currently Implemented] (Continued)	52
4.68. Get Session Config Request	53
4.69. Get Session Config Response - BASIC Format	53
4.70. Get Session Config Response - EXTENDED Format	54
4.71. Get Invite Request	54
4.72. Get Invite Response - BASIC Format	54
5.1. Basic Start Recording Session Request	57
5.2. Basic Start Recording Session Request (Continued)	58
5.3. Basic Start Recording Session Request (Continued)	59
5.4. Advance Start Recording Session Request [Not Implemented]	60
5.5. Advance Start Recording Session Request [Not Implemented] (Continued)	61
5.6. Advance Start Recording Session Request [Not Implemented] (Continued)	62
5.7. Basic Start Playback Session Request	63
5.8. Advance Start Playback Session Request [Not Implemented]	64
5.9. Advance Start Playback Session Request [Not Implemented] (Continued)	65
5.10. Advance Start Playback Session Request [Not Implemented] (Continued)	65
5.11. StopSession Request	66
5.12. Publish Request	66
5.13. Basic View Request by Title	67
5.14. Basic View Request by NID	67
5.15. View Request with Profile Title	67
5.16. View Request with Profile NID	68
5.17. View Request with PVR	68
5.18. View Request to add a connection from an existing Src to a new Dst	68
5.19. View Request to add a connection from an existing Src to a new Dst with profileid	68
5.20. View Request to add a connection from an existing Src to existing Dst	69
5.21. View Request Response	70

5.22. View Request Response (Continued)	71
5.23. View Request Response [Not Implemented]	72
5.24. View Request Response [Not Implemented] (Continued)	73
5.25. View Request Response [Not Implemented] (Continued)	74
5.26. Basic Playback Request by Title	75
5.27. Basic Playback Request by NID	75
5.28. Basic Playback Request with file name/date spec	75
5.29. Advanced Playback Request including grpFileId and StorePort spec	76
5.30. Basic Playback Request by group file NID	76
5.31. Basic Playback Request by srcEntityId	76
5.32. Basic Playback Request by srcEntityId and dstEntityId	77
5.33. Playback Request Response	77
5.34. Basic Record Request	78
5.35. Basic Record Request with file name/dir spec	78
5.36. Record from an existing Src	78
5.37. Append to Recording Based on Title	78
5.38. Append to Recording Based on MetaData	79
5.39. Record Request with Profile NID	79
5.40. Record Request with Profile Title	79
5.41. Record from an existing Src with profile id	80
5.42. Advanced Record Request including grpFileId and StorePort spec	80
5.43. Record Request Response	81
5.44. Record Request Response [continued]	81
5.45. SwitchSrc Request	83
5.46. SwitchSrc Response	83
5.47. SwitchDst Request	83
5.48. SwitchProfile Request	83
5.49. DeleteConn Request	84
5.50. DeleteConn Response	84
6.1. Simple Invite by userId or jid	85
6.2. Invite by userId(who) and/or deviceId(from where)	85
6.3. Kick	85
6.4. Join without invite Request	86
6.5. Join without invite Response	86
6.6. Join without invite Response - MUC Room Invite Message	86
6.7. Join with invite	86
6.8. Leave	86
7.1. Request with Session scope	87
7.2. Request with ConnGroup scope	87
7.3. Request with Connection scope	87
7.4. Start Request	88
7.5. Stop Request	88
7.6. Resume Request w/ nowOffset [Deprecated Replace With Play Request w/ nowOffset]	89
7.7. Resume Request w/ startOffset [Deprecated Replace With Play Request w/ startOffset]	89
7.8. Resume Request w/ liveOffset [Deprecated Replace With Play Request w/ liveOffset]	89
7.9. Resume Request w/ mediaOffset [Deprecated Replace With Play Request w/ mediaOffset]	90
7.10. Resume Request w/ mediaTime [Deprecated Replace With Play Request w/ mediaTime]	90
7.11. Resume Request w/ Bookmark [Deprecated Replace With Play Request w/ Bookmark]	90
7.12. Play Request w/ nowOffset	90
7.13. Play Request w/ startOffset	90
7.14. Play Request w/ liveOffset	90
7.15. Play Request w/ mediaOffset	91
7.16. Play Request w/ mediaTime	91
7.17. Play Request w/ Bookmark	91

7.18. Pause Request	91
7.19. Pause Response	91
7.20. Resume Request	92
7.21. Resume Response	92
7.22. Play Request	92
7.23. Play Response	92
7.24. PauseRecord Request	93
7.25. PauseRecord Response	93
7.26. ResumeRecord Request	93
7.27. ResumeRecord Response	93
7.28. Catchup Request	93
7.29. Catchup Response	94
7.30. SkipBack Request	94
7.31. SkipBack Response	94
7.32. SkipFwd Request	94
7.33. SkipFwd Response	95
7.34. Resume Request with connection scope	95
7.35. Resume Response with connection scope	95
7.36. Resume Error Response with connection scope	95
7.37. NewBookmark Request	96
7.38. NewBookmark Response	97
7.39. AddBookmark Request	97
7.40. AddBookmark Response	97
7.41. UpdateBookmark Request	98
7.42. UpdateBookmark Response	98
7.43. DeleteBookmark Request	98
7.44. DeleteBookmark Response	99
7.45. GetBookmark Request	99
7.46. GetBookmark Response - BASIC Format	99
7.47. GetBookmark Response - EXTENDED Format	100
8.1. EntityPresence Notification	101
8.2. ConnState Notification: CONNECTED	102
8.3. ConnState Notification: Error	102
8.4. SessionState Notification	103
8.5. Timeline Notification	104
9.1. Get Port Display Configuration Request	105
9.2. Get Port Display Configuration Response	105
9.3. Get Port Display Status Request	105
9.4. Get Port Display Status Response	106
9.5. Set Port Display Configuration Request	106
9.6. Set Port Display Configuration Response	106
9.7. Set Port Window Properties Request	107
9.8. Set Port Window Properties Response	108
10.1. AuthConectionInfo element in AuthorizationRequired Notification	110
10.2. PortConectionInfo element in PortConnectionUpdate Notification - BASICCONNECTIONINFO format	110
10.3. PortConectionInfo element in PortConnectionUpdate Notification - EXTENDEDCONNECTIONINFO format	110
10.4. FileConnectionInfo element in FileConnectionUpdate Notification - BASICCONNECTIONINFO format	111
10.5. ConectionState element in ConnectionStateUpdate Notification	111
10.6. Port Update Notification BASIC	111
10.7. Port Update Notification EXTENDED	112
10.8. Port Connection Update Notification	112

10.9. Port Connection State Update Notification	112
10.10. Port DisplayConfigUpdate Notification	113
10.11. Port DisplayStatusUpdate Notification	113
10.12. File Update Notification BASIC	114
10.13. File Update Notification EXTENDED	114
10.14. File Update Notification EXTENDEDURL	115
10.15. File Connection Update Notification	115
10.16. File Connection State Update Notification	116
11.1. AuthorizationRequired Message	117
11.2. Authorize Request	117
11.3. Authorize Response	117
12.1. Set File MetaData Request With NID	118
12.2. Set File MetaData Request With MetaData	118
12.3. Set File MetaData Response	119
12.4. DeleteFile Request by NID	119
12.5. DeleteFile by clip file nid or image file nid Response	120
12.6. DeleteFile by Group File nid Response	120
12.7. Set User Password Request	121
12.8. Set User Password Response	121
12.9. Add Device Request	122
12.10. Add Device Response	122
12.11. Set Device Info Request	123
12.12. Set Device Info Response	123
12.13. Get Device Request	124
12.14. Get Device Response - BASIC Format	125
12.15. Get Device Response - EXTENDED Format	126
12.16. Get Device Response - EXTENDED Format [Continued]	127
12.17. Get Device Response - EXTENDED Format [Continued]	128
12.18. Get Device Response - RAW Format [Continued]	129
12.19. Delete Device Request	129
12.20. Delete Device Response	129
12.21. Device Object Response XML	130
12.22. Device Object responses XML [Continued]	131
12.23. Device Object responses XML [Continued]	132
12.24. Set Port Info Request	133
12.25. Set Port Info Response	133
12.26. Set Port StreamURL Request	133
12.27. Set Port StreamURL Response	134
12.28. Set Port Observer Info Request	134
12.29. Set Port Observer Info Response	134
12.30. Set Port PVR Info Request	135
12.31. Set Port PVR Info Response	135
12.32. Port Object Response XML	135
13.1. Audit Event Request	136
13.2. Audit Event Response	136

Glossary

XMPP Login	Establishing a "Client-Server" XMPP "Session" as described in RFC 3921 Section 3 (with RFC3920 Sections 5 (TLS) and 6 (SASL)).
XMPP Message	The XMPP "message" container as described in RFC 3920.
XMPP Server/Client	XMPP Entities as described in RFC 3920 Section 2.
XMPP MUC	XMPP Multi-User Chat as described in XEP-0045.
Application Server	An XMPP Service that provides the Server side of the IPVS Collaboration Service API.
Service Client	An XMPP Agent that implements the Client side of the IPVS Collaboration Service API. Depending on the range of functionality required, a subset of all the Service Use Cases will be implemented by each Service Client.
Media Room	The runtime Entity that manages/represents a Collaboration Session. Logically, this is an MUC "ChatRoom" with extensions for IPVS Collaboration "MediaStream" functionality.
EPOCH	Refers to the Unix epoch which corresponds to the time 00:00:00 UTC on January 1, 1970.
Time	All Time values in the system are expressed as a number of milliseconds since EPOCH. Time at EPOCH = 0.
Offset	A "delta" relative to some Time value.
WC	The "WallClock" Time at any moment. Every entity in the system potentially has it's own idea of WC. Entities that have their clocks synchronised (via NTP) are considered to have an identical WC at any moment.
TC	TimeCode: The "Time" embedded in the Media.
Asset	Asset: A representation of the physical device in the database. There is 1 per device
Resource	Resource: A representation of the media/port of a device in the database e.g. encoder port
Device	Device: A physical device with media resources.
DisplayProfile	DisplayProfile: A template of a canvas(see canvas below)
DisplayConfig	DisplayConfig: The display profile associated with a device, note this is a copy not a reference of the DisplayProfile
Canvas	Canvas: A canvas can have multiple layouts with one of them being active
Layout	Layout: An xml representation of a set of windows and their geometry i.e x,y coordinates, width,height,z etc
Window	Window: The smallest unit within a layout. A window has x,y coordinates, width,height and other attributes

PortWindow

PortWindow: The window in a DisplayConfig associated with a device will be realized as a window painted by the renderer for a physical/logical port. So calls to set the set of properties on this window are prefixed by SetPortWindowProperty(...)

Introduction

This document contains parts of the IPVS Collaboration Protocol specifications pertaining setup and control of Collaboration Sessions.

Chapter 1. Service Overview

This chapter provides the version information and a brief description of the "Entities" involved in the IPVS Collaboration Service.

Version

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

The Central Server Entity

This is called the Application Server. This has the following main functional areas:

- A centralized Database of Service Objects related to:

Service Assets: Devices, Media Ports
Service Policy: Policy Tags, Filters and Rules
Service Config: Connection Rules, Profiles, Routing Tables
- A centralized Media Archive:

Media Directories
Media Files (Recordings)
Playlists
- A centralized run-time Service Co-ordinator that manages:

Session Setup Requests (w/ Policy checks)
Service Client Entity Invitations (with Role Information based on the Setup requests)
Session Runtime Objects: Session MediaRooms, Entity Presence, Connection State

The Client Entities

There are of different types of client entities based on the Collaboration Role they are designed to perform.

- Media Agent Entities:

Media Source (eg: a V2D Tx encoder)
Media Destinations (eg: a V2D Rx Decoder)
Media Relay (eg: a DMS)
Media Store (eg: a DMS)
- Session Agent Entities:

Session Participant
Session Controller

Multiple of these Entities are typically co-located within one Service endpoint. eg: A typical PC Client would have a Media Destination (the Player) and a Session Controller (the Pause/Play Buttons).

The Service Messaging

The service messaging between the Server and the Clients (and among the Clients) is realized as a set of XML messages over an XMPP message bus. These messages constitute the IPVS Collaboration Service API. Entities will need to implement the relevant sub-set of this API based on their intended Service Role.

Chapter 2. Service Message Overview

All API calls are transacted as extensions of standard "XMPP IQ / Message" packets.

All IPVS message payloads will use the namespace "com.ipvs.session"

Service Request

Example 2.1. Request

```
<iq from='roomuser-default@localhost/userpc'
  id='zzz-123'
  to='managementserver.localhost'
  type='set/get'>
  <query xmlns='com.ipvs.session' serviceVer=''>
    <!-- Request data goes here -->
  </query>
</iq>
```

The default value for type='get' unless specifically mentioned in the request

Service Response

Example 2.2. Response

```
<iq from='roomuser-default@localhost/userpc'
  id='zzz-123'
  to='managementserver.localhost'
  type='result'>
  <query xmlns='com.ipvs.session' serviceVer=''>
    <!-- Response data goes here -->
  </query>
</iq>
```

Service Error

If there is an error, the service MUST return an error to the user:

Example 2.3. MediaRoom Service returns Error

```
<iq from='roomuser-default@localhost/userpc'
  id='zzz-123'
  to='managementserver.localhost'
  type='error'>
  <query xmlns='com.ipvs.session' serviceVer=''>
    <!-- Request data goes here -->
  </query>
  <error code='400' type='modify'>
    <!-- Error data goes here -->
    <x xmlns="com.ipvs.session">
      <ipvserror code="2033">
        <description> Pause Action Not Valid For Current State</description>
      </ipvserror>
    </x>
  </error>
</iq>
```

Service Notification

Async events are sent as payloads within XMPP message packets

Example 2.4. Notification

```
<message from='managementserver.localhost'
  id='reg2'
  to='roomuser-default@localhost/userpc'
  type='normal'>
  <x xmlns='com.ipvs.session'>
    <!-- Event data goes here -->
  </x>
</message>
```

Chapter 3. Generic Service Client API

The following set of Operations will typically be performed by most Service Clients "outside" of a Session context

Startup Operations

XMPP Login

First the Client does a standard XMPP Login. The credentials needed for this need to be provided to the Client by the Administrator by some out-of-band means.

- Username@DomainName.com (Who)
- Password
- XMPPResourceID (From Where)

ServiceLogin

After the base XMPP login the Client does a "Service Login" with the IPVS Collaboration Application Server.

- During this process the client's contact Info and state are updated in the Server DB
- Client service version number and licensing is also checked at this time
- The Client version number can be obtained from the Version section of the API document corresponding to the client implementation
- If the client is to be dynamically created, an optional clientTemplateTitle has to be specified. The server will create a client using the data available for the clientTemplate. The clientTitle will be based on the XMPPResourceID specified at the time of login
- The ready attribute if set to "true" indicates that all resources on the device are ready. The server will get the list of resources associated with the device and mark them as ready as well

This will change the status to not ready indicating that it has logged in but cannot participate in a session as a source or destination.

Example 3.1. Service Login Request

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

If all is well, the response will contain the Client's private data(preferences) that has been configured/stored on the server.

- The current date/time is also returned

Example 3.2. Service Login Response

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

Get Service Client

This will return the logged in client information

Example 3.3. Get Service Client Request

```
<GetServiceClient whereContactJID='' selectFormat='BASIC/EXTENDED' />
```

- *whereContactJID=""* [Optional]

Example 3.4. 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 3.5. 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 3.6. 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 3.7. Get Service Client Response - EXTENDED Format[Continue]

```
<AVCodec>
<Port id=''>
  <Displayconfig type='IPVSCClientDisplayConfig'>
    <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 3.8. 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>
```

Chapter 4. Session Object API

The following set of Operations will typically be performed by most Service Clients "outside" of a Session context

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='Location' and Value='Sunnyvale' " [Optional]
- *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]
- *whereTextSubStr*=" " [Optional]

Resource Info Operations

After logging in with the App Service the Client can start to make calls to discover info about Src / Dst / Media Resources. See Parameter Listing for common parameter explanation.

Get Port Tags

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

Example 4.1. Get Port Tags Request

```
<GetPortTags countToFetch="10" startFrom="1"
              sortBy=" " sortOrder="ascending"
              whereNameTextSubString=" " whereValueTextSubString=" "
              wherePermissions=" " selectFormat=" " />
```

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

Example 4.2. Get Port Tags Response - NameList Format

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

Example 4.3. Get Port Tags Response - NameValueList Format

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

Example 4.4. Get Port Tags Response - NameValueTree Format

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

Get Port

This is the list of Src/Dst Ports that the Client has access to. Refer to the Appendix for the format of the Port object

Example 4.5. Get PortList Request

```
<GetPort type=""
  startFrom="1" countToFetch="100" sortBy="" sortOrder=""
  selectFormat="BASIC/EXTENDED/EXTENDEDCONNECTIONINFO/BASICCONNECTIONINFO/BASICU
  whereNID=""
  whereTextSubStr=""
  whereDateRange=""
  whereTagExpr="" />
  whereStreamType=""
  wherePermissions=""
  subscriptionID=""
  dstPortTitle="" />
```

- *sortBy*="title/state"
- *sortBy*="NID/portType/presence" [Not Supported]
- *type*="SrcPort/DstPort" [Required]
- *whereDateRange*
- *whereStreamType*="V2D/RTP/MPEGTS/UDP"
- *wherePermissions*="ADD,MODIFY,DELETE,VIEWFROM,STREAMTO"
- *dstPortTitle* [Optional]

Example 4.6. Get Port Response - BASIC Format

```
<GetPort 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>
</GetPort>
```

Example 4.7. 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>
```

Example 4.8. Get Port Response - BASICCONNECTIONINFO Format

```
<GetPort type="SrcPort/DstPort/RelayPort/StorePort" from="0"
total="0">
  <Port NID='' title='' streamType='' presence='' state='' portType='' portId='' t
    <Asset NID='' title='' deviceType='' deviceNID='' />
    <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='' window
        </PortConnectionInfo>
      </PortSessionInfo>
    </ConnectionStatus>
    <InstanceList>
      <Instance streamURL='' />
    </InstanceList>
  </Port>
</GetPort>
```

Example 4.9. Get Port Response - EXTENDEDCONNECTIONINFO Format

```
<GetPort type="SrcPort/DstPort/RelayPort/StorePort" from="0"
total="0">
  <Port NID='' title='' streamType='' presence='' state='' portType='' portId='' t
  <Asset NID='' title='' deviceType='' deviceNID='' />
  <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=''>

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

Example 4.10. Get Port Response - BASICURL Format

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

Example 4.11. Get Port Response - EXTENDEDURL Format

```
<GetPort 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='' deviceNID='' />
    <MaxInstanceCount />
    <FeatureList>
      <Feature name="" />
    </FeatureList>
  </Port>
</GetPort>
```

Get Profile

[Not Implemented] This is the list of Profiles that the Client has access to. Refer to the Appendix for the format of the Profile object

Example 4.12. Get Profile Request

```
<GetProfile startFrom="1" countToFetch="100" sortBy=""
  sortOrder=""
  selectFormat="BASIC/EXTENDED"
  whereNID=""
  whereTextSubStr=""
  whereStreamType=""
  whereTagExpr="" />
```

- *sortBy="NID/title/streamType/bandwidth/multicast/avOption/enableKbm"*

Example 4.13. Get Profile Response - BASIC Format

```
<GetProfile from="0" total="50">
  <Profile NID="" title="" streamType="" bandwidth=""
    multicast="" avOption="" enableKbm="" />
</GetProfile>
```

Example 4.14. Get Profile Response - EXTENDED Format

```
<GetProfile from="0" total="50">
  <Profile NID="" title="" streamType="">
    <description />
    <Groups>
      <Tag name="" value="" />
    </Groups>
    <ProfileInfo/>
  </Profile>
</GetProfile>
```

Get Dir Tags

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

Example 4.15. Get Dir Tags Request

```
<GetDirTags countToFetch="10" startFrom="1"
  sortBy="" sortOrder="ascending"
  whereNameTextSubString="" whereValueTextSubString=""
  wherePermissions="" selectFormat="NameList/NameValueList/NameValueTree" />
```

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

Example 4.16. Get Dir Tags Response - NameList Format

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


Example 4.17. Get Dir Tags Response - NameValueList Format

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

Example 4.18. Get Dir Tags Response - NameValueTree Format

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

Get Dir

This is the list of Dirs that the Client has access to. Refer to the Appendix for the format of the Dir object

Example 4.19. Get Dir Request

```
<GetDir countToFetch=" "  
  selectFormat=" "  
  sortBy=" "  
  sortOrder=" "  
  startFrom=" "  
  whereDateRange=" "  
  whereNID=" "  
  whereTagExpr=" "  
  whereTextSubStr=" "  
  whereParentDirNID=" "  
  wherePermissions=' '  
  whereFilePermissions=' '  
  subscriptionID=' '  
  subscriptionScope=" "  
  expireTime=' '  
  dstPortTitle=' ' />
```

- *sortBy*="title/state/size/sizeUsed/dateCreated/dateModified"
- *sortBy*="NID" [Not Supported]
- *whereNID*="DirNID"
- *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]
- *dstPortTitle* [Optional]

Example 4.20. Get Dir Response - BASIC Format

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

Example 4.21. Get Dir Response - EXTENDED Format

```
<GetDir from="0" total="50">
<Dir NID='' title='' dirType='' presence='' state='' size=''
  sizeUsed='' dateCreated='' dateModified=''
  readOnly=''>
  <description />
  <Groups>
    <Tag name='' value='' />
  </Groups>
  <StorageConfig mode=''>
    <DirSpanning diskFullThreshold=''>
      <PortInfo>
        <Port NID='' title='' deviceElementId='' priority='' presence='' readOnly='' />
      </PortInfo>
    </DirSpanning>
    <CapacityManagement schedule='' state='' lastRunDate='' lastRunTime='' lastRunError='' />
    <ThresholdRule enable='' diskCapacityReached='' reduceDiskCapacity='' type='' />
    <ILMRule enable='' timestamp='' type='' nasDriveURL='' />
  </StorageConfig>
  <SubDirInfo>
    <Dir NID="" title="" dirType="RECYCLEBIN" />
    <Dir NID="" title="" dirType="PVR" />
  </SubDirInfo>
  <Permissions modify='' delete='' read='' import='' playback='' record='' />
</Dir>
</GetDir>
```

Example 4.22. Get Dir Response - INJECTURL Format

```
<GetDir from="0" total="0">
<Dir NID='' title='' dirType='' state='' size='' sizeUsed='' dateCreated='' dateModified=''
<description />
<Groups>
  <Tag Name="" Value="" />
</Groups>
<StorageConfig mode=''>
  <DirSpanning diskFullThreshold=''>
    <PortInfo>
      <Port NID='' title='' deviceElementId='' priority='' presence='' readOnly='' />
    </PortInfo>
  </DirSpanning>
  <CapacityManagement schedule='' state='' lastRunDate='' lastRunTime='' lastRunStatus='' />
    <ThresholdRule enable='' diskCapacityReached='' reduceDiskCapacity='' type=''/>
    <ILMRule enable='' timestamp='' type='' nasDriveURL=''/>
  </CapacityManagement>
</StorageConfig>
<SubDirInfo>
  <Dir NID="" title="" dirType="RECYCLEBIN" />
  <Dir NID="" title="" dirType="PVR" />
</SubDirInfo>
  <Permissions modify='' delete='' read='' import='' playback='' record='' />
</Dir>
</GetDir>
```

Get File Tags

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

Example 4.23. Get File Tags Request

```
<GetFileTags countToFetch="0" startFrom="1"
    sortBy=" " sortOrder="ascending"
    whereNameTextSubString=" " whereValueTextSubString=" "
    wherePermissions=" " selectFormat=" " />
```

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

Example 4.24. Get File Tags Response - NameList Format

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

Example 4.25. Get File Tags Response - NameValueList Format

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

Example 4.26. Get File Tags Response - NameValueTree Format

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

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. Refer to the Appendix for the format of the File object

Example 4.27. Get File Request

```
<GetFile countToFetch=" "
    selectFormat="BASIC/EXTENDED/BASICURL/EXTENDEDURL"
sortBy=" "
sortByMetaData=' '
sortByTagName=' '
sortOrder="ascending"
startFrom="1"
whereType=" "
whereMediaGroupID=" "
whereDateRange=" "
whereNID=" "
whereTagExpr=" "
whereTextSubStr=" "
searchTextSubStrIn=TITLE,DESCRIPTION,METADATA,BOOKMARK
whereState=" "
whereMetaDataExpr=" "
whereParentDirNID=" "
whereSessionConfigNID = " "
wherePermissions=' '
subscriptionID=' '
subscriptionScope=" "
expireTime=' '
dstPortTitle=' '
whereDownloadState=' ' />
```

- *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='TestAPI' and Value='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*]
- *selectFormat="BASIC/EXTENDED/BASICURL/EXTENDEDURL/BASICCONNECTIONINFO"* [Optional. Default=BASIC]
- *dstPortTitle* [Optional]
- *whereDownloadState="Ready/Busy/NotReady"* [Optional (filter for cached files)]

Example 4.28. Sample sortByTagName="Location"

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

Example 4.29. Sample sortByMetaData="TestAPI"

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


Example 4.30. Get File Response - BASIC Format

```
<GetFile from="0" total="0">
  <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>
  <File NID='' title='' fileType='MediaGroup' streamType='' size=''
    playlength='' presence='' 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>
  <FileList>
    <File NID='' title='' fileType='MediaClip' streamType='' size=''
      playlength='' presence='' 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>
  </FileList>
</File>
</GetFile>
```

Example 4.31. Get File Response - EXTENDED Format

```
<GetFile from="0" total="0">
  <File NID='' title='' fileType='' streamType='' size=''
    playlength='' presence='' state='' dateCreated='' dateModified=''
    readOnly='' mediaGroupId='' ffTracks='' bypassAutoStorageRules='true/false'>
    <description />
    <Groups>
      <Tag Name="" Value="" />
    </Groups>
    <Permissions modify='' delete='' read='' export='' playback='' />
    <MetaData>
      <Tag Name="">Value</Tag>
    </MetaData>
    <Download>
      <Media format="Standard" state="NotReady" stale="" downloadURL="" />
      <Media format="Proprietary" state="NotReady" stale="" downloadURL="" />
    </Download>
    <Dir dirNID="" dirTitle="" />
  </File>
</GetFile>
```

Example 4.32. Get File Response - BASICURL Format

```
<GetFile from="0" total="0">
  <File NID='' title='' fileType='MediaClip' streamType='' size=''
    playlength='' state='' dateCreated='' dateModified=''
    readOnly='' mediaGroupId='' ffTracks='' mediaURL='' thumbnailURL='' bypassAuto
    <Groups>
      <Tag Name='' Value='' />
    </Groups>
    <Permissions modify='' delete='' read='' export='' playback='' />
    <MetaData>
      <Tag Name=''>Value</Tag>
    </MetaData>
  </File>
</GetFile>
```

Example 4.33. Get File Response - EXTENDEDURL Format

```
<GetFile from="0" total="0">
  <File NID='' title='' fileType='' streamType='' size=''
    playlength='' presence='' state='' dateCreated='' dateModified=''
    readOnly='' mediaGroupId='' ffTracks='' mediaURL='' thumbnailURL='' bypassAuto
    <description />
    <Groups>
      <Tag Name='' Value='' />
    </Groups>
    <Permissions modify='' delete='' read='' export='' playback='' />
    <MetaData>
      <Tag Name=''>Value</Tag>
    </MetaData>
    <Download>
      <Media format="Standard" state="NotReady" stale="" downloadURL="" />
      <Media format="Proprietary" state="NotReady" stale="" downloadURL=""
    </Download>
    <Dir dirNID='' dirTitle='' />
  </File>
</GetFile>
```

Example 4.34. Get File Response - BASICCONNECTIONINFO Format

```
<GetFile from="0" total="0">
  <File NID='' title='' fileType='' streamType='' size=''
    playlength='' presence='' state='' dateCreated='' dateModified=''
    readOnly='' mediaGroupId='' ffTracks='' mediaURL='' thumbnailURL='' bypassAuto
    <description />
    <Groups>
      <Tag Name='' Value='' />
    </Groups>
    <Permissions modify='' delete='' read='' export='' playback='' />
    <MetaData>
      <Tag Name=''>Value</Tag>
    </MetaData>
    <Download>
      <Media format="Standard" state="NotReady" stale="" downloadURL="" />
      <Media format="Proprietary" state="NotReady" stale="" downloadURL="" />
    </Download>
  <Dir dirNID='' dirTitle='' />
  <ConnectionStatus>
    <FileSessionInfo id='' name=''>
      <FileConnectionInfo id='' name='' state='' type='VIEW/PLAYBACK/RECORD/PVR'
        srcNID='' srcTitle='' dstNID='' dstTitle='' errorCode='' errorData='' wind
      </FileConnectionInfo>
    </FileSessionInfo>
  </ConnectionStatus>
</File>
</GetFile>
```

Prepare File For Download

This is the file that the Client has access for download.

Example 4.35. Prepare File For Download Request

```
<PrepareFile byNID=" "
  format=" "
  requestTimeStamp=" "
  startOffset=" "
  endOffset=" " />
```

- *byNID*="FileNID"
- *format*="Proprietary/Standard/Both" [Optional Default='Proprietary']
- *requestTimeStamp* [Optional]
- *startOffset* [Optional]
- *endOffset* [Optional]

Example 4.36. Prepare File for Download Response

```
<PrepareFile>
  <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>
</PrepareFile>
```

Clear Download Cache

Remove cache memory , which is occupied while downloading media.

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

Example 4.37. Clear Download Cache

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

Example 4.38. Clear Download Cache Response

```
<ClearDownloadCache>
<File
  NID="default.mediastreamfileresourcelist.msfrl_a9504438-5bb1-4e09-a389-c60e
  bypassAutoStorageRules="false" clipID=" " dateCreated="1397505139401"
  dateModified="1397505139401" fileType="MediaClip" mediaGroupID=" "
  mediaID="480193eb-7930-4d04-8d7a-0af670f2e5dd" playlength="0"
  readOnly="false" sessionConfigNID=" " size=" " state="Ready" streamType=" "
  title="tx4">
  <Dir
    NID="default.mediastreamdirresourcelist.msdr1_4e09075d-cf15-44f8-98c3-ae82
    dirTitle="srcrelay1" />
  <Download>
    <Media format=" " stale=" " state="Busy" />
  </Download>
  <Groups>
    <Tag>
      <e Name="Location" Value="Sunnyvale" />
      <e 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>
</ClearDownloadCache>
```

Create Media From Cutlist

Creates Media from existing media using start and end offsets within it

Example 4.39. Create Media From Cutlist Request

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

- *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 4.40. Create Media From Cutlist Response

```
<CreateMediaFromCutList>
  <File NID=" " title=" " fileType=" " streamType=" "
    size=" " playlength=" " mediaID=" " dateCreated=" " dateModified=" "
    readOnly=" " mediaGroupID=" " clipID=" " sessionConfigNID=" "
    bypassAutoStorageRules=" " state=" " ffTracks=" ">
    <description/>
    <Dir NID=" " dirTitle=" "/>
    <MetaData>
      <Tag Name=" "/>
    </MetaData>
    <Groups>
      <Tag Name=" " Value=" "/>
    </Groups>
  </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 4.41. Trim File with Cutlist

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


Example 4.42. 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.msdr1_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>
```

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 4.43. 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 the provided media file to the media store. 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 4.44. 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=txl_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.

Get User Tags

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

Example 4.45. Get User Tags Request

```
<GetUserTags countToFetch="0" startFrom="1"
  sortBy="" sortOrder="ascending"
  whereNameTextSubString="" whereValueTextSubString=""
  selectFormat="" />
```

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

Example 4.46. Get User Tags Response - NameList Format

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

Example 4.47. Get User Tags Response - NameValueList Format

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

Example 4.48. Get User Tags Response - NameValueTree Format

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

Get User

This is the list of Users that the Client has access to. Refer to the Appendix for the format of the User object

Example 4.49. Get User Request

```
<GetUser startFrom="1" countToFetch="100" sortBy=" " sortOrder=" "
selectFormat=" "
whereNID=" "
whereUsername=" "
whereTextSubStr=" "
whereTagExpr=" "
subscriptionID=" " />
```

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

Example 4.50. Get User Response - BASIC Format

```
<GetUser 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>
</GetUser>
```

Example 4.51. Get User Response - EXTENDED Format

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

Get Tags

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

Example 4.52. Get Tags Request

```
<GetTags whereObjectType="" countToFetch="50" startFrom="1"
      sortBy="" sortOrder="ascending"
      whereNameTextSubString="" whereValueTextSubString=""
      selectFormat="NameList/NameValueList/NameValueTree"/>
selectFormat="" />
```

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

Example 4.53. Get Tags Response - NameList Format

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

Example 4.54. Get Tags Response - NameValueList Format

```
<GetTags from="0" total="0">
<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"/>
</GetTags>
```

Example 4.55. Get Tags Response - NameValueTree Format

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

Get DisplayProfile

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

Example 4.56. Get DisplayProfile Request

```
<GetDisplayProfile countToFetch="50" selectFormat=""
sortBy="" sortOrder="ascending" startFrom="1" whereDateRange=""
whereNID="" whereTagExpr="" whereTextSubStr="" />
```

- *sortBy="title"* [Optional]
- *sortBy="NID"* [Not Supported]

Example 4.57. Get DisplayProfile Response - BASIC Format

```
<GetDisplayProfile from="1" total="50">
  <DisplayProfile NID='' title='' portType='' priority='' />
  <DisplayProfile NID='' title='' portType='' priority='' />
  <DisplayProfile NID='' title='' portType='' priority='' />
</GetDisplayProfile>
```

Example 4.58. Get DisplayProfile Response - EXTENDED Format

```
<GetDisplayProfile type="" from="1" total="50">
<DisplayProfile NID='' title='' portType='' priority=''>
  <description />
  <Groups>
    <Tag name='' value='' />
  </Groups>
  <Canvas/>
</DisplayProfile>
</GetDisplayProfile>
```

Session Info Operations

After logging in with the App Service the Client can start to make calls to discover info about current Sessions

Get Session

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 4.59. Get Session Request

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

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

Example 4.60. 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 4.61. 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 4.62. 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>
  </Session>
</GetSession>
```

**Example 4.63. 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 4.64. 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 4.65. 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 4.66. 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 4.67. 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>
```

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 4.68. Get Session Config Request

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

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

Example 4.69. 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 4.70. Get Session Config Response - EXTENDED Format

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

Get Invite

[Not Implemented]

This is a list of explicit session Invites that the client has been sent.

Example 4.71. Get Invite Request

```
<GetInvite startFrom="1" countToFetch="100"
  sortBy="" sortOrder="ascending/descending"
  selectFormat="BASIC/EXTENDED"
  whereId=""
  whereTextSubStr="" />
```

Example 4.72. Get Invite Response - BASIC Format

```
<GetInvite from="0" total="50">
  <Invite id="" title="" sessionId="" entityId="" />
  <Invite id="" title="" sessionId="" entityId="" />
  <Invite id="" title="" sessionId="" entityId="" />
</GetInvite>
```

Chapter 5. Session Setup API

These are the operations a Client would need in relation to "Creating" Sessions.

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 referenced 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 referenced 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 5.1. Basic Start Recording Session Request

```
<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 5.2. 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 5.3. Basic Start Recording Session Request (Continued)

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

Example 5.4. 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="" st
        <User type="" userNID="" username="" userJID="" />
        <Port type="" resourceNID="" resourceTitle="tx1" trickPlay="" />
        <PVR enable="" trimLength="" startTC="" sessionId="" connId="" recEntit
      </Entity>
```

Example 5.5. Advance Start Recording Session Request [Not Implemented]
(Continued)

```
<Entity type="DstPort" id="dst1" name="" streamType="" presence="" gc="" s
  <User type="" userNID='' username='' userJID=""/>
  <Port type="" resourceNID="" resourceTitle="playerpc" trickPlay='false
</Entity>
<Entity type="RecFile" id="recordd1" name="" streamType="" presence="" gc=
  <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="" s
  <User type="" userNID='' username='' userJID=""/>
  <Port type="" resourceNID="" resourceTitle="rtp_tx1" trickPlay=""/>
  <PVR enable="" trimLength="" startTC="" sessionId="" connId="" recEnti
</Entity>
<Entity type="DstPort" id="dst2" name="" streamType="" presence="" gc="" s
  <User type="" userNID='' username='' userJID=""/>
  <Port type="" resourceNID="" resourceTitle="playerpc" trickPlay="true"
</Entity>
<Entity type="RecFile" id="recordd2" name="" streamType="" presence="" gc=
  <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 5.6. 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 5.7. 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.
        msfml_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="true"
      </Entity>
      <Entity type="PlyFile" id="recordd1" name=""
        streamType="" presence="" gc="" state="" errorCode="" errorData="">
        <File type=""
          resourceNID="default.mediastreamfileresourcelist.
          msfml_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="false"
      </Entity>
      <Entity type="PlyFile" id="recordd2" name=""
        streamType="" presence="" gc="" state="" errorCode="" errorData="">
        <File type=""
          resourceNID="default.mediastreamfileresourcelist.
          msfml_c13887ae-73f4-4029-9d37-c4a366156d17"
          resourceTitle="" grpFileId="" />
        </Entity>
    </EntityList>
    <ConnectionList>
      <Connection id="strmc1" srcEntityId="recordd1" dstEntityId="clientd1" canvasI
      <Connection id="strmc2" srcEntityId="recordd2" dstEntityId="clientd2" canvasI
    </ConnectionList>
  </Session>
</StartSession>
```

Example 5.8. 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.
        msfml_081a2b85-b62a-425f-a11a-7a7b6c3bc681"
        resourceTitle=" " dateCreated=" " recsessionConfigNID=" "
        recSessionId="recordSessionID"
        syncEntityId=" " recSessionConfigNID="recordSessionConfigNID"/>
    </GrpFileList>
    <DirList/>
    <ProfileList>
      <Profile id=" " name=" " profileNID=" " ffTracks=" "/>
    </ProfileList>
```

**Example 5.9. 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 5.10. 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 5.11. StopSession Request

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

Publish

Used to publish sessions to users. Specify the Tag name/value pairs for Users who should be able to see this session.

[Not Implemented]

Example 5.12. Publish Request

```
<Publish sessionId="">
  <Tag name="" value="" />
  <Tag name="" value="" />
</Publish>
```

Session Connection Operations

Sessions are created/modified as a result of Connection Requests. All Connection requests share a common set of attributes:

sessionId	if this matches an existing session the connection is added to it - else new session
connGroupId	an optional attribute to put the connection in a connection group
id	the unique Id for this connection

The system will put in values for any attributes left blank

Besides the common attributes, all connections can have an optional explicit routing spec.

The success response in all cases returns the new/modified Session object (refer to the Appendix for details)

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 5.13. Basic View Request by Title

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="" portTitle="tx1" pvrEnable="false"/>
  <DstPort id="" portNID="" portTitle="srcpc1"/>
</View>
```

Example 5.14. 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 5.15. 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 5.16. 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 5.17. 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 5.18. View Request to add a connection from an existing Src to a new Dst

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

Example 5.19. View Request to add a connection from an existing Src to a new Dst with profileid

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

Example 5.20. 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 5.21. 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_elcb45a9-f433-40b2-8156-258c0
      name="V2D_LAN" ffTracks="" />
  </ProfileList>
```


Example 5.22. 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-d0edbfcae3f2" trickPlay="" />
  </Entity>
  <Entity gc="true" id="72a3d1d7-6aa3-481c-ae32-02e7a0a9929a"
    name="pcl" presence="available" streamType="MULTI" type="DstPort" state=""
    <Port resourceTitle="pcl"
      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?enablelkbm=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-d0edbfcae3f2"
      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="" />
    </ConnSeg>
  </Connection>
  <Groups>
    <Tag Name="Location" Value="Sunnyvale" />
  </Groups>
</Session>
```

Example 5.23. 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 5.24. 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="pcl"
  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="pcl" trickPlay="true" />
  />Entity>
</EntityList>
```

Example 5.25. 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.mediastreamdstportresource1ist.
msdprl_ea1024f3-b1a4-4b06-8395-ec6961b61637"
dstResourceTitle="pcl"
profileNID="default.mediastreamprofile1ist.
mspl_b337c45e-5bea-4574-9909-9a623b9cc22d"
profileTitle="V2D_LAN"
srcResourceNID="default.mediastreamrelayportresource1ist.
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 5.26. Basic Playback Request by Title

[Not Implemented]

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

Example 5.27. 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 5.28. 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 5.29. 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 5.30. 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 5.31. 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 5.32. 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="recorddd1" dstEntityId="vdst1" restartSession="" canvasId=""
  <GrpFile id="grpId1" fileNID="" fileTitle="" />
  <SrcFile id="" fileNID="" fileTitle="" grpFileId="grpId1" />
  <DstPort id="" portNID="" portTitle="" />
</Playback>
```

Example 5.33. Playback Request Response

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

Record

Used to setup connections for Recording media.

Use the SrcPort and DstFile 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]
- *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) (Overwritten[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 5.34. Basic Record Request

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

Example 5.35. Basic Record Request with file name/dir spec

```
<Record sessionId="" sessionName="" id="" name="" >
  <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 5.36. Record from an existing Src

```
<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 5.37. Append to Recording Based on Title

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


Example 5.38. 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 5.39. Record Request with Profile NID

```
<Record sessionId="" sessionName="" id="" name="" >
  <Profile id="" profileNID="1234" 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 5.40. 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 5.41. Record from an existing Src with profile id

```
<Record sessionId="" sessionName="" id="" name="" profileId="profileID"
  <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 5.42. 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 5.43. Record Request Response

```
<Session name="srcuser1 Mar 10 21:24" timelineId="c29ccf26-74c4-4e5a-9d3e-f2818e5
recordTimelineId="c29ccf26-74c4-4e5a-9d3e-f2818e54cd1d.RecordTrack"
type="LIVE" gc="true" id="set_file_metada_test_1299792256" state="Started">
  <Properties sessionConfigNID=" " ownerUserJID="srcuser1@localhost/srcpcl" dateC
  <TimelineList/>
  <GrpFileList/>
  <DirList/>
  <ProfileList>
    <Profile id="3980fb57-5fd7-4398-be26-2ca9056b0d63"
      profileNID="default.mediastreamprofilelist.mspl_elcb45a9-f433-40b2-8156-258c
      name="V2D_LAN" />
  </ProfileList>
  <EntityList>
    <Entity gc="false" id="021cf69d-b6e4-4f5e-88e2-56930830d243" type="SrcPort" na
      <Port resourceNID="default.mediastreamsrcportresourcecelist.mssprl_e22023ce-98
    </Entity>
    <Entity gc="false" id="47bdcfe0-b7a1-4e73-alb9-01318f99e654" type="RecFile" na
      <File resourceTitle="set_file_metada_test_1299792256" grpFileId=" " key=" " a
        clipId="47bdcfe0-b7a1-4e73-alb9-01318f99e654" resourceNID="default.med
        playLength="0" mediaGroupId=" ">
      <MetaData>
        <Tag Name="ID">1299792256</Tag>
        <Tag Name="TestCase">AddMediaStreamFile</Tag>
      </MetaData>
    </File>
  </Entity>
</EntityList>
```

Example 5.44. Record Request Response [continued]

```
<ConnectionList>
  <Connection dstEntityId="47bdcfe0-b7a1-4e73-alb9-01318f99e654"
    id="set_file_metada_test_1299792256" srcEntityId="021cf69d-b6e4-4f5e-88e2-5
    type="RECORD" name="set_file_metada_test_1299792256" canvasId="set_file_me
    windowId="set_file_metada_test_1299792256" state="WAITING_FOR_SOURCE-DEST_
  </ConnectionList>
  <Groups>
    <Tag Name="Location" Value="Sunnyvale" />
  </Groups>
</Session>
```

PostSnapshot

Use HTTP POST method to save image on dms. Use GetDir API call with selectformat UPLOADURL to receive upload url from appserver

```
HTTP POST SNAPSHOTURL
request
  Content-Type: multipart/form-data
  entity: a multipart form containing a single file named "file" having content type
response
  Status: 201
  Headers:
    Location: MEDIAURL
    X-IPVS-mediaNID: File NID
```

ViewSnapshot

Use HTTP GET method to view/download snapshot. Use GetFIL API call with selectformat BASICURL/EXTENDEDURL to receive media url from appserver

```
HTTP GET MEDIAURL
request:
  Accept: image/jpeg
response:
  Status: 200
  Content-Type: image/jpeg
  entity: bits
```

ViewThumbnail

Use HTTP GET method to view/download thumbnail. Use GetFile API call with selectformat BASICURL/EXTENDEDURL to receive thumbnail url from appserver

```
HTTP GET THUMBNAIURL
request:
  Accept: image/jpeg
response:
  Status: 200
  Content-Type: image/jpeg
  entity: bits
```

SwitchSrc

Used to switch the src for an existing connections.

Use the SrcEntityId to specify the source to switch to.

Example 5.45. SwitchSrc Request

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

Example 5.46. SwitchSrc Response

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

SwitchDst

[Not Implemented]

Used to modify the dst for an existing connections.

Use the DstPort or DstFile to specify the new resource. Use the attribute form to refer to an existing Dst

Example 5.47. SwitchDst Request

```
<SwitchDst sessionId="" connId="" newDstEntityId=""/>
```

SwitchProfile

[Not Implemented]

Used to modify the profile for an existing connections.

Use the attribute form to refer to an existing Profile

Example 5.48. SwitchProfile Request

```
<SwitchProfile sessionId="" connId="" newProfileId=""/>
```

DeleteConn

Used to terminate and cleanup Conns / ConnGroups.

Example 5.49. DeleteConn Request

```
<DeleteConn sessionId="mySession123" connGroupId=" " connId="myConn123"/>
```

Example 5.50. DeleteConn Response

```
<DeleteConn sessionId="mySession123" connGroupId=" " connId="myConn123"/>
```

Chapter 6. Session Participant API

These are the operations a Client would need in relation to "Participating" in a Session

Session Participant Operations

Invite

[Not Implemented]

Issue an Invite to explicitly invite another user to a session

Example 6.1. Simple Invite by userId or jid

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

Example 6.2. Invite by userId(who) and/or deviceId(from where)

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

Kick

[Not Implemented]

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

Example 6.3. Kick

```
<Kick sessionId="" id="" userName="" jid="" 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 6.4. Join without invite Request

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

Example 6.5. Join without invite Response

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

Example 6.6. Join without invite Response - MUC Room Invite Message

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

Example 6.7. Join with invite

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

Leave

[Not Implemented]

A client may leave a session gracefully at any time

Example 6.8. Leave

```
<Leave sessionId="" />
```

Chapter 7. Session Controller API

These are the operations a Client would need in relation to "Controlling" a Session

Scope for Session Control Operations

- Session Control Operations are qualified by their scope.
- Scope could be at the Session, ConnGroup, or Connection level

Session Scope

Example 7.1. Request with Session scope

```
<xxx sessionId=" " />
```

ConnGroup Scope

Example 7.2. Request with ConnGroup scope

```
<xxx sessionId=" " connGroupId=" " />
```

Connection Scope

Example 7.3. Request with Connection scope

```
<xxx sessionId=" " connGroupId=" " connId=" " />
```

Session MediaStream Operations

[Not Implemented]

These relate to the management of Media Streams within a Sessions. ie: they cannot be used at the Session scope.

Groups of connections within a Session can be Stop/Start'ed using ConnGroup scopes. Typically this is used for Stop/Start of all Recordings etc..

Start

Example 7.4. Start Request

```
<Start sessionId="" connGroupId="" />
```

Stop

Example 7.5. Stop Request

```
<Stop sessionId="" connGroupId="" connId="" />
```

Session PVR Operations

All PVR Operations take an "Action" and a "TimeSpec" and apply to a certain "Scope".

Time for PVR Operations

There are three "Time" references that are important w.r.t. Media Stream Operations during Sessions.

- The "TimeSpec" for PVR Operations is defined in terms of these "Time"s.

NOW	The current Media location.
START	The Media location when the "Start/Create Request" occurred (for the referenced Scope).
LIVE	Where the Media would be without the effects of any PVR Actions .

TimeSpec for PVR Operations

- TimeSpec can be an "Offset" or a "Time" reference.
- In order to maintain "Sync" without requiring the controllers to have their clocks synchronized, PVR commands are designed to work in terms of "Offset" specs (relative to some Server Time) instead of "Time" specs (which they would have to derive from their own (potentially different) idea of WC).
- The only "Time" spec used is based on the TCs already embedded in the Media.

The different types of "TimeSpec" are:

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

<glossentry>*bookmarkId*

<glossdef>

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

</glossdef>

</glossentry>

Example 7.6. Resume Request w/ nowOffset [Deprecated Replace With Play Request w/ nowOffset]

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

Example 7.7. Resume Request w/ startOffset [Deprecated Replace With Play Request w/ startOffset]

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

Example 7.8. Resume Request w/ liveOffset [Deprecated Replace With Play Request w/ liveOffset]

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

Example 7.9. Resume Request w/ mediaOffset [Deprecated Replace With Play Request w/ mediaOffset]

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

Example 7.10. Resume Request w/ mediaTime [Deprecated Replace With Play Request w/ mediaTime]

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

Example 7.11. Resume Request w/ Bookmark [Deprecated Replace With Play Request w/ Bookmark]

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

Example 7.12. Play Request w/ nowOffset

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

Example 7.13. Play Request w/ startOffset

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

Example 7.14. Play Request w/ liveOffset

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

Example 7.15. Play Request w/ mediaOffset

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

Example 7.16. Play Request w/ mediaTime

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

Example 7.17. Play Request w/ Bookmark

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

Action for PVR Operations

All PVR Actions can be done by using just the "Pause" and "Resume" with different TimeSpecs. The others are provided as convenience functions.

- Without a TimeSpec, PVR Actions apply "NOW".

Pause

Example 7.18. Pause Request

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

Example 7.19. Pause Response

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

Resume [Deprecated Replace with Play command]

Example 7.20. Resume Request

```
<Resume sessionId="" />
```

Example 7.21. Resume Response

```
<Timeline sessionId="xyz123" connGroupId="" connId="" id="">
  <Start WC="1252016072510" TC="" Offset="0" parentTimelineId="" />
  <Last WC="1252016139827" TC="" Offset="47358" TimeScale="1" requestedAction="Res
</Timeline>
```

Play

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

- *timeScale* [Playback Speed. Supported Speed 1, 2, 5, 10, 60]

Example 7.22. Play Request

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

Example 7.23. 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>
```

PauseRecord

Example 7.24. PauseRecord Request

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

Example 7.25. 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>
```

ResumeRecord

Example 7.26. ResumeRecord Request

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

Example 7.27. 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>
```

Catchup

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

Example 7.28. Catchup Request

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

Example 7.29. 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>
```

SkipBack

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

Example 7.30. SkipBack Request

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

Example 7.31. 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 7.32. SkipFwd Request

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


Example 7.33. 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>
```

Connection Scope

The "connId" can be a list of connection ids. If any of the connections has a problem with the PVR action, the entire request is flagged as an error

Example 7.34. Resume Request with connection scope

```
<Resume sessionId="" connId="conn1,conn2"/>
```

Example 7.35. Resume Response with connection scope

```
<Timeline sessionId="xyz123" connGroupId="" connId="" id="">
  <Start WC="1252016072510" TC="" Offset="0" parentTimelineId=""/>
  <Last WC="1252016139827" TC="" Offset="47358" TimeScale="1" requestedAction="Res
</Timeline>
```

Example 7.36. Resume Error Response with connection scope

```
<error>
  <Exception>
    <Code>4801</Code><Description>2033:Transition not found for beginState=STOPPED_L
  </Exception>
</error>
<error>
  <Exception>
    <Code>4801</Code><Description>2064:Stream Timeline Not In Sync With Session Ti
  </Exception>
</error>
```

Bookmark Operations

Parameter Listing

- *sortBy*="NID/title/TC" [Optional]
- *sortOrder*="ascending/descending" [Optional]
- *selectFormat*="BASIC/EXTENDED" [Required]
- *whereSessionInfo*="sessionId:userRequestedsessionId, connGroupId:userRequestedConnGroupId, connId:userRequested_connId, entityId:userRequested_entityID" [Optional (Supports sessionId only)]
- *whereFileNID*="FileNID" [Optional]
- *whereNID*="bookmarkNID" [Optional]
- *whereTagExpr*="Name='Location' and Value='Sunnyvale'" [Optional]
- *whereTCRange*="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]
- *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]
- *whereTextSubStr*="" [Optional (Search the given string in Title and MetaData)]

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). [Not Implemented]

Example 7.37. NewBookmark Request

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

- *Either whereSessionInfo or whereFileNID Required*

Example 7.38. NewBookmark Response

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

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 7.39. AddBookmark Request

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

- *Either whereSessionInfo or whereFileNID Required*

Example 7.40. AddBookmark Response

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

UpdateBookmark

Example 7.41. UpdateBookmark Request

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

- *whereNID* Required

Example 7.42. UpdateBookmark Response

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

DeleteBookmark

Example 7.43. DeleteBookmark Request

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

- *whereNID* or *whereSessionInfo* or *whereFileNID* Required

Example 7.44. DeleteBookmark Response

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

GetBookmark

Example 7.45. GetBookmark Request

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

Example 7.46. 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 7.47. GetBookmark Response - EXTENDED Format

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

Chapter 8. Session Notifications

EntityPresence Notification

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

Example 8.1. 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 8.2. 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 8.3. ConnState Notification: Error

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


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 8.4. 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 Timeline includes Info about the Start of the Entity.

- StartWC = The WC at the time this Entity 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 its 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 entities 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 8.5. 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>
```

Chapter 9. Windowing API

The following set of Operations will typically be performed by most Service Clients "outside" of a Session context

Parameter Listing

This is the list of common parameters in Get Requests

- *whereNID*='nid1,nid2,nid3' [Either NID or TextSubString Required]
- *whereTextSubStr*="" [Either NID or TextSubString Required]
- *subscriptionID*='' [optional (if subscriptionID give will subscribe for object updates)]

Get Operations

Get Port Display Configuration Request

Gets the display configuration for given port.

Example 9.1. Get Port Display Configuration Request

```
<GetPortDisplayConfig whereNID='NID1,NID2' whereTextSubStr='' subscriptionID=''
```

Example 9.2. Get Port Display Configuration Response

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

Get Port Display Status Request

Gets the display status for given port.

Example 9.3. Get Port Display Status Request

```
<GetPortDisplayStatus whereNID='NID1,NID2' whereTextSubStr='' subscriptionID=''
```

Example 9.4. Get Port Display Status Response

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

Set Operations

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 9.5. Set Port Display Configuration Request

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

Example 9.6. Set Port Display Configuration Response

```
<SetPortDisplayConfig>
</SetPortDisplayConfig>
```

Set port command Parameter Listing

- *bySessionId* [Either (sessionId and connId) or portNID required]
- *byConnId* [Either (sessionId and connId) or portNID required]
- *byNID* [Either (sessionId and connId) or portNID required]
- *byCanvasId* [Required]

- *byLayoutId* [Required]
- *byWindowId* [Required]

Set Port Window Properties

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

Example 9.7. Set Port Window Properties Request

```
<SetPortWindowProperties 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=''/>
</SetPortWindowProperties>
```

- *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 9.8. Set Port Window Properties Response

```
<SetPortWindowProperties>  
</SetPortWindowProperties>
```

Chapter 10. 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:allowKMSSwitch*='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 10.1. AuthConectionInfo element in AuthorizationRequired Notification

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

Example 10.2. 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=''
    </PortConnectionInfo>
  </PortSessionInfo>
</ConnectionStatus>
```

Example 10.3. 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 10.4. 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 10.5. 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 10.6. Port Update Notification BASIC

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

Example 10.7. 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 10.8. 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 10.9. 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 10.10. 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 10.11. 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 10.12. 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 10.13. 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 10.14. 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 10.15. File Connection Update Notification

```
<FileConnectionUpdate>
  <File NID='' subscriptionID='' operation='Add/Delete'>
  <FileSessionInfo .....
```

File Connection State Update

This is a notification for file connection state update.

Example 10.16. File Connection State Update Notification

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

Chapter 11. External Authorization

AuthorizationRequired Message

Auth user receive this message from appserver when user need external authentication

Example 11.1. AuthorizationRequired Message

```
<AuthorizationRequired>
  <AuthConnectionInfo ...../>
  <!-- Check AuthConnectionInfo element in ConnectionUpdate Section of
</AuthorizationRequired>
```

Authorize Request

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

Example 11.2. Authorize Request

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

Example 11.3. Authorize Response

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

Chapter 12. Admin API

SetFileMetaData

replace file metadata in database

- *byNID* [byNID required]
- *MetaData* [Required]

Example 12.1. Set File MetaData Request With NID

```
<SetFileMetaData byNID='nid1'>
  <MetaData>
    <Tag Name='ID'>12345</Tag>
    <Tag Name='TestCase'>UpdateWithByNID</Tag>
  </MetaData>
</SetFileMetaData>
```

Example 12.2. Set File MetaData Request With MetaData

```
<SetFileMetaData byTitle="1300383357">
  <MetaData>
    <Tag Name="ID">1300383357</Tag>
    <Tag Name="TestCase">UpdateMediaStreamFile-ID</Tag>
  </MetaData>
</SetFileMetaData>
```


Example 12.3. 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='' />
</File>
</SetFileMetaData>
```

DeleteFileRequest

Delete the given file from the system

- *byNID* [Either byNID]
- *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 12.4. DeleteFile Request by NID

```
<DeleteFile byNID='' moveToRecycleBin='false' />
```

Example 12.5. DeleteFile by clip file nid or image file nid Response

```
<DeleteFile>
<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>
</DeleteFile>
```

Example 12.6. DeleteFile by Group File nid Response

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

Set User Password

This is used to set the user password.

Example 12.7. Set User Password Request

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

Example 12.8. Set User Password Response

```
<Resource>  
  <User NID="" username="" email="" fullName="" creationDate="" modifiedDate="" pr  
    <Groups>  
      <Tag Name="" Value="" />  
    </Groups>  
    <UserPrivateData />  
  </User>  
</Resource>
```

Device Object

This section explains list of Device Admin API calls

AddDevice

Add asset,user,ports and device in database

Example 12.11. Set Device Info Request

- *descriptions* [Required]

```
<SetDeviceInfo byNID='' byTitle='' description='' password='' applyTagST
<Groups>
  <Tag Name='' Value='' />
</Groups>
<DevicePrivateData />
</SetDeviceInfo>
```

Example 12.12. Set Device Info Response

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

Get Device

Lists the devices present in the database.

Example 12.13. Get Device Request

```
<GetDevice startFrom='1' countToFetch='50' sortBy='' sortOrder='ascending'  
selectFormat='BASIC/EXTENDED' whereNID='' whereTitle='' whereType='' whereTextS
```

- *sortBy*="title/type" [Optional]
- *sortOrder*="ascending/descending" [Optional]
- *whereNID* [Optional]
- *whereTitle* [Optional]
- *whereType*="IPVSCClient-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* [Optional]
- *subscriptionID* [Optional]

Example 12.14. Get Device Response - BASIC Format

```
<GetDevice 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>
</GetDevice>
```

Example 12.15. Get Device Response - EXTENDED Format

```
<GetDevice from='0' total='50'>
  <Device NID='' title='' type='' description='' presence='' state='' softwareVersion=''>
    <Groups>
      <Tag name='' value='' />
    </Groups>
    <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=''>
          <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>
        </XMPPServer>
      </Services>
    </DeviceConfig>
  </Device>
</GetDevice>
```


Example 12.16. Get Device Response - EXTENDED Format [Continued]

```
<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' mu
  </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>
<AVCodec>
  <Port id=''>
    <Displayconfig type='IPVSCClientDisplayConfig'>
      <Display configurable="" />
    </Displayconfig>
  </Port>
</AVCodec>
```

Example 12.17. Get Device Response - EXTENDED Format [Continued]

```
<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' />
  <ConnectionContent MediaStreamIOPortTypeTitle="" MediaStreamIOPortElementTitle="" />
  <ThumbnailContent MediaStreamIOPortTypeTitle="" MediaStreamIOPortElementTitle="" />
  <Geometry zIndex="" border="NONE/THIN/THICK" vpos='ALIGNTOP/ALIGNMIDDLE/ALIGNBOTTOM' />
  aspect='SourceWidth/SourceHeight/Auto/None' transparency="NNN%" />
</Window>
</Canvas>
</Displayconfig>
<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>
</GetDevice>
```

Example 12.18. Get Device Response - RAW Format [Continued]

```
        <GetDevice 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>
<DeviceConfig> <!-- Device specific response --> </DeviceConfig>
<DeviceStatus> <!-- Device specific response --> </DeviceStatus>
<DeviceList />
</Device>
</GetDevice>
```

DeleteDevice

Delete device from database

Example 12.19. Delete Device Request

```
<DeleteDevice byNID='' byTitle='' />
```

Example 12.20. Delete Device Response

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

Device Object Response XML

Example 12.21. Device Object Response XML

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

Example 12.22. Device Object responses XML [Continued]

```
<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 isPrimary='' portID='' enabled='true/false' multicast='true/f
  </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>
```

Example 12.23. Device Object responses XML [Continued]

```
        <AVCodec>
    <Port id=''>
    <Displayconfig type='IPVSCClientDisplayConfig'>
    <Display configurable="" />
    <Canvas>
    <Background enable='true/false' zIndex='0' content='OS/Image/Session'>
    <ImageContent imageURL="" color="" />
    <SessionContent enableConnPanel="" enableBookmarkPanel="" enableCtrlPane
    </Background>
    <Window id='userDefinedStr' enable='true/false' content='MediaThumbnail/M
    <ConnectionContent MediaStreamIOPortTypeTitle="" MediaStreamIOPortElemen
    <ThumbnailContent MediaStreamIOPortTypeTitle="" MediaStreamIOPortElement
    <Geometry zIndex="" border="NONE/THIN/THICK" vpos='ALIGNTOP/ALIGNMIDDLE/
    aspect='SourceWidth/SourceHeight/Auto/None' transparency="NNN%" />
    </Window>
    </Canvas>
    </Displayconfig>
    <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 Object

This section explain list of Port object API calls

Parameter Listing Of Set API

Either *byNID* or *byTitle* are used to get the port object from the database. It is not required to use both together.

- *byNID* [Required]
- *byTitle* [Required]

SetPortInfo

Update port title, description, and groups in the database

Example 12.24. Set Port Info Request

- *title* [Required]
- *description* [Optional]
- *Groups* [Optional]

```
<SetPortInfo byNID='' byTitle='' title='' description=''>
  <Groups>
    <Tag Name='' Value='' />
  </Groups>
</SetPortInfo>
```

Example 12.25. Set Port Info Response

```
<SetPortInfo>
  <!-- Check Port Object Response XML-->
</SetPortInfo>
```

SetPortStreamURL

Update port stream URL in the database

Example 12.26. Set Port StreamURL Request

- *streamURL* [Optional]

```
<SetPortStreamURL byNID='' byTitle='' streamURL='' />
```

Example 12.27. Set Port StreamURL Response

```
<SetPortStreamURL>
  <!-- Check Port Object Response XML-->
</SetPortStreamURL>
```

SetPortObserverInfo

Update observer information in the database

Example 12.28. Set Port Observer Info Request

- *xAuth* [Optional]
- *monitor* [Optional]
- *UserNID* [Optional]
- *ResourceNID* [Optional]

```
<SetPortObserverInfo byNID='' byTitle='' >
<Observer xAuth='' monitor=''>
  <UserNID></UserNID>
  <ResourceNID></ResourceNID>
</Observer>
</SetPortObserverInfo>
```

Example 12.29. Set Port Observer Info Response

```
<SetPortObserverInfo>
  <!-- Check Port Object Response XML-->
</SetPortObserverInfo>
```

SetPortPVRInfo

Update the PVR settings in the database

Example 12.30. Set Port PVR Info Request

- *enable* [Required]
- *trimLength* [Optional - default is 4hrs]
- *mediaStreamDirResourceNID* [Required]

```
<SetPortPVRInfo byNID='' byTitle='' enable='true/false' trimLength='' medi
```

Example 12.31. Set Port PVR Info Response

```
<SetPortPVRInfo>  
  <!-- Check Port Object Response XML-->  
</SetPortPVRInfo>
```

Port Object Response XML

Example 12.32. Port Object Response XML

```
<Port NID='' title='' streamType='' presence='' state='' portType=''>  
<Asset NID='' title='' deviceType='' deviceNID=''/>  
<PVRInfo enable='' trimLength='' mediaStreamFileResourceNID='' mediaStreamDi  
<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>
```

Chapter 13. Report API

This chapter focuses on requests that are specific to generating reports.

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 13.1. Audit Event Request

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

Example 13.2. Audit Event Response

```
<AuditEvent/>
```

Chapter 14. Frequently Asked Questions

- 14.1.** Do I need to create 4 SetPortDisplayConfig commands One for each video stream?

A single call can include all the window configurations in a single canvas.

- 14.2.** Can a SetPortDisplayConfig command include multiple canvases?

A single call can include a single canvas only.

- 14.3.** Can a SetPortDisplayConfig command include multiple windows?

A single call can include all the window configurations in a single canvas.

- 14.4.** One more thread on Section 10 of the API. I am confused with the use of “whereNID” in this section. It does not appear to be consistent with usage elsewhere. I would expect the setPortDisplayConfig and setServiceClientDisplayConfig messages to include NID attributes that would set the NID for these data structure. From here, the other messages would use the “whereNID” to reflect the proper DisplayProfile object when either getting or reconfiguring aspects of that data object?

The only call available is setPortDisplayConfig which requires the NID to specify the port for which the display config is to be updated

- 14.5.** After I do a start session, I get the following message (which seems malformed). I have also noticed that the sessions I start from my SW no longer show up in the sessions tab on the Management UI. Do I now need to do something else after I start a session (such as Join that session)?
<message id="ao59r-16077" to="vms1@localhost/vms1" from="managementserver.localhost">
<body><MediaRoomInvite><roomNID>b8aeda70-3e32-4df3-92b4-a6b0a4660f1d< /roomNID></MediaRoomInvite></body> </message>

You do not have to join the session This is only for agents (Source/Dest) or if you want to receive Session Events.

- 14.6.** Are there parameters in the window that we do not allow the UI/API user to change ?

transparency, border,aspect attribute is not allowed from the UI/API User. <Background/>, <ThumbnailContent/>, <ConnectionContent/> are not used in the UI

- 14.7.** Does the UI set the sessionId, windowId in the ConnectionContent ?

UnManned client does not update the session Id and connection Id, since we cannot get the sessionid and connection id

Appendix A. Data Model

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

Resource Objects

The "Resource" Objects are retrieved using the Get Request API and used as parameters during session connection operations.

```
<User NID="" title="" jid="" presence="" />

<Dir NID="" title="" dirType="" presence="" state="" />

<File NID="" title="" fileType="" streamType=""
      size="" playlength=""
      presence="" state=""
      dateCreated="" dateModified=""
      readOnly="" mediaGroupId="" />
```

The dateCreated/dateModified are epoch times in milliseconds.
The readOnly flag can be set to "true"/"false"
The fileType can be set to "MediaClip"/"MediaGroup"

```
<Port NID="" title="" portType="" streamType="" presence="" state="" />

The portType can be set to "SrcPort"/"DstPort"/"StorePort"/"RelayPort"

<Profile NID="" title="" streamType="" />

<Session id="" title="" />
```

Resource Attributes

The "Resource" attributes reflect the state and capabilities of the "Resource" Objects. This section lists the common resource attributes.

```
<PresenceValueList>
  <Presence value="unavailable"/> <!-- not available for communication -->
  <Presence value="available"/> <!-- available for communication -->
</PresenceValueList>

<StateValueList>
  <State value="Offline"/> <!-- owner not logged in -->
  <State value="Ready"/> <!-- available for use in a connection -->
  <State value="NotReady"/> <!-- not available for use in a connection -->
```

```
<State value="Busy"/> <!-- being currently updated -->
</StateValueList>

<StreamTypeValueList>
  <StreamType value="V2D"/>
  <StreamType value="RTP"/>
  <StreamType value="MPEGTS"/>
  <StreamType value="UDP"/>
</StreamTypeValueList>
```

Object Info

The "Resource" Objects have more detailed Info that can be retrieved using the Get Request API with the higher select Level. All objects support the common extInfo. Additional Info may be available for some objects at higher select levels.

```
<extInfo>
  <description/>
  <Groups>
    <Tag name=" " value=" "/>
  </Groups>
  <MetaData>
    <Tag Name=" ">Value</Tag>
  </MetaData>
</extInfo>

<File>
  <Dir dirNID=" " dirTitle=" "/>
</File>

<Port>
  <Device NID=" " title=" " deviceType=" "/>
</Port>

<Session>
  <Properties/>
  <Groups/>
  <MetaData/>
  <TimeLineList/>
  <GrpFileList/>
  <DirList/>
  <ProfileList/>
  <EntityList/>
  <ConnectionList/>
</Session>
```

Session Objects

The "Session" is the runtime Object that represents a collaboration config and state.

```
<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="" TC="" Offset="" parentTimelineId=""/>
      <Last WC="" TC="" Offset="" TimeScale="" requestedAction=""/>
    </Timeline>
  </TimelineList>
  <GrpFileList>
    <GrpFile type="" id="" resourceNID="" resourceTitle=""
      dateCreated="" recSessionConfigNID="" 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>

  <EntityList>
    <Entity type='SrcPort' id="" name="" streamType="" presence="" gc="" state=""
      <User type="" userNID="" username="" userJID=""/>
      <Port type="" resourceNID="" resourceTitle="" trickPlay=""/>
      <PVR enable="" trimLength="" startTC="" connId="" recEntityId="" plyEntityId="">
    </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="AuthClient" id="" name="" streamType="" presence="" gc="" stat
  <User userNID="" username="" userJID=""></User>
  <AuthCtrl resourceNID="" resourceTitle=""></AuthCtrl>
</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=""/>
</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>

<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>
```

DisplayProfileInfo

The DisplayProfile is the runtime Object that represents a Canvas.

```
<Canvas>
  <Background enable="" zIndex="" content="">
    <ImageContent imageURL="" color=""></ImageContent>
    <SessionContent enableConnPanel="" enableBookmarkPanel=""
      enableCtrlPanel="" enableHeaderPanel=""
      enablePlayerPanel="" sessionId="">
    </SessionContent>
  </Background>

  <Window id="" enable="" content="">
    <ConnectionContent MediaStreamIOPortTypeTitle=""
      MediaStreamIOPortElementID=""
      sessionId="" connId="">
    </ConnectionContent>
    <ThumbnailContent MediaStreamIOPortTypeTitle=""
      MediaStreamIOPortElementID=""
      imageURL="" refreshInterval="">
    </ThumbnailContent>
    <Geometry zIndex="" border="" vpos="" hpos=""
      vres="" hres="" aspect=""
      transparency="">
    </Geometry>
    <SubCanvas></SubCanvas>
  </Window>
</Canvas>
```

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>
```

XMPP API Tutorial

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

Prerequisites

Prerequisites for the tutorial:

- Login to admin UI

- Add user: proxyuser
- Add Device ProxyClient: proxypc

Tutorial

Run XMPP Client

Start XMPP Client

- Run the XMPP client "java -jar ipvsxmppclient.jar -p port
- Connect over telnet using the port number enter in above command

Help Command

Help command

```
<help/>
```

Login Command

```
<login ip='10.51.48.190' user='proxyuser' password='Device123' resource='proxypc' />
```

- ip : MS ipaddress
- user : username
- resource : proxy client added from AdminUI.

Service Login

```
<iq><query><ServiceLogin clientVersion='2.30.2' /></query></iq>
```

For more information about API check chapter 3. Copy service version from API document.

Record API

Start Record Session

```
<iq><query><Record><SrcPort portTitle='rtptx' /><DstFile fileTitle='record' /></Record></query></iq>
```

- Add rtptx unmanaged source in Admin UI
- This will return

```
<Session id='c00eab00-ad3d-4865-9ed1-4d14ac5a0c7c' .../>
```

- For more information about API check chapter 5 page number 79

Make sure recording has started use GetSession API

```
<iq><query><GetSession whereId='c00eab00-ad3d-4865-9ed1-4d14ac5a0c7'
```

- This will check return recording session information
- For more information about API check chapter 4 page number 47

Stop Recording Session

```
<iq><query><StopSession sessionId='c00eab00-ad3d-4865-9ed1-4d14ac5a0c7c' /></i>
```

For more information about API check chapter 5 page number 67

Make sure recording stopped use GetSession API

```
<iq><query><GetSession whereId='c00eab00-ad3d-4865-9ed1-4d14ac5a0c7c' /></i>
```

- This will not return requested session information.
- For more information about API check chapter 4 page number 47

Release Notes

Summary of changes from 3.10 to 3.12

Added Requests

AddReport: Add a user configured report
AuditEvent: Send an audit event corresponding to each user action
DeleteReport: Delete the selected report
DownloadReport: Download the report for analysis
GetOrphanFile: Get list of orphan files
GetPVRFile: Get list of PVR files
GetReport: View the list of configured reports
GetSystemConfig: Get the configuration of the management server
SetReport: Update the existing report
StopSession: Stop a specific session

Added Notifications

ConnectionUpdate: Connection is added or deleted
ConnectionStateUpdate: State of a connection is updated
PortUpdate: Port is updated
PortConnectionUpdate: Connections are added/deleted
PortConnectionStateUpdate: State of a port connection updated

Updated Requests

SetUserPassword: Added whereNID attribute
GetFile: Added whereState, whereMetaDataExpr, wherePermissions, whereType, where
GetFileTags: Added wherePermissions
GetDir: Added wherePermissions and whereFilePermissions
GetDirTags: Added wherePermissions
GetSession: Added whereId
GetPortTags: Added wherePermissions
View: Added pvrEnable attribute to SrcPort element
StartSession: Added type, gc, and state attributes
Record: Added resetTimeline attribute

Summary of changes from 3.12 to 3.14

Updated Requests

All the Set requests: Changed where parameters

Summary of changes from 3.14 to 3.16

Updated Requests

GetPort: Added subscriptionID attribute in request and portId attribute in Port
GetPort: Separated CONNECTIONINFO response type into BASICCONNECTIONINFO and EXTE
GetPort: Changed ConnectionInfo element to PortConnectionInfo and SessionInfo el
AuthorizationRequired: Changed element ConnectionInfo to AuthConnectionInfo
Implemented connection notification for Port Object
Added subscriptionID attribute for GetPortDisplayConfig and GetPortDisplayStatus
Updated SetPortDisplayConfig and SetPortWindowProperties Responses
Added Display Config and Status Update notification for Port Object
Updated GetDevice, GetDir, GetFile, GetUser, GetSessionConfig request and respon

Summary of changes from 3.16 to 3.18

Updated Requests

GetFile: Added new selectFormat BASICURL and EXTENDEDURL which will return media
GetDir: Added new selectFormat SANPSHOTURL which will return snapshot URL. Which
Added Add/Get/Set/Delete Device API and Port Set API
Added HTTP API
GetFile, SetFileMetaData, DeleteFile: Added bypassAutoStorageRules attribute in t
GetDir: Added StorageConfig element in the response and moved port element to th

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