



# Session Resource Manager

Version 5.1STD1

## SRM User Guide

Filename	SRM_USE_SRMUserGuide_5.1STD1_pub.pdf
Document Nr	SRM-USE
Date	2018-10-15
Author(s)	KUDELSKI SA
Information domain	
Data Owner	Nagra Media UK Limited

KUDELSKI SA

22-24, Route de Geneve, Case Postale 134, 1033 Cheseaux , Switzerland

tel: [+41 21 732 01 01](tel:+41217320101), +41 21 732 01 00

<http://www.nagra.com/>

All trademarks and registered trademarks are the property of their respective owners.

This document is supplied with an understanding that the notice(s) herein or any other contractual agreement(s) made that instigated the delivery of a hard copy, electronic copy, facsimile or file transfer of this document are strictly observed and maintained.

The information contained in this document is subject to change without notice.

**Security Policy of Nagra Media UK Limited Kudelski Group**

Any recipient of this document, without exception, is subject to a Non-Disclosure Agreement (NDA) and access authorization.

Nagra

## Tracking data

### Contributors

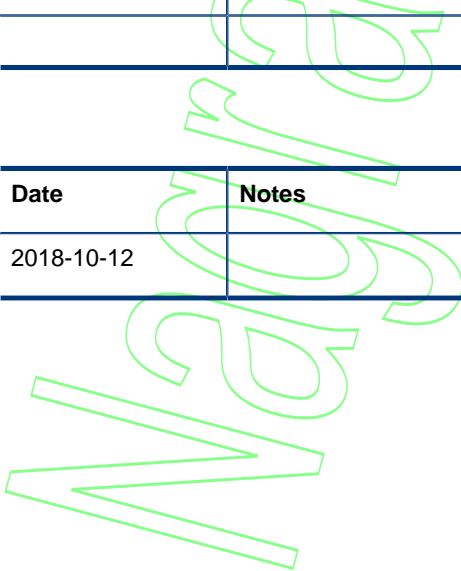
Event	Actor	Date	Notes
reviewer	Matt Powell	2018-10-12	
approver	Andreas Daniel	2018-10-12	

### Reviewers

Reviewer	Date	Notes

### Approvers

Approver	Date	Notes
Andreas Daniel	2018-10-12	



## Contents

<b>1 SRM.</b>	<b>10</b>
1.1 Session management.	10
1.1.1 Session contributor.	10
1.1.2 Contributor device.	10
1.1.3 Pav device.	11
1.1.4 SrmServer device.	11
1.1.5 Action matrix.	11
1.1.6 Asynchronous teardown.	12
1.1.7 Partial teardown.	12
1.2 Resource management.	12
1.2.1 Table-based bandwidth allocation.	13
1.3 A typical deployment.	13
1.4 SRM and NGOD.	14
<b>2 User tasks.</b>	<b>16</b>
2.1 Deploy the SRM.	16
2.2 Install and configure MongoDB.	16
2.3 Install the SRM.	18
2.3.1 Share an application server with the SDP.	18
2.4 Configuring the SRM.	19
2.4.1 Configure logging.	20
2.4.2 Default logging configuration.	20
2.4.3 Default users and passwords.	21
2.5 Start and stop the SRM.	22
2.5.1 Shut down an SRM cleanly.	22
2.6 Ingest data.	23
2.6.1 Constraints on data.	23
2.7 Migrate from a legacy SRM.	24
2.7.1 SRM Typical Manual Migration.	26

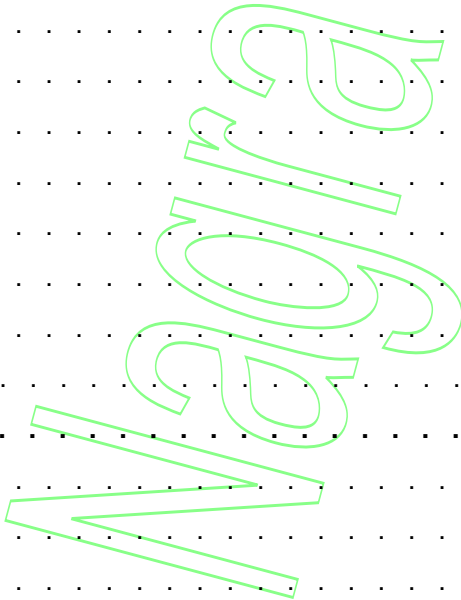
2.7.2 SrmClient Ingest Samples. . . . .	29
2.8 SRM User Session Limitation. . . . .	31
<b>3 Monitoring. . . . .</b>	<b>33</b>
3.1 Monitored quantity. . . . .	33
3.2 Monitored viewing periods. . . . .	33
3.3 Monitored histogram. . . . .	34
3.4 Monitored object. . . . .	35
3.4.1 Monitored quantities of contributor devices. . . . .	36
3.4.2 Monitored quantities of Pav devices. . . . .	36
3.4.3 Monitored interface. . . . .	36
3.4.4 Monitored session manager. . . . .	37
3.4.5 Monitored connection manager. . . . .	38
3.4.6 Monitored thread manager. . . . .	39
3.5 Monitored device health status. . . . .	40
<b>4 Configuration reference. . . . .</b>	<b>42</b>
4.1 core-conf.xml. . . . .	42
4.1.1 Configuration bean: bodcmConfig. . . . .	42
4.1.2 Configuration bean: mongoDB. . . . .	43
4.2 srm-conf.xml. . . . .	46
4.2.1 Configuration bean: ermConfig. . . . .	46
4.2.2 Configuration bean: sessionConfig. . . . .	47
4.2.3 Configuration bean: srmCoreConfig. . . . .	48
4.2.4 Configuration bean: vsConfig. . . . .	58
4.3 stats-conf.xml. . . . .	61
4.3.1 Configuration bean: monitoredHistogramTemplates. . . . .	61
<b>5 Data ingest reference. . . . .</b>	<b>63</b>
5.1 Data ingest file format. . . . .	63
5.2 Block headings. . . . .	63
5.2.1 Record types. . . . .	64

5.2.2 Operations. . . . .	65
5.3 Column headings. . . . .	66
5.4 Data records. . . . .	66
5.4.1 Data ingest: ChannelBandwidth fields. . . . .	67
5.4.2 Data ingest: SrmServer device fields. . . . .	68
5.4.3 Data ingest: DeviceGroup fields. . . . .	68
5.4.4 Data ingest: Range. . . . .	69
5.4.5 Data ingest: Erm device fields. . . . .	70
5.4.6 Data ingest: Pav device fields. . . . .	71
5.4.7 Data ingest: ServiceGroup fields. . . . .	73
5.4.8 Data ingest: ServiceGroupResource fields. . . . .	75
5.4.9 Data ingest: SrmClient fields. . . . .	76
5.4.10 Data ingest: Sts device fields. . . . .	78
5.4.11 Data ingest: Scs device fields. . . . .	79
<b>6 SRM API reference. . . . .</b>	<b>82</b>
6.1 Operator and Management APIs. . . . .	82
6.1.1 Overall URL structure for web UI. . . . .	82
6.1.2 Bulk data ingest. . . . .	82
6.1.3 Force disconnect. . . . .	86
6.1.4 Machine diagnostics. . . . .	93
6.1.5 Monitoring device health. . . . .	95
6.1.6 Monitoring device list. . . . .	97
6.1.7 Monitoring statistics. . . . .	99
6.1.8 Session Discovery. . . . .	103
6.1.9 Data Discovery. . . . .	109
6.1.10 User-behavior reporting. . . . .	112
6.2 OTT Session Management. . . . .	119
6.2.1 Session Setup. . . . .	120
6.2.2 Session Teardown. . . . .	127



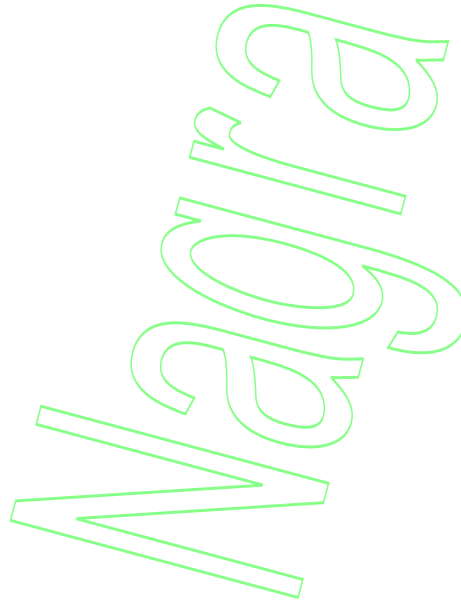
6.2.3 Session Keepalive. . . . .	132
6.3 RTSP Gateway. . . . .	136
6.3.1 SETUP. . . . .	136
6.3.2 TEARDOWN. . . . .	162
6.3.3 GET_PARAMETER. . . . .	168
6.3.4 SET_PARAMETER. . . . .	172
6.3.5 ANNOUNCE handling. . . . .	175
6.3.6 SDV. . . . .	179
6.3.7 OpenTV 2.4. . . . .	188
6.3.8 NGOD C1 Interface. . . . .	189
6.3.9 NGOD R2 Interface. . . . .	206
6.3.10 NGOD Response And Notice Codes. . . . .	262
6.3.11 NGOD S1. . . . .	265
6.4 Error Codes and status values. . . . .	313
6.5 SRM-Adapter SETUP. . . . .	316
6.6 SRM-Adapter Setup Examples. . . . .	320
<b>7 User reference. . . . .</b>	<b>322</b>
7.1 Default users and passwords. . . . .	322
7.2 MongoDB write safety levels. . . . .	322
7.3 Ports used by the SRM. . . . .	323
7.4 QAM modulation codes. . . . .	324
7.5 Session contributors. . . . .	325
<b>8 Puppet Modules . . . . .</b>	<b>327</b>
8.1 Exposing SRM Operator API. . . . .	327
<b>9 Release Notes. . . . .</b>	<b>329</b>
<b>10 Release Notes: 3.0. . . . .</b>	<b>330</b>
10.1 SRM 3.0STD1. . . . .	330
10.2 SRM 3.0STD2. . . . .	331
10.3 SRM 3.0STD3. . . . .	333

<b>11 Release Notes: 3.1.</b>	<b>336</b>
11.1 SRM 3.1STD1.	336
11.2 SRM 3.1STD2.	337
11.3 SRM 3.1STD3.	338
11.4 SRM 3.1STD4.	340
11.5 SRM 3.1STD5.	341
<b>12 Release Notes: 3.2.</b>	<b>344</b>
12.1 SRM 3.2STD0.	344
12.2 SRM 3.2STD1.	346
12.3 SRM 3.2STD2.	348
12.4 SRM 3.2STD3.	349
12.5 SRM 3.2STD4.	351
12.6 SRM 3.2STD5.	353
12.7 SRM 3.2STD6.	355
12.8 SRM 3.2STD7.	355
12.9 SRM 3.2STD8.	356
12.10 SRM 3.2STD9.	357
<b>13 Release Notes: 4.0.</b>	<b>359</b>
13.1 SRM 4.0STD1.	359
13.2 SRM 4.0STD3.	360
13.3 SRM 4.0STD4.	361
13.4 SRM 4.0STD5.	362
<b>14 Release Notes: 5.0.</b>	<b>365</b>
14.1 SRM 5.0STD0.	365
14.2 SRM 5.0STD1.	366
14.3 SRM 5.0STD2.	368
14.4 SRM 5.0STD4.	370
14.5 SRM 5.0STD5.	371
14.6 SRM 5.0STD6.	372





14.7 SRM 5.0STD7. . . . .	372
14.8 SRM 5.0STD8. . . . .	373
14.9 SRM 5.0STD9. . . . .	374
<b>15 Release Notes: 5.1. . . . .</b>	<b>376</b>
15.1 SRM 5.1STD0. . . . .	376
15.2 SRM 5.1STD1. . . . .	377



# 1 SRM

The Session Resource Manager, or SRM, orchestrates the process of streaming video to cable set-top boxes. It provides

- ▶ [Session management](#) <sup>p.10</sup>, including forced session termination
- ▶ [Resource management](#) <sup>p.12</sup>
- ▶ Health monitoring and reporting data

## 1.1 Session management

Session management is the primary function of the SRM.

A session manager

- ▶ receives requests from clients to set up and tear down streaming sessions
- ▶ keeps track of the status of active streaming sessions
- ▶ coordinates back-office devices and other elements of the system, in particular ERM and streaming servers (video servers), to provide the resources required by each session.

### 1.1.1 Session contributor

A session contributor is a provider of some kind of resource that needs to be allocated to deliver a stream.

Most streaming sessions require resources from different locations to be coordinated. For example, to deliver a stream in a cable environment, bandwidth needs to be allocated on a QAM and also on a video server. The SRM models the QAM and video server as session contributors.

When the SRM sets up a session, it instructs all the session contributors for that type of session to perform their own set up tasks. The stream can be delivered only if all the session contributors report successful setup.

The most important session contributors are those which manage other devices.

### 1.1.2 Contributor device

A **contributor device** is a device which is involved in setting up and tearing down sessions, and which stores data related to sessions which are set up on it. Examples of these include Edge Resource Managers (Erm records) and Streaming servers (Sts records)

You must ingest at least one Sts record before attempting to start any sessions.

Unless you are using [Table-based bandwidth allocation](#) <sup>p.13</sup>, you must ingest at least one Erm record before attempting to start any sessions.

#### Caution!

When a session is being set up or torn down, the appropriate set of contributor devices will be called by the corresponding [Session contributors](#) <sup>p.325</sup>.

Although a [Pav device](#) <sup>p.11</sup> will be called (by a session contributor) during session setup, it is not considered to be a contributor device. This is because it does not store any session-related information itself, and so it is not subject to the [Constraints on data](#) <sup>p.23</sup> which apply to contributor devices.

Contributor devices support <https://atlassian.hq.k.grp/confluence/display/SRM/Monitoring+device+health>, exposing the [Monitored quantities of contributor devices](#) <sup>p.36</sup>

### 1.1.3 Pav device

A **Pav** device ("Playout Authorization Verification") checks that clients are authorized to set up a session on the SRM to play out the content they request.

Currently all Pav devices are instances of the SDP.

Pav devices support <https://atlassian.hq.k.grp/confluence/display/SRM/Monitoring+device+health>, providing the [Monitored quantities of Pav devices](#) <sup>p.36</sup>

**Note** You must ingest at least one Pav device.

### 1.1.4 SrmServer device

An SrmServer device represents an instance of the SRM itself. This is used to send inter-SRM announcements in the case where a session is terminated administratively. This allows the client keepalive connection for the terminated session to be closed, even when the keepalive connection is on a different SRM instance.

SrmServer devices support [Monitoring device health](#) <sup>p.95</sup>

SrmServer devices do not expose any [Monitored quantity](#) <sup>p.33</sup>

You must ingest at least one SrmServer device. Even if there is only one SRM instance in the deployment (not recommended), it uses the inter-SRM announcement mechanism to talk to itself.

### 1.1.5 Action matrix

Each type of session handled by the SRM is associated with an action matrix, which defines which session contributors need to be coordinated, and what order they need to be called in.

When a session is set up, each session contributor is called upon in turn to add some kind of resource to the session.

When all contributors have successfully added their contribution, the session is ready to use, and relevant information is returned to the client. Similarly, when a session is torn down, the session contributors are called in order, to free the resources they contributed. The ordering used for teardown is independent from the setup ordering: some contributors may not appear in the teardown ordering at all.

There may be multiple action matrixes in use in a given deployment: for example, some equipment is older, and works one way, while other equipment is newer, and works another way. The required workflows depend on the hardware in use.

**Caution!** Although the file which defines the action matrix is not hard to find, the matrix is **not** user-configurable. Contact Nagra if your deployment will need to define a new action matrix.

## 1.1.6 Asynchronous teardown

When a client requests that a session be torn down, the SRM acknowledges the request quickly, but performs the teardown itself later. (Usually only a few milliseconds later, but potentially much later.) This strategy improves the SRM's performance.

### Note

Correctly formatted teardown requests always return a success response:

- ▶ If the session does not exist, it's effectively torn down already, so it's legitimate to say that the teardown has succeeded.
- ▶ If the session could not be torn down for some reason, that is beyond the control of the client, and is not the client's concern.
- ▶ Logically, a failed teardown response forbids the client to stop watching the stream. Clearly this makes no sense!

## 1.1.7 Partial teardown

When a session is torn down, it can happen that one or more of the external systems which initially contributed resources to that session, is not available.

When this happens, the SRM assumes that the device is still operational, and that the SRM has just lost contact with it temporarily. (This is the only safe assumption to make.) At this point, the SRM aborts the teardown process part-way through: resources which were torn down prior to the encounter with the unavailable device, are free, but the rest are still reserved.

When the SRM regains contact with the device, it performs a recovery process. As part of that process, the session teardown is resumed.

## 1.2 Resource management

Resource management is the process of allocating and deallocating the resources required by a streaming session in systems where this needs to be done explicitly.

### Note

Resource management is not concerned with actually delivering the resources—only allocating them appropriately. The process of contacting the various contributors to set up and tear down sessions is covered by [Session management](#) <sup>p.10</sup>.

The SRM can operate in two modes:

- ▶ A dedicated system has responsibility for bandwidth allocation, known as an ERM (Edge Resource Manager). The SRM models the set of devices serving a particular client as a Service Group, but it does not need to know about individual resources. Instead, it instructs the ERM to reserve bandwidth on the appropriate Service Group as part of the session management process. This is **dynamic bandwidth allocation**, the recommended mode.
- ▶ Alternatively, simple resource management can be performed by the SRM itself. The SRM models the operator's streaming infrastructure as a set of Service Group Resources associated with each Service Group. It reserves one Service Group Resource for each stream. This is **table-based bandwidth allocation**. Table-based bandwidth allocation is sometimes called **static bandwidth allocation**.

## 1.2.1 Table-based bandwidth allocation

This is a simple resource management algorithm provided by the SRM for use in systems which do not have dedicated ERM devices.

To use table-based bandwidth allocation, the operator ingests a fixed set of Service Groups and Service Group Resources into the SRM in a CSV file. Each Service Group corresponds to a single QAM output port; each Service Group Resource corresponds to a single program, on a single frequency. Effectively, set-top boxes will tune to the Service Group Resource.

### Supported features

- ▶ Each allocation may be for a different amount of bandwidth.
- ▶ Each allocation requires exactly one Service Group Resource, no matter how much bandwidth is reserved. (This is different from the legacy SRM facility provided by SDP, where high bandwidth sessions could require multiple resources.)

### Unsupported features and assumptions

- ▶ The SRM assumes that QAMs are invulnerable. It cannot detect QAM failures.
- ▶ No support for dynamic UDP port or program number remapping.
- ▶ The SRM cannot discover QAM data dynamically. Details of QAM infrastructure must be discovered ahead of time and provided in an ingest file.

## 1.3 A typical deployment

The diagram shows a small deployment; however the minimum supported deployment is three SRM nodes with matching MongoDB nodes; production deployments can vary but the MongoDB node count must be a multiple of 3.

Check your System architecture document for the details of your configuration.

The application servers are connected to the clients via load balancers, to provide resiliency.

Component	TCP Port	Protocol	Description
Nginx	80	HTTP	HUE requests from clients
SDP	8180	HTTP	<ul style="list-style-type: none"> <li>▶ Filtered HUE requests from clients</li> <li>▶ WS requests from OSS / BSS</li> <li>▶ WS requests from SRM</li> </ul>
SRM	5580	HTTP	<ul style="list-style-type: none"> <li>▶ REST requests from OSS / BSS</li> <li>▶ Web (HTML over HTTP) requests from human operators</li> </ul>
SRM	5544	RTSP	Session management requests from clients

Component	TCP Port	Protocol	Description
SRM	5541	RTSP	ANNOUNCE propagation for terminated sessions
MongoDB	27017	MongoDB	<ul style="list-style-type: none"> <li>▶ SRM database queries</li> <li>▶ Mongo cluster propagation</li> </ul>

**Caution!**

Although the SRM is shown connecting directly to the SDP, it is quite common for the SRM to connect via Nginx instead. This causes the load on the SDP, due to the SRM, to be counted for throttling purposes: without this, there is a risk that the combination of throttled load directly from clients, plus load from the SRM which is only throttled indirectly, could overload the SDP. The SDP's performance degrades in gracefully when it is overloaded.

**Note**

- ▶ Larger deployments may require more app servers, ERMs, QAMs and Streaming servers (video servers).
- ▶ All SDP instances within the deployment talk to the same Oracle schema. In the same way, all the MongoDB instances in the deployment belong to the same replica set.
- ▶ As shown, the clients must be able to talk directly to the streaming servers - this is necessary in order to do trick-play. Only the SRM talks to the ERMs.
- ▶ All SRM instances must be able to talk to all streaming servers and ERMs. The links between the App server 2 SRM and the ERM / Streamer have been omitted from the diagram for the sake of clarity.
- ▶ The diagram only shows the IP deployment. There is an RF network as well - every QAM output port (RF) is connected to a set of STBs.
- ▶ The SRM assumes that the load balancers will not support any kind of client-to-app-server affinity, and that the connection the client establishes to keep the session alive may be to a different app server than the connection used to establish the session in the first place. Because of this, when a session is terminated prematurely (by any actor other than the client itself), an RTSP ANNOUNCE method is propagated, from the SRM which initiated the termination, to all the SRMs. Whichever SRM is maintaining the session keepalive connection with the client, will send the ANNOUNCE to the client and then close the keepalive connection.

## 1.4 SRM and NGOD

The SRM supports some parts of the Comcast Next-Generation On-Demand (NGOD) architecture.

NGOD system component	Nagra system component	Interaction	Protocol
On Demand Client	STB	all	NGOD S1

NGOD system component	Nagra system component	Interaction	Protocol
Session Manager	SRM	all	various
On Demand Resource Manager	SRM	all	NGOD R2
	<p><b>Note</b></p> <p>An NGOD ODRM manages streaming servers.</p>		
Edge Resource Manager	<p>SRM for table-based bandwidth allocation.</p> <p>NGOD ERM for dynamic allocation (recommended)</p> <p><b>Note</b></p> <p>An ERM manages QAMs.</p>	all	NGOD S6
Purchase Server	SDP	<p>STB: browse and purchase</p> <p>STB: selection start</p> <p>SRM: playout authorization</p>	<p>HUE</p> <p>NGOD E2 over HUE</p> <p>Web Services</p>

## 2 User tasks

### 2.1 Deploy the SRM

To deploy the SRM component, you will need to perform these tasks.

1. [Install and configure MongoDB](#) <sup>p.16</sup>. Every SRM instance within a cluster must use the same MongoDB database (this may be replicated).
2. [Install the SRM](#) <sup>p.18</sup>.
3. Edit the SRM's configuration files. This is described in [Configuring the SRM](#) <sup>p.19</sup>.
4. [Start and stop the SRM](#) <sup>p.22</sup>.
5. [Ingest data](#) <sup>p.23</sup>. This can be done on one SRM instance, on behalf of the entire cluster.
6. Configure the load-balancer to forward client traffic to the SRM.

**Caution!**

Some deployments, which use the legacy SRM within the SDP, currently use an SRM proxy between the clients (STBs) and the legacy SRM itself.

This proxy must **not** be used with the SRM component.

### 2.2 Install and configure MongoDB

#### Retrieve Package

MongoDB servers require 64-bit Red Hat Enterprise Linux 5 (RHEL5).

**Warning!**

RHEL4 is not supported. 32-bit systems are not supported.

MongoDB server installation is RPM based. The RPM's are available by downloading the following software components from IST:

<http://ist.hq.k.grp/cgi-bin/WebObjects/ist.woa/wa/inspectRecord?entityName=HESoftwareComponentVersion&id=59417>

Always download the versions of this package from IST that is compatible with the software products you are trying to install.

#### Install the packages

Install the RPM using:

```
rpm -iv nagra-mongodb-<version>.x64_64.rpm
```



Where <version> is the version you downloaded from IST. E.G. To install version 2.4.3:

```
rpm -iv nagra-mongodb-2.4-STD0.e15.x86_64.rpm
```

## Start the service

To start the MongoDB database server, run:

```
service mongodb2.4 start
```

### Caution!

This may take considerable time to execute if it is the first time it has been run.

If this is the first time that a module which uses MongoDB has been used, the initial starting database will not exist. MongoDB will not be useable in this state. You must Initialize your MongoDB instance, to build the initial database.

Only do this once, on first install of the RPM.

## Initialize Database

Please only follow **one** of these two options.

### Standalone MongoDB instance (Option 1)

First ensure that MongoDB has been started on the server.

To create a standalone MongoDB server, run this command (note replace <IP> with the external IP address of the MongoDB server):

```
service mongodb2.4 createCluster <IP>
```

### Caution!

It takes many minutes to create the initial database.

### Clustered MongoDB instance (Option 2)

To create a MongoDB cluster across multiple nodes, first ensure that MongoDB has been started on all the cluster nodes.

Secondly, run this on only **ONE** of the cluster nodes (where <IP> list, is the list of IP addresses of all the MongoDB servers in your cluster):

```
service mongodb2.4 createCluster <IP> <IP> <IP> <IP>
```

### Caution!

It takes many minutes to create the initial database.

## Service check

You can run the MongoDB service check command, by running:

```
service mongodb2.4 status
```

**Caution!**

This command calls the `rs.conf()` function of MongoDB, that returns a JSON structure with the status of your cluster. Please refer to the MongoDB website for information on the output of the `rs.conf()` function.

## 2.3 Install the SRM

1. If the SRM is to be installed on the same machine as the SDP, follow the instructions to [Share an application server with the SDP](#) <sup>p.18</sup> and skip the rest of this step.

If the SRM is to be installed on its own machine, obtain and install the JRE from the latest available Sun/Oracle Java 7:

```
rpm -Uhv jre-7u21-linux-x64.rpm
```

The "7u21" is just an example. If a later version is available, use that.

2. Install the container:

```
rpm -Uhv srm-tomcat-7.0.35-2.noarch.rpm
```

Use the RPM version which applies to your release - 7.0.35-2 is just an example.

3. Install the SRM itself:

```
rpm -Uhv srm-1.0-STD0.noarch.rpm
```

Use the RPM version which applies to your release - 1.0-STD0 is just an example.

### 2.3.1 Share an application server with the SDP

The SRM and the SDP can readily co-exist, with one exception: they require different versions of the JRE (Java Runtime Environment).

The SDP is smart enough to look for the latest installed Java6 JRE when starting. Similarly, the SRM is smart enough to look for the latest installed Java7 JRE. However, if care is not taken, the RPM package manager may replace the Java6 JRE, which will cause SDP to stop working.

Follow these instructions to get the two JREs working side by side.

1. Install the SDP first, following the regular deployment procedure.
2. Obtain the JRE from the latest available Sun/Oracle Java 7 and install it with

```
rpm -ihv --replacefiles jre-7u21-linux-x64.rpm
```

**Caution:**

**Warning!** Do NOT use `rpm -Uhv` to install the Java7 JRE for the SRM.

**Note** 7u21 is just an example. If a later version is available, use that.

3. [Install the SRM](#) <sup>p.18</sup> as normal, starting at step 2 rather than step 1.
4. Proceed with the rest of the SRM deployment as normal.

**Tip**

To verify that both JREs are installed, use

```
rpm -qi --all jre
```

The SDP installer's "Check status of installed RPMs" doesn't do this, so you may get the Java7 JRE being reported in this case, rather than the Java6 one. This is not an issue if the command above reports both JREs.

## 2.4 Configuring the SRM

The SRM is configured by setting various **properties** before the SRM is started. These properties are grouped into **beans**, which are defined in XML files.

**Note** The "beans" are called this because they are "Spring beans" - part of the <http://static.springsource.org/spring/docs/3.2.x/spring-framework-reference/html/beans.html#beans-definition>.

For each file you need to change:

1. Take a backup copy of the file.
2. Edit the file with a plain-text editor (eg. `vi`) and change the properties as required.

You must configure at least the properties listed here. For unusual configurations, you may need to change some of the other configuration properties as well. See the reference documentation for details.

**Note** [Configure logging](#) <sup>p.20</sup> is configured separately.

**Warning!** It is your responsibility to ensure that you do not corrupt the configuration files. The SRM cannot check the format or the values you enter.

If values are invalid (for example, if you entered an alphabetic value for an integer field), then the SRM's behavior is undefined. Although it is likely that the SRM will not start correctly, it may fail in unexpected ways.

## Essential configuration

All SRM deployments will need to configure these properties in `/opt/srm/webapps/srm-deployer/WEB-INF/srm-conf.xml`

Bean	Property	Notes
<a href="#">Configuration bean: srmCoreConfig</a> <sup>p.48</sup>	bandwidthMap	
<a href="#">Configuration bean: srmCoreConfig</a> <sup>p.48</sup>	sdpGatewayPrimaryPavDevice	Only required if SRM is not sharing a machine with the SDP
<a href="#">Configuration bean: srmCoreConfig</a> <sup>p.48</sup>	sdpGatewayUsername	
<a href="#">Configuration bean: srmCoreConfig</a> <sup>p.48</sup>	sdpGatewayPassword	
<a href="#">Configuration bean: srmCoreConfig</a> <sup>p.48</sup>	outboundAnnounceCallbackUrl	
<a href="#">Configuration bean: vsConfig</a> <sup>p.58</sup>	effectiveSrmlpAddress	

### 2.4.1 Configure logging

Currently logging is configured by manually editing a configuration file. (Unlike the SDP, there is currently no UI or tool for this).

1. Take a backup copy of the file `/opt/srm/webapps/srm-deployer/WEB-INF/log4j.xml`
2. Use a plain text editor (eg. `vi`), to edit this file.

This is a standard log4j configuration file. Please consult the log4j documentation for more details, eg. <http://logging.apache.org/log4j/1.2/manual.html>

**Warning!** It is your responsibility to ensure that you do not corrupt this file - the SRM cannot check it for you.

### 2.4.2 Default logging configuration

#### Default appenders

By default, the following loggers are defined.

Both log files are located in directory `/opt/srm/webapps/srm-deployer/log`.

Appender / file	Default level	Purpose
usage.log	INFO	Called on entry to every low-level service within the SRM. This is useful for tracing the flow of control.
server.log	INFO	Debug and error logging. Usage entries are also included.

**Available levels**

The following logging levels are supported.

Level	For what
ERROR	The request has failed. Depending on the severity of the failure, future requests may also be impacted.
WARN	An anomalous or unexpected condition has been detected. This has not caused a failure directly, but investigation is required.
INFO	Routine status / progress update. No further action is required.
DEBUG	For troubleshooting. Debug logging is highly verbose, and should be suppressed in normal operation.

**2.4.3 Default users and passwords**

Most of the HTTP UI requires authentication (username / password). The out-of-the-box values are as follows.

Username	Password	Purpose
operations	argan_\$tdq24	Human operator
nagra-agent	2856r4_\$tdq24	Another machine, eg. monitoring application

You can change these passwords by editing the file

`/opt/srm/conf/tomcat-users.xml`

**Warning!**

You are strongly advised to change these passwords on a production system!

**Note**

The UI uses HTTP BASIC authentication. Your browser should negotiate this authentication mode automatically.

## 2.5 Start and stop the SRM

The SRM is deployed in a Web "Servlet" Containers and WARs; you start and stop the container, and the container starts and stops the SRM itself.

The container is started and stopped using a regular init script. You must be logged in as root to do this.

To start, use

```
service srm start
```

To stop, use

```
service srm stop
```

### 2.5.1 Shut down an SRM cleanly

1. Instruct the load balancer to stop forwarding connections to the SRM. (Leave existing connections alone.)

**Note**

On our standard Barracuda load balancers, this can be done by setting the SRM service to MAINTENANCE mode.

2. Wait for all active connections on the SRM to close.

**Caution!**

**This may take several hours.**

VOD sessions have a defined lease period. After the lease ends, the session will be terminated even if the client wants to keep it open. Therefore this is guaranteed to happen eventually.

The lease period can be configured in [Configuration bean: srmCoreConfig](#) <sup>p.48</sup>

To discover how many clients are still connected, you can query the [Monitored connection manager](#) <sup>p.38</sup> via its REST interface.

3. As root, stop the SRM service:

```
service srm stop
```

## 2.6 Ingest data

The SRM's view of the deployment infrastructure is provided in ingest files.

You must ingest at least one Sts record, one Pav record, one SrmServer record, and one ServiceGroup record before attempting to start any sessions.

If you are using [Table-based bandwidth allocation](#) <sup>p.13</sup>, you must also ingest at least one ServiceGroupResource record. Otherwise, you must also ingest at least one Erm record.

1. Prepare the data, following the [Data ingest file format](#) <sup>p.63</sup>.

**Tip**

The easiest way to do this is usually to use a spreadsheet application.

2. Check that the data obeys the [Constraints on data](#) <sup>p.23</sup> enforced by SRM.

**Caution!**

Data you ingest must obey these constraints, or the file will be rejected.

3. Use [The SRM's Web UI](#) to navigate to the "SRM data ingest" page.
4. Use the web form to upload the data file from step 1.
5. Look at the log which is returned after the upload, to determine the result of the ingest.

**Note**

You may call the [Bulk data ingest](#) <sup>p.82</sup> API directly, if you wish, rather than going via the UI. This is an advanced technique.

### 2.6.1 Constraints on data

The SRM imposes some constraints on the data it holds about devices, to maintain lifecycle integrity.

#### Contributor devices

A [Contributor device](#) <sup>p.10</sup> device cannot be disabled if it is still in use. This constraint therefore applies to Erm and Sts devices.

In detail:

- ▶ A contributor device cannot be deleted when its status is ACTIVE.
- ▶ A contributor device cannot be deleted if there are still active sessions on the device.
- ▶ A contributor device cannot be set from ACTIVE to DISABLED directly.
- ▶ A contributor device cannot be set from INACTIVE to DISABLED if there are still active sessions on the device.

If you need to delete or disable such a device while it is active, you will need to perform two separate operations.

1. Ingest a file setting the device's status to **INACTIVE**.

The SRM will no longer allocate resources from this device.

2. Wait for all existing sessions using the device to finish.
3. Ingest a file deleting or disabling the device as required.

The SRM will update the records.

## AppServer and Pav devices

These devices cannot be set to INACTIVE. They must be either ACTIVE or DISABLED.

AppServer and Pav devices do not store information about sessions, so you can disable them at any time.

## ServiceGroups and ServiceGroupResources

The SRM enforces these restrictions on updates to ServiceGroups and ServiceGroupResources:

- ▶ A ServiceGroup cannot be updated if there are active sessions using it.
- ▶ A ServiceGroupResource cannot be updated if its ServiceGroup is active.

This minimizes the risk of updates interfering with active sessions.

To enable operators to make changes to ServiceGroups which are in use, the SRM provides an additional **Deactivate** operation for ServiceGroups. This operation sets the ServiceGroup to inactive, but does not change anything else. When a ServiceGroup is inactive, no new sessions will be permitted to use it, but any sessions which are using it already are left alone.

To update ServiceGroups or their ServiceGroupResources safely:

1. Ingest a file specifying **Deactivate** operations on the ServiceGroups you want to update.  
The SRM will no longer allocate sessions on this ServiceGroup.
2. Wait for all existing sessions on the ServiceGroup to finish.
3. Ingest a file with the required updates.

The SRM will update the records. If you need the ServiceGroups to be reactivated, specify a **status** of **ACTIVE** in each relevant record.

### Caution!

Do not attempt to deactivate ServiceGroups by setting their **status** directly to **INACTIVE**.

## 2.7 Migrate from a legacy SRM

### Warning!

This section may require further review from support engineering and the region. You must perform a successful dry run in a lab before attempting this in production!

## Prepare for migration

This work can be done ahead of time, without affecting operations. No SDP update is required, provided that a suitably modern SDP is in place (3.7 or better).



1. Ensure that (new) ERMs and (existing) streaming servers are **not** reachable from the new SRM.

**Note**

This is necessary because, if the SRM can reach these devices, it will synchronize with them, and tell them to drop the existing sessions (which the new SRM doesn't know about).

2. Install and prepare ERMs.

**Caution!**

As part of this process, the ERMs will be told about the QAMs which they will be managing. However, at this point, these QAMs are carrying active sessions managed by the legacy SRM. Don't let the ERM tell the QAMs to drop these sessions!

It may be necessary to carry out this process with the ERMs isolated from the network.

3. Install and configure MongoDB on app servers.
4. Install, configure, and start new SRM on app servers. (This does not include ingesting any data).
5. Migrate existing data (optional).

There are various in-house migration tools which allows applicable legacy data to be transferred from the SDP's database, directly into the SRM's database. The following entities can be transferred:

- Devices: Sts, Erm.
- DeviceGroups.
- ServiceGroups.
- ServiceGroupResources. Only ServiceGroupResources for ServiceGroups which use table-based bandwidth allocation will be transferred; any others are not wanted.

This script is not shipped as part of the regular SRM distribution. If you have a need to migrate data in this way, please contact Nagra support.

**Caution!**

If you choose this migration option, make sure you keep some kind of database dump available for backup purposes. While SRM is active, it is not possible to obtain a consistent backup of the Mongo DB database, and we do not recommend that you even bother.

**Tip**

Depending upon how you created the legacy data in the first place, it may be easier (especially when considering backup) to convert the original source data to a compatible CSV, and ingest that afresh rather than attempting to migrate.

6. Prepare new data. Create a CSV containing the following:

- SrmServers.
- Pavs.
- If you didn't migrate existing data, the devices, DeviceGroups, ServiceGroups and ServiceGroupResources (only for ServiceGroups which use table-based bandwidth allocation, if any).

**Tip**

For larger deployments, it may be convenient to split the total dataset out over several CSVs. There are two reasonable ways to do such a split:

- Typed partition: Ingest all devices and DeviceGroups, then all the ServiceGroups, then all the ServiceGroupResources (table-based allocation only).
- Zoned partition: Ingest all SrmServers and Pavs. Then ingest multiple "zones". Each zone contains a self-contained set of DeviceGroups, Erms, Sts, ServiceGroups and (if table-based) ServiceGroupResources.

7. Ingest prepared SRM data into the new SRM. If you have a cluster of SRM instances (recommended), only do this on one of them - the DB is shared between them all.

## Perform migration

This does disrupt operations and should be done during a period when minimal load is expected.

1. Disconnect the old SRM port from the STBs at the load balancer.
2. Mark all active sessions as closed in the SDP.
3. Shut down all sessions on the streaming servers.
4. De-isolate ERMs from QAMs. Verify that ERMs have taken control of the QAMs.
5. De-isolate SRM from ERMs and streaming servers. Verify that the SRM has opened connections to these devices.
6. Connect the new SRM to the STBs at the load balancer.
7. Verify that new SRM begins to serve sessions.

### 2.7.1 SRM Typical Manual Migration

#### Software Components

It is sometimes the case, when upgrading the SDP, the a full MediaLive upgrade will not be performed. When this is the case the configuration and installation will not necessarily be managed by Foreman and Puppet.

This section will detail the manually installation process required.

#### Note

As a pre-requisite a MongoDB instance also needs to be installed. Either MongoDB 2.4 or 3.0 is supported. The recommendation is 3.0

Firstly, all of the RPMs should be downloaded from IST.

Software Component	Purpose	Details
Nginx	HTTP Routing	Nginx is not supplied as part of the module release. It is recommended that the latest applicable HTTP Router is included with the customer

Software Component	Purpose	Details
		release, and that this should be installed.
JRE 1.7	Java Runtime	This is a pre-requisite for installation of the other rpms. Installation is part of the MediaLive OS when performing an install onto a Media Live platform.
Tomcat	Servlet Container	The SRM requires a servlet container and Apache Tomcat is used for this purpose. The application users (operations, srm-ingest ...) are created as part of the tomcat installation.
SRM Nginx Conf	Nginx config for the SRM	The SRM nginx conf is required for access to OTT/HTTP support within the SRM
SRM Adapter Nginx Conf	Nginx config for the SRM Adapter	The adapter config is required for re-routing specific calls, originally destined for the SDP, to the SRM Adapter
SRM Deployer	This is the main component of the SRM	The SRM
SRM Adapter	An adapter facade for the SRM	<p>The adapter is used to provide support for legacy APIs within the SDP that should not architecturally reside within the SRM.</p> <p>At this stage specifically the HTTP pre-integrity SETUP call. The adapter receives this HTTP request and converts to a suitable RTSP request for processing by the SRM.</p>
SRM-MLDS	Puppet	The SRM deployment files requires for the SRM to be managed by puppet/foreman

Software Component	Purpose	Details
SRM-MLM	Monitoring	MediaLive Monitoring (Nagios) configuration.

## Installation

Simply install each of the downloaded RPMs in the order specified above. For a non-ML deployment, SRM-MLDS and SRM-MLM may not be required.

Further information on this procedure is provided in the [Install the SRM](#) <sup>p.18</sup> section.

### Note

If the SRM is installed onto the same server as other modules, such as SDP, it may be necessary to force install the JRE - so that it can co-exist with an earlier version.

```
rpm -ivfh --force
```

## SRM Ports

The SRM and legacy SRM are configured to run on separate ports.

RTSP requests are handled via port 5544. In order to have a seamless migration, either the SDP access point address should be altered - to reference port 5544 instead of the existing 8184. Alternatively the SRM can be re-configured via [Configuration bean: srmCoreConfig](#) <sup>p.48</sup>.

## Starting the SRM

```
service nginx restart
service srm restart
service srm-adapter restart
```

## Data Migration from SDP

A utility has been created which will connect to an existing SDP database and export the data into a csv format suitable for ingest into the SRM.

### Caution!

The SRM needs to be started and operational during the migration process.

This includes Video-Servers, Service-Groups, etc.

```
>./ runme.sh
Please specify your service (e.g. qube:world): qube
Please specify your schema (without _o, _u, _m): ncqsp2prod
Please enter the IP address of your master SRM instance: 127.0.1
```

```
The script will not run the data extraction and the ingest process, please be patient.
You have specified the SERVICE: qube
You have specified the SCHEMA: ncqsp2prod
You have specified the HOST: 127.0.0.1
If this information is correct please enter Yes
Yes
```

## Configuration of SRM-Clients

Post installation only a handful of generic clients are configured. These are for standard OTV and OTT support.

Generally a customer will require additional clients to be configured, and these can be configured via an SRM ingest.

### 2.7.2 SrmClient Ingest Samples

#### Full Sample:

**Note**

```
SrmClient,overrideName,UserAgent,ClabSessionGroup,KeepAlive,KeepAliveTimeoutSecs,KeepAlive
Misses,RemoteAddress,RemotePort,SessionType,SessionVariant,Description,Require,CheckEntitlement,
OverrideSessionVariantNC_UGC,NC-UGC,,FALSE,,,,,VOD,NCVOD,NC VOD,,FALSE,RESUMEVODNC_
VOD,NC-VOD,,FALSE,,,,,VOD,NCVOD,NC VOD,,FALSE,NC_OTV,OpenTV VOD 1,,FALSE,,,,,VOD,
OTV,NC VOD,,TRUE,NC_SDB,OpenTV SDV 1,,TRUE,30,2,,,SDB,NCOTV,NC SDV,,FALSE,BYTEL,
Thomson,SMARTVISION_CRM_GROUP,FALSE,,,,,BANDWIDTH_ONLY,CLABS,,com.cablelabs.ermi,
FALSE,PLAYCAST,Playcast,PLAYCAST_CRM_GROUP,TRUE,30,2,,,BANDWIDTH_ONLY,CLABS,,com.
cablelabs.ermi,FALSE,
```

#### NC\_VOD

**Note**

```
SrmClient,override
Name,UserAgent,ClabSessionGroup,KeepAlive,KeepAliveTimeoutSecs,KeepAliveMisses,Remote
Address,RemotePort,SessionType,SessionVariant,Description,Require
NC_VOD,NC-VOD,,FALSE,,,,,VOD,NCVOD,,FALSE
```

#### NC\_OTV

**Note**

```
SrmClient,override
Name,UserAgent,ClabSessionGroup,KeepAlive,KeepAliveTimeoutSecs,KeepAliveMisses,Remote
Address,RemotePort,SessionType,SessionVariant,Description,Require
NC_OTV, OpenTV VOD 1, , FALSE, , , , VOD, OTV, NC VOD,
```

## NC\_SDB

**Note**

SrmClient,override

Name,UserAgent,ClabSessionGroup,KeepAlive,KeepAliveTimeoutSecs,KeepAliveMisses,Remote Address,RemotePort,SessionType,SessionVariant,Description,Require

NC\_SDB, OpenTV SDV 1, , TRUE, 30, 2, , , SDB, NCOTV,NC SDV ,

## BYTEL

**Note**

SrmClient,override

Name,UserAgent,ClabSessionGroup,KeepAlive,KeepAliveTimeoutSecs,KeepAliveMisses,Remote Address,RemotePort,SessionType,SessionVariant,Description,Require

BYTEL, Thomson, SMARTVISION\_CRM\_GROUP, FALSE, , , , BANDWIDTH\_ONLY, CLABS, , com.cablelabs.ermi

## PLAYCAST

**Note**

SrmClient,override

Name,UserAgent,ClabSessionGroup,KeepAlive,KeepAliveTimeoutSecs,KeepAliveMisses,Remote Address,RemotePort,SessionType,SessionVariant,Description,Require

PLAYCAST, Playcast, PLAYCAST\_CRM\_GROUP, TRUE, 30, 2, , , BANDWIDTH\_ONLY, CLABS, , com.cablelabs.ermi

## NAGRA\_OTT

**Note**

SrmClient,override

Name,UserAgent,ClabSessionGroup,KeepAlive,KeepAliveTimeoutSecs,KeepAliveMisses,Remote Address,RemotePort,SessionType,SessionVariant,Description,Require

NAGRA\_OTT, , , TRUE, 60, , , , VOD, NAGRA\_OTT, ,

## GDC\_NGOD

**Note**

SrmClient,override

Name,UserAgent,ClabSessionGroup,KeepAlive,KeepAliveTimeoutSecs,KeepAliveMisses,Remote Address,RemotePort,SessionType,SessionVariant,Description,Require

GDC\_NGOD, , , TRUE, 60, 3, 127.0.0.1, 5541, VOD, NGOD\_S1, , com.comcast.ngod.s1,

## 2.8 SRM User Session Limitation

### Overview

User Session Limitation is basically a feature that allows an operator to limit the number of OTT user sessions on the SRM side. This means that an operator can limit the number of sessions given to users that belong to a particular account.

This API is featured when a client sends an OTT SETUP request to the SRM, in order to setup a session to watch a particular content.

### Client-to-SRM Protocol

The protocol is REST. The request type is POST.

### Overall Flow Diagram

Clients should follow this use case to:

- ▶ Setup user session limitation on SrmClient
- ▶ SETUP an OTT session on the SRM

@startumlactor Userparticipant Operatorparticipant User1participant NGINXparticipant SRMparticipant PAVgroup Scenario1: UserSessionLimitation is enabled and within limit on SRM + within limit on PAV Operator -> SRM: Ingest SrmClient Operator <- SRM: 200 OK User1 -> NGINX: [REST] SETUP (token) NGINX -> SRM: [REST] SETUP (userId, acctId, devId) SRM -> SRM: [Internal] check UserSessionLimit SRM -> PAV: [SOAP] checkPavOtt (contentId, userId) SRM <- PAV: authorized NGINX <- SRM: SRM sessionId User1 <- NGINX: SRM sessionIdendgroup Scenario2: UserSessionLimitation is enabled and exceeds limit on SRM + within limit on PAV Operator -> SRM: Ingest SrmClient Operator <- SRM: 200 OK User1 -> NGINX: [REST] SETUP (token) NGINX -> SRM: [REST] SETUP (userId, acctId, devId) SRM -> SRM: [Internal] check UserSessionLimit SRM -> PAV: [SOAP] checkPavOtt (contentId, userId) SRM <- PAV: notAuthorized NGINX <- SRM: Error is returned (200 OK, SRM-009) User1 <- NGINX: Error is returned (200 OK, SRM-009)endgroup Scenario3: UserSessionLimitation is disabled on SRM + within limit on PAV Operator -> SRM: Ingest SrmClient Operator <- SRM: 200 OK User1 -> NGINX: [REST] SETUP (token) NGINX -> SRM: [REST] SETUP (userId, acctId, devId) SRM -> SRM: [Internal] check UserSessionLimit SRM -> PAV: [SOAP] checkPavOtt (contentId, userId) SRM <- PAV: authorized NGINX <- SRM: SRM sessionId User1 <- NGINX: SRM sessionIdend@enduml

Reference	<TBD>
Main actor	SRM
Secondary actors	PAV, Client application, NGINX, Operator
Pre-Conditions	- SRM and PAV are running and operational

- Operator ingested SrmClient into SRM with correct Client data.

Trigger

Client sends a SETUP, PLAY or DESCRIBE RTSP request to the SRM

## SETUP Steps

1. Operator ingests SrmClient into SRM with correct Client data User Session Limitation details.
2. SRM validates the ingest data and send an OK response to the Operator.
3. User, with a token and subscription, send an OTT POST SETUP request to the SRM to get a session ID to watch a content using API [Session Setup](#) <sup>p.120</sup>
4. SRM checks if UserSessionLimit is enabled and that the user does not exceed the limit setup by the operator (step 1) in the SrmClient.
5. SRM sends a request to the PAV to check that this user is authorised to watch the content using API [PAV interface](#)
6. PAV checks if user is authorised to watch content and returns authorised to SRM.
7. SRM creates a session and returns the session id to the Client.

## SETUP Extensions

Ref	From step	Description
<b>A</b>	2	If ingestion fails, SRM returns an error to the Operator.
<b>B</b>	4	If UserSessionLimitation is disabled on SRM, SRM continues by sending an authorisation request to the PAV.
<b>C</b>	4	If UserSessionLimitation exceeds the limit on SRM, SRM returns an error to the Client.
<b>D</b>	6	If user is not authorised on the PAV, and error is returned to the SRM. Subsequently, SRM returns an error to the Client.



## 3 Monitoring

The SRM exposes health monitoring information via a set of REST APIs. It is compatible with the MediaLive Monitoring system, API version 1.1. It can also be used with the SDPMon tool.

### 3.1 Monitored quantity

A **monitored quantity** is a numeric value which varies over time. The SRM samples the quantity, and exposes a [Monitored histogram](#) <sup>p.34</sup> which summarizes the sample population. The histogram is exposed as a JSON structure.

Every quantity has a name. A given type of [Monitored object](#) <sup>p.35</sup> exposes a fixed set of quantities.

The histograms for each quantity are taken, and made available, over several [Monitored viewing periods](#) <sup>p.33</sup>.

### 3.2 Monitored viewing periods

The SRM exposes three sample populations for each <https://atlassian.hq.k.grp/confluence/display/SRM/Monitored+quantity>:

- ▶ last minute
- ▶ last hour
- ▶ last day

The "last minute" is based on calendar / "wall clock" time, and exposes the most recently completed minute (not the one currently in progress). However, the "last hour" is for a rolling 60-minute period. Similarly the "last day" is a rolling 24-hour period, where each of the hours is a calendar hour.

As a concrete example, a call at **2013-06-17 03:31:14** would return:

Viewing period	Earliest sample time	Latest sample time
lastMinute	2013-06-17 03:30:00.000	2013-06-17 03:30:59.999
lastHour	2013-06-17 02:31:00.000	2013-06-17 03:30:59.999
lastDay	2013-06-16 03:00:00.000	2013-06-17 02:59:59.999

A fixed timezone is used. This is configurable. Typically the local timezone is used.

**Note**

Quantities are recorded for statistical purposes. Where a sample is taken very close to the edge of an interval, the SRM does not guarantee whether it will be recorded in one such interval or the next. However, it does guarantee that it will be recorded exactly once - it won't be missed out, and it won't be counted twice.

The timezone is required in order to establish when the calendar hour rolls over. Timezones exist which are offset from UTC by a non-integer number of hours. Eg. Australian Central Standard Time is UTC +09:30.

### 3.3 Monitored histogram

The values taken by a [Monitored quantity](#)<sup>p.33</sup> are exposed as a histogram. Each bin / column of the histogram is assigned a name and an upper limit (but see note) by the operator; this set of options defines a **template** for the histogram. The number of bins is also configurable by the operator.

**Note**

- ▶ The lower limit of the first (leftmost / lowest value) bin is effectively Long.MIN\_VALUE.
- ▶ The lower limit of subsequent bins is one more than the upper limit of its predecessor (by definition of a histogram).
- ▶ The upper limit of the last (rightmost / highest value) bin is effectively Long.MAX\_VALUE; whatever value provided by the operator is overridden.

Typically the names will refer to warning levels in the monitoring system. For example, the template for a Pav device might have:

Name	Upper limit
fast	50
normal	150
slow	400
warn	1000
critical	9999 (dummy value - see above)

**Note**

The unit for the values is defined by the [Monitored quantity](#)<sup>p.33</sup> itself. In the example, they're in ms.

Finally, there are six more attributes for the quantity, which always appear:

Name	Description
successCounter	The number of operations which were successful.

Name	Description
	For quantities which don't define success or failure, all samples are assumed to be successful, ie. successCounter == totalCounter.
failedCounter	The number of operations which failed due a fault on the system. For quantities which don't define success or failure, all samples are assumed to be successful, ie. failedCounter == 0.
rejectionCounter	The number of operations which were rejected for business reasons: authorization declined, lack of resources, or an invalid client request.
totalCounter	The total number of operations. Always successCounter + rejectionCounter + failedCounter
min<quantity>	The minimum sample value
max<quantity>	The maximum sample value

**Note**

<quantity> depends on what the quantity is - Time, Threads, Connections, and so on. See the entries for individual quantities, and the [Monitoring statistics](#) <sup>p.99</sup> API examples.

"failedCounter" is so named (where "failureCounter" might have been expected) for historical reasons.

### 3.4 Monitored object

The SRM monitors a set of objects. It exposes [Monitored quantity](#) <sup>p.33</sup> and health information for these objects.

There are several different types of monitored object.

- ▶ [Contributor device](#) <sup>p.10</sup> (several)
- ▶ [Pav device](#) <sup>p.11</sup> (one now, several later)
- ▶ [SrmServer device](#) <sup>p.11</sup> (several)
- ▶ [Monitored interface](#) <sup>p.36</sup> (exactly two)
- ▶ [Monitored session manager](#) <sup>p.37</sup> (exactly one)
- ▶ [Monitored connection manager](#) <sup>p.38</sup> (exactly one)
- ▶ [Monitored thread manager](#) <sup>p.39</sup> (exactly one)

### 3.4.1 Monitored quantities of contributor devices

Contributor devices expose a [Monitored quantity](#) <sup>p.33</sup> for each of the two flows:

Name	Description
setupExecution	How long it took to set up the session on the device, and the success rate.
teardownExecution	How long it took to tear down the session on the device, and the success rate.

**Note** | All execution times are in milliseconds.

### 3.4.2 Monitored quantities of Pav devices

Pav devices expose one [Monitored quantity](#) <sup>p.33</sup>.

Name	Description
pavExecution	Request processing times and success rate for the authorization check. Times in milliseconds.

### 3.4.3 Monitored interface

A monitored interface represents an incoming interface to the SRM.

There are exactly two monitored interfaces.

Name	Description
inboundCustomerRtsp	Inbound connections from end-user clients. This is usually the most interesting interface.
inboundManagementRtsp	Inter-SRM notifications. These occur when sessions are terminated administratively, or because of a failure reported by a contributor device.

Each interface exposes a [Monitored quantity](#) <sup>p.33</sup> for each of several items of interest.

**Note** | All execution times are in milliseconds.

Name	Description
<code>setupExecution</code>	Request processing times and success rate for session setups.
<code>teardownExecution</code>	Request processing times and success rate for session teardowns.
	<p><b>Note</b>   Teardown operations occur asynchronously, and the request returns quickly (unless the SRM is heavily overloaded). This figure does not include the time taken by the teardown operation itself.</p>
<code>pingLightExecution</code>	Request processing times and success rate for session keepalives, other than the first keepalive for a given session. These are handled using a fast-track process which reduces load on the SRM.
<code>pingHeavyExecution</code>	Request processing times and success rate for session keepalives, each of which is the first keepalive for a given session. These cannot be fast-tracked.
<code>announceExecution</code>	Request processing times and success rate for propagated session failure announcements.

### 3.4.4 Monitored session manager

The session manager exposes two [Monitored quantity](#) <sup>p.33</sup> for some items of interest, which do not fall into any other category.

The object-name of the session manager, as used in the [Monitoring statistics](#) <sup>p.99</sup> API, is `sessionManager`.

The quantities are as follows.

Name	Description
<code>vodSessionCount</code>	How many VOD sessions are active.
	<p><b>Note</b>   The SRM polls the session count once every second. These samples are the population which is summarized by the histogram.</p> <p>This count includes VOD with BTV-based content - SO, CU, TS.</p>

Name	Description
	In the API response, the range is represented by fields named <b>minSessions</b> and <b>maxSessions</b> , not <b>minTime</b> and <b>maxTime</b> as shown in the API example.
<b>teardownBacklogSize</b>	How many sessions are waiting to be torn down asynchronously.
	<b>Note</b> The SRM polls the queue size every second. These samples are the population which is converted into a histogram. In the API response, the range is represented by fields named <b>minJobs</b> and <b>maxJobs</b> , not <b>minTime</b> and <b>maxTime</b> as shown in the API example.

### 3.4.5 Monitored connection manager

The monitored connection pool represents the state of inbound connections to the SRM. It does not include connections to the REST interface.

The object-name of the connection pool, as used in the <https://atlassian.hq.k.grp/confluence/display/SRM/Monitoring+statistics> API, is **connectionManager**.

The quantities are as follows.

Name	Description
<b>customerConnectionCount</b>	How many connections are currently open from clients.
	<b>Note</b> This includes a logical "connection" which represents the mechanism which listens for incoming requests. Therefore in normal use the connectionCount will be 1 even when no clients are connected. In the API response, the range is represented by fields named <b>minConnections</b> and <b>maxConnections</b> , not <b>minTime</b> and <b>maxTime</b> as shown in the API example.
<b>throttlePermitCount</b>	How many more concurrent requests, excluding fast-track keepalives, can be supported.

Name	Description
	<p><b>Note</b></p> <p>The first ping for every connection is handled like a regular request. However, subsequent pings are handled using a "fast track" process which uses less resources.</p> <p>In the API response, the range is represented by fields named <b>minOperations</b> and <b>maxOperations</b>, not <b>minTime</b> and <b>maxTime</b> as shown in the API example.</p> <p>Note that this has the opposite sense to most histograms: it idles at maximum, and drops to zero when the SRM is flooded.</p>

### 3.4.6 Monitored thread manager

The thread manager manages the various thread pools which are used internally.

The object-name of the thread manager, as used in the <https://atlassian.hq.k.grp/confluence/display/SRM/Monitoring+statistics> API, is **threadManager**.

The quantities are as follows.

Name	Description
ioThreadCount	<p>Count of threads being used to do socket I/O. This includes accepting new connections. These threads are shared between the management <a href="https://atlassian.hq.k.grp/confluence/display/SRM/Monitored+interface">https://atlassian.hq.k.grp/confluence/display/SRM/Monitored+interface</a> and both halves (TCP and UDP) of the customer-facing <a href="https://atlassian.hq.k.grp/confluence/display/SRM/Monitored+interface">https://atlassian.hq.k.grp/confluence/display/SRM/Monitored+interface</a>.</p> <p><b>Note</b></p> <p>The SRM polls the thread count once every second. These samples are the population which is converted into a histogram.</p> <p>This information is not available to lower-level monitors such as the JVM monitor. The JVM monitor can see how many threads are in use, but not what they are being used for.</p> <p>This thread pool is unbounded - it will grow as required, and shrink again once the spike has passed. (It may take around 1 minute before the pool shrinks.)</p> <p>The thread count is typically around 32 threads when the SRM is idle.</p> <p>In the API response, the range is represented by fields named <b>minThreads</b> and <b>maxThreads</b>, not <b>minTime</b> and <b>maxTime</b> as shown in the API example.</p>

Name	Description
<code>customerLogicThreadCount</code>	<p>Count of threads being used to perform customer requests. These threads are shared between both halves (TCP and UDP) of the customer-facing <a href="https://atlassian.hq.k.grp/confluence/display/SRM/Monitored+interface">https://atlassian.hq.k.grp/confluence/display/SRM/Monitored+interface</a>.</p> <p><b>Note</b></p> <p>The SRM polls the thread count once every second. These samples are the population which is converted into a histogram.</p> <p>This information is not available to lower-level monitors such as the JVM monitor. The JVM monitor can see how many threads are in use, but not what they are being used for.</p> <p>This thread pool is unbounded - it will grow as required, and shrink again once the spike has passed. (It may take around 1 minute before the pool shrinks.)</p> <p>In the API response, the range is represented by fields named <code>minThreads</code> and <code>maxThreads</code>, not <code>minTime</code> and <code>maxTime</code> as shown in the API example.</p>
<code>managementLogicThreadCount</code>	<p>Count of threads being used to perform management requests.</p> <p><b>Note</b></p> <p>The SRM polls the thread count once every second. These samples are the population which is converted into a histogram.</p> <p>This information is not available to lower-level monitors such as the JVM monitor. The JVM monitor can see how many threads are in use, but not what they are being used for.</p> <p>In the API response, the range is represented by fields named <code>minThreads</code> and <code>maxThreads</code>, not <code>minTime</code> and <code>maxTime</code> as shown in the API example.</p>

### 3.5 Monitored device health status

Monitored objects which are devices have a health status which can be monitored. The health status fields are as follows.

Name	Description
<code>id</code>	A unique ID in a format specified by the monitoring system.
<code>apiVersion</code>	The API version. Currently always 1.



Name	Description
status	The current status of the device.
faultReason	If the status is FAULT, the reason for the fault. Otherwise blank.
faultDuration	If the status is FAULT, How long the fault has now persisted for, in seconds. Otherwise zero.
recoverAfter	If the status is FAULT, an ISO-format date-time. The SRM will attempt to restore the connection to the device shortly after this time.
	<p><b>Note</b> A fixed timezone is used. This timezone is a configuration option. It's the same value which is used when calculating <a href="#">Monitored viewing periods</a> <sup>p.33</sup>.</p> <p>Otherwise blank.</p>

Possible status values are as follows.

Value	Description
CONNECTING	Attempting to establish a TCP connection to the device.
OPEN	TCP connection established. Negotiating protocol. This may include recovery / resynchronization operations following a failure.
VALID	The device is ready to use.
DISCONNECTING	The connection is being shut down.
DISCONNECTED	The connection has been shut down. This state only lasts for a very short period of time.
FAULT	The connection has failed.

## 4 Configuration reference

**Warning!**

All of these configuration parameters are for advanced users only.

They have all been added to puppet and can be configured via Foreman - there should NEVER be a reason to modify the xml directly

### 4.1 core-conf.xml

This Spring bean file is located at `/opt/srm/webapps/srm-deployer/WEB-INF/core-conf.xml` .

#### 4.1.1 Configuration bean: bodcmConfig

These properties allow advanced users to tune the behaviour of the BODCM.

This bean is located in file `/opt/srm/webapps/srm-deployer/WEB-INF/core-conf.xml` .

**Caution!**

All of these properties are for advanced users only. For most deployments, the defaults are fine. Support will not be provided for deployments where these values have been changed, unless they were changed on the specific instructions of Nagra support.

Property	Type	Description	Version Available in Foreman
connectTimeout Secs	Integer	Timeout for initial TCP connection to the device.	3.0STD3
firstRecovery DelaySecs	Integer	After an endpoint fails, the BODCM will attempt to recover it, using exponential backoff. This is the first delay before recovery; the recovery attempt will not happen before this long after the initial failure.	3.0STD3
maxRecovery DelaySecs	Integer	The exponential backoff is capped at this delay. Should be a positive integer power of two multiplied by the firstRecoveryDelaySecs.	3.0STD3
lifelockTimeout Millis	Integer	Maximum time to wait for the lifelock. If the lifelock is not available almost immediately this means that a recovery attempt is in progress. You shouldn't need to alter this.	3.0STD3

Property	Type	Description	Version Available in Foreman
		<p><b>Note</b></p> <p>The <b>lifelock</b> is an internal semaphore which separates regular traffic from connection-management traffic (setup / recovery). Regular operations share the connection, but connection-management operations need to have the connection to themselves.</p>	
requestTimeout Secs	Integer	How long to wait for a response to a blocking request.	3.0STD3
rendezvous TimeoutMillis	Integer	How long to wait for the requesting thread to reach the rendezvous with the comms thread which has the response. If the requesting thread does not get there almost immediately, something is wrong and / or the SRM is severely overloaded (to the point where it's probably unusable). You shouldn't need to alter this.	3.0STD3
callbackExecutor PoolSize	Integer	Size of callback executor pool. These threads are used to handle unsolicited incoming requests (eg. RTSP ANNOUNCE messages) and responses which don't correspond to any known request. (This is an error condition, and should be rare.)	3.0STD3
mgmtExecutor PoolSize	Integer	Size of management executor pool. More gives faster, more resource-intensive startup and recovery from networking failures.	3.0STD3

## 4.1.2 Configuration bean: mongoDB

This bean defines how the SRM talks to MongoDB.

This bean is located in file `/opt/srm/webapps/srm-deployer/WEB-INF/core-conf.xml` .

**Caution!**

Properties marked "Sometimes" in the "Customize?" column may need to be changed to support more unusual configurations.

Properties marked "Rarely" typically do not need to be changed. Don't change these unless you know what you're doing.

Properties marked "Only when advised" can be dangerous to change. Support will not be provided for deployments where these values have been changed, unless they were changed on the specific instructions of Nagra support.

Property	Type	Description	Customize?	Version Available in Foreman
hostName	String	The IP address or DNS name of the MongoDB to connect to. Recommended configuration is to run MongoDB instances on the same hosts that run the SRM, in which case the default value ( <code>localhost</code> ) should work.  <b>Caution!</b> If the OS-level DNS is not configured correctly, you must change the hostName to a suitable IP address. Lookups to a misconfigured or malfunctioning DNS can be extremely slow.	Sometimes	3.0STD3
port	Integer	The port of the DB to connect to; this defaults to 4230 ready for use with Mongo 3.0 instances.	Sometimes	4.0STD1
dbName	String	The name of the DB to connect to. The equivalent of an RDBMS schema.	Rarely	3.0STD3
adminUserName	String	The username of the MongoDB administrator account.	Rarely	3.0STD3
adminPassword	String	The password for the MongoDB administrator account.	Sometimes	3.0STD3
normalUserName	String	The username for the MongoDB account to use for regular operations. Must be different than the administrator username.	Rarely	3.0STD3

Property	Type	Description	Customize?	Version Available in Foreman
normal Password	String	The password to the regular-operations account.	Sometimes	3.0STD3
max Connections	Integer	The maximum number of concurrent connections to the DB. The correct setting for this should be calculated based on system sizing parameters: see <a href="#">Configuration bean: srmCoreConfig</a> <sup>p.48</sup> .	Sometimes	3.0STD3
maxWaiters PerConnection	Integer	The maximum number of threads which may be waiting for each connection. If this limit is exceeded, the system is severely overloaded, and further incoming requests will fail until the load drops.	Only when advised	3.0STD3
initialConnect TimeoutMillis	Integer	How long to wait for an initial connection to the DB (ms).	Only when advised	3.0STD3
waitTimeout Millis	Integer	How long to wait for a reply to a DB request (ms).	Only when advised	3.0STD3
ioTimeoutMillis	Integer	How long to wait for an outgoing request to reach the network.	Only when advised	3.0STD3
allowSlave Reads	true or false	Where MongoDB is replicated (the usual case), whether it's safe to read from a slave replica. This is faster but not always safe; currently this is disabled by default.	Only when advised	3.0STD3
keepaliveWrite Safety	One of: <b>ERRORS_IGNORED, UNACKNOWLEDGED, ACKNOWLEDGED, JOURNALED, FSYNCD, REPLICA_ACKNOWLEDGED, MAJORITY</b>	The write-safety level to use when writing last-kept-alive time updates to existing sessions.	Only when advised	3.0STD3

Property	Type	Description	Customize?	Version Available in Foreman
	See <a href="#">MongoDB write safety levels</a> <small>p.322</small>			
defaultWrite Safety	One of: <b>ERRORS_</b> <b>IGNORED,</b> <b>UNACKNOWLEDGED,</b>  <b>ACKNOWLEDGED,</b> <b>JOURNALED,</b> <b>FSYNCEd,</b> <b>REPLICA_</b> <b>ACKNOWLEDGED,</b> <b>MAJORITY</b>	The write-safety level to use for regular operations.	Only when advised	3.0STD3
extraWrite Safety	One of: <b>ERRORS_</b> <b>IGNORED,</b> <b>UNACKNOWLEDGED,</b>  <b>ACKNOWLEDGED,</b> <b>JOURNALED,</b> <b>FSYNCEd,</b> <b>REPLICA_</b> <b>ACKNOWLEDGED,</b> <b>MAJORITY</b>	The write-safety level to use for critical cases where conflicts are expected.	Only when advised	3.0STD3
dropOnStartup	Constant	<b>Undocumented.</b> For internal testing only. Don't change this!	Never	3.0STD3

## 4.2 srm-conf.xml

This Spring bean file is located at `/opt/srm/webapps/srm-deployer/WEB-INF/srm-conf.xml` .

### 4.2.1 Configuration bean: ermConfig

This bean defines additional parameters for the connections to ERMs.

This bean is located in file `/opt/srm/webapps/srm-deployer/WEB-INF/srm-conf.xml` .

You can use the default values for all of these properties.

Property	Type	Description	Version Available in Foreman
timeoutRecheckIntervalMins	Integer	<p>The connections between the SRM and its ERMs will time out after a while; if this happens, the ERM will close the connection. The SRM is required to ensure that the connection stays open, so it must send keepalive messages periodically. The ERM itself defines the connection timeout interval. When the SRM opens a connection to any ERM, it queries the ERM to discover the connection timeout interval, and uses this to determine the frequency with which it will send keepalive messages.</p> <p>This property tells the SRM how frequently to re-query the connection timeout interval. Specified in minutes. <b>NB. This is not</b> the keepalive interval itself.</p>	3.0STD0
sessionKeepaliveIntervalMins	Integer	<p>How frequently to send session keepalive (PING) messages. Cisco VERM only - NGOD S6 doesn't do this.</p>	3.0STD0
fullResyncIntervalMins	Integer	<p>How frequently to force a full resync, even if the connection is healthy. NGOD S6 only - Cisco VERM doesn't do this.</p>	3.0STD0
maxInFlightMessages	Integer	<p>How many messages can be awaiting responses at once before the connection to the ERM blocks. If set to zero, there is no upper limit. This limit may be needed in the case where overloading an ERM causes it to malfunction or reject traffic ingracefully.</p>	3.0STD0

## 4.2.2 Configuration bean: sessionConfig

These properties control details about session management and session ID generation. They are broken out into a separate bean for technical reasons.

This bean is located in file `/opt/srm/webapps/srm-deployer/WEB-INF/srm-conf.xml` .

Property	Type	Description	Version Available in Foreman
minScsSessionId	Integer	<b>Undocumented.</b> Reserved for future expansion.	Not Puppetised
maxScsSessionId	Integer	<b>Undocumented.</b> Reserved for future expansion.	Not Puppetised

### 4.2.3 Configuration bean: srmCoreConfig

This bean is a "catch-all" for properties which don't fit into the other beans.

This bean is located in file `/opt/srm/webapps/srm-deployer/WEB-INF/srm-conf.xml`.

**Caution!**

Properties marked "Sometimes" in the "Customize?" column may need to be changed to support more unusual configurations.

Properties marked "Rarely" typically do not need to be changed. Don't change these unless you know what you're doing.

Properties marked "Only when advised" can be dangerous to change. Support will not be provided for deployments where these values have been changed, unless they were changed on the specific instructions of Nagra support.

Property	Type	Description	Customize?	Version Available in Foreman
bandwidthMap	(Complex)	<p>The SRM needs to know how much bandwidth to reserve for a given playout request. It supports having different bandwidths for each asset (ie. each asset defines exactly how much bandwidth it needs). However, it's more common to define a fixed bandwidth for each supported definition (SD / HD / 3D), and have the individual assets only specify their definition.</p> <p>The bandwidth map defines both the set of supported definitions, and the bandwidths required for each. Bandwidths are specified in kilobits per second.</p>	Sometimes	3.2STD4



Property	Type	Description	Customize?	Version Available in Foreman
maxContentLengthInBytes	Integer	<p>This property allows the SRM to change the maximum length of the content within an RTSP response. The default within the SRM is 8192.</p> <p>There is currently no need to change this, and it would only need modification should a content be &gt; 8K in size.</p> <p>It was originally implemented in order to allow the SRM to support a large content for the GET_PARAMETER responses - however the specific large GET_PARAMETER call was not required and removed.</p>	Never	3.2STD5
<b>Outbound ANNOUNCEs to clients</b>				
outboundAnnounceCallbackUrl	String	When the SRM sends an ANNOUNCE message to a client via NGOD.S1, it must specify its own URL in the request top line. The URL to send.	Always	3.0STD0
outboundAnnounceRequestTimeoutSecs	Integer	Time to wait when the SRM sends an ANNOUNCE message. Reserved for future use.	Rarely	3.0STD0
announceOutboundThirdpartyTimeoutSecs	Integer	Time to wait when the SRM sends an ANNOUNCE message to a Cablelabs client.	Rarely	3.2STD2
<b>PAV Connection to SDP</b>				
pavUriPrefix	String	A Prefix to be used to invoke the PAV. For ML releases this will be "/api". For older SDP releases "/ws-gateway"	Rarely	3.1STD3

Property	Type	Description	Customize?	Version Available in Foreman
pavGateway Password	String	The password corresponding to the username. The default will work with an "out of the box" PAV configuration, but that configuration should will have been changed as part of the SDP installation process.	Always	3.0STD0
pavGateway Username	String	The username to use when contacting the PAV. The default will work with an "out of the box" SDP configuration, but the SDP should be locked down more than that.	Sometimes	3.0STD0
pavGateway PrimaryPav Device	String	The ingest name of the <a href="#">Pav device</a> <sup>p.11</sup> which the SRM should contact for authorization. If this device is found to be faulty or the request fails due to a communications failure, the SRM will fail over and try other Pav devices (if any). (The SRM will not fail over if the request gets through, but is denied - if this happens, every Pav should give the same answer.)	Sometimes	3.0STD0
		<p><b>Note</b></p> <p>In the most common arrangement, the SRM and its primary SDP are on the same machine. In this case, ingest a Pav device called <b>loopback</b>, set its address to <b>127.0.0.1</b>, and leave the sdp GatewayPrimaryPav Device set to <b>loopback</b> on all SRM instances. This will cause each SRM to call its own local SDP first. Mostly this is what you want.</p>		
		<p><b>Caution!</b></p> <p>If the OS-level DNS is not configured correctly, you must customize this to specify a suitable</p>		

Property	Type	Description	Customize?	Version Available in Foreman
		IP address. Lookups to a misconfigured or malfunctioning DNS can be extremely slow.		
pavGatewayConnectTimeoutMillis	Integer	How long to wait for the initial connection to the SDP. Connections are cached.	Rarely	3.0STD0
pavGatewayResponseTimeoutMillis	Integer	How long to wait for a response to a PAV request.	Rarely	3.0STD0
pavGatewayFirstRecoveryDelaySecs	Integer	After a PAV (SDP) endpoint fails, the SRM will attempt to recover it, using exponential backoff. This is the first delay before recovery; the recovery attempt will not happen before this long after the initial failure.	Rarely	3.0STD0
pavGatewayMaxRecoveryDelaySecs	Integer	The PAV endpoint exponential backoff is capped at this delay. Should be a positive integer power of two multiplied by the pavGatewayFirstRecoveryDelaySecs	Rarely	3.0STD0
<b>System Sizing</b>				
maxConcurrentRequests	Integer	The number of client requests which the SRM will support concurrently. If more requests than this are received concurrently, extra requests will fail fast (the SRM will return 503 Service Unavailable.)	Sometimes	3.0STD0
		<b>Note</b>		
		This is the main sizing parameter for the system. Increase it if clients are being rejected; decrease it if the SRM is using		

Property	Type	Description	Customize?	Version Available in Foreman
		<p>excessive system resources.</p> <p><b>Caution!</b> If you change this, you must also change maxConnections in <a href="#">Configuration bean: mongoDB</a> <sup>p.43</sup>. You should make sure that maxConnections is at least 10 larger than (max ConcurrentRequests + teardownPoolSize).</p>		
logicPoolSize Divisor	Floating-point	<p>Used to determine the number of threads to use to process client requests. The number of threads is obtained by dividing maxConcurrentRequests by this value. Increasing this value slightly will take advantage of non-blocking setup requests to reduce the SRM's memory footprint.</p> <p><b>Note</b> Setup requests contain both blocking and asynchronous steps. The request doesn't need a logic thread while an asynchronous step is taking place, so it can give that thread back to allow another request to use it. In this way, on average, the SRM uses slightly less than one logic thread per request. This effect means that, for a given number of concurrent requests (which include setup requests), the SRM needs a slightly smaller number of logic threads.</p>	Only when advised	Not yet applicable - Future Config

Property	Type	Description	Customize?	Version Available in Foreman
teardownPool Size	Integer	<p>The number of threads available for performing asynchronous teardowns.</p> <p><b>Note</b> It can be helpful to increase this if you often have spiky load patterns. Don't decrease it.</p> <p><b>Caution!</b> If you change this, you must also change maxConnections in <a href="#">Configuration bean: mongoDB</a> <sup>p.43</sup>. You should make sure that maxConnections is at least 10 larger than (max ConcurrentRequests + teardownPoolSize).</p>	Sometimes	3.0STD0
teardownMax Backlog	Integer	<p>The maximum permitted backlog (queue) size for pending asynchronous teardowns. Keep this large - if the queue gets full, further teardown requests will block the threads which invoked them, which will reduce the responsiveness of the SRM or even jam it completely until the backlog clears. The backlog must have a maximum size: if it didn't then, it could expand until the SRM ran out of memory (which would cause it to malfunction.)</p>	Only when advised	3.0STD0
monitorQueue Limit	Integer	<p>The monitoring system uses an internal queueing mechanism; this avoids requests from interfering with each other or with the background mechanisms which keep the monitoring information up to date. The maximum queue length must be much larger than the peak incoming requests per second (including all keepalive requests). At the same time, a very large queue</p>	Rarely	3.0STD0

Property	Type	Description	Customize?	Version Available in Foreman
		limit will cause excessive memory consumption.		
management PoolSize Divisor	Floating-point	As well as the main customer-facing RTSP interface on port 5544, there is a <b>management</b> RTSP interface on port 5541. This allows ANNOUNCE messages to be propagated between the various SRM instances. The number of threads used to process management requests is obtained by dividing maxConcurrentRequests by this value.	Only when advised	Not yet applicable - Future Config
management BacklogDepth	Integer	To be replaced by managementPoolSizeDivisor	Only when advised	3.0STD0
management PoolSize	Integer	To be replaced by managementPoolSizeDivisor	Only when advised	3.0STD0
overload Protection TimeoutSecs	Integer	When the SRM overloaded, it protects itself by rejecting all new requests for a period of time, so that it can catch up. The protection times out after a while - typically this should be a few seconds.	Rarely	3.0STD0
		<b>Note</b> When overload protection is active, the first PING of a new session will be rejected. However, other PINGs (non-initial PINGs on keepalive connections which were already in place before the overload happened) are still allowed through. This is because they can be handled using a "fast track" process which uses less resources.		

SDB

Property	Type	Description	Customize?	Version Available in Foreman
maxTuners	Integer	<b>Undocumented.</b> Reserved for future expansion. The maximum number of tuners that can be registered by a single STB. Required to prevent DoS attacks by rogue STBs.	Never	3.0STD0
sdbMax Channel AllocationPer STB	Integer	<b>Undocumented.</b> Reserved for future expansion. A given STB is only allowed to watch a certain number of SDB channels simultaneously.	Never	3.0STD0
sdbAuto ProvisionSTBs	Enum: true or false	<b>Undocumented.</b> Reserved for future expansion. Whether to automatically add new STB records when an unknown MAC address is encountered. Must be true if STBs are not ingested to SDP. This is vulnerable to rogue / pirate STBs. False is more secure.	Never	3.0STD0
sdbAuto ProvisionSpid	Integer	<b>Undocumented.</b> Reserved for future expansion. Where a new STB record is added, the SPID it should be added to.	Never	3.0STD0
externalRange StbSpid	Integer	<b>Undocumented.</b> Reserved for future expansion. Where an external-range is used, a dummy STB record will be auto-created. The SPID for such STBs.	Never	3.0STD0
sdbUse Definition Fallback	Integer	<b>Undocumented.</b> Reserved for future expansion. When a channel (number) is available in multiple definitions, attempt to fall back to a lesser definition if a setup at higher definition failed. This is sometimes called "Lesser stream" support.	Never	3.0STD0
sdbSrmClient Name	String	<b>Undocumented.</b> Reserved for future expansion. The name of the SrmClient used for SDB. Currently only one such is supported.	Never	3.0STD0
ottSrmClient Name	String	The name of the SrmClient used for OTT.	Never	3.0STD0

Property	Type	Description	Customize?	Version Available in Foreman
ottKeepalives Cached	Boolean	When true uses a Redis cache for keepalive requests rather than Mongo DB.	Rarely	3.2STD3
<b>VOD</b>				
vodLease PeriodMins	Integer	For a VOD payout, we effectively lease resources to an individual STB. However, this lease is time-limited. If the session is still alive after this period, it will be forcibly torn down, even if the session is still in good order.	Sometimes	3.0STD0
<b>User Behaviour Reporting</b>				
ubrPurgeTime	String	The time of day when time-expired User Behavior Reporting records should be deleted. Timezone as defined by <code>systemTimezone</code> (see below). Format is <code>HH:mm</code> using 24-hour clock. Eg <code>04:00</code> = 4 am.	Sometimes	3.0STD0
ubrPurgeAfter Days	Integer	A User Behavior Reporting record may be deleted, this many days after the end of the session to which it refers.	Sometimes	3.0STD0
<b>VOD Failure Reporting</b>				
vodFailure PurgeTime	String	The time of day when time-expired VOD Failure Reporting records should be deleted. Timezone as defined by <code>systemTimezone</code> (see below). Format is <code>HH:mm</code> using 24-hour clock. Eg <code>04:00</code> = 4 am.		
vodFailure PurgeAfter Days	Integer	A VOD Failure Reporting record may be deleted, this many days after the end of the session to which it refers.		



Property	Type	Description	Customize?	Version Available in Foreman
<b>Immediate Session Termination</b>				
force Disconnect SleepMillis	Integer	When force-disconnecting multiple sessions, how long to wait between initiating each teardown. This delay spreads out the load spike which the process causes, and mitigates the risk that the spike will make the SRM unresponsive.	Rarely	3.0STD0
force Disconnect StartTimeout Millis	Integer	When force-disconnecting multiple sessions, how long the initiating REST call will wait for the process to start, so that it can return the count of affected sessions. Normally this will happen very quickly, but there may be a delay if the system is heavily loaded.	Rarely	3.0STD0
<b>Environment</b>				
system Timezone	String	The Timezone the SRM should use by default. Currently this is used for the <b>ubrPurgeTime</b> , and by the monitoring system to determine when hourly and daily turnover should take place. If left blank, uses the OS default timezone. Otherwise, use one of the specified Timezones.	Sometimes	3.0STD0
<b>Data Ingest File Size</b>				
maxIngestFile Size	Integer	The maximum size for a <a href="#">Bulk data ingest</a> <sup>p.82</sup> file. Value is in megabytes.	Rarely	3.0STD0
<b>HTTP Router Auth Token</b>				
accUidToken Index	Integer	The position within the token of the account uid. This is configurable to	Rarely	3.0STD0

Property	Type	Description	Customize?	Version Available in Foreman
		avoid changes to the SRM if the token entries are amended/re-ordered		
accNoToken Index	Integer	The position within the token of the account number. This is configurable to avoid changes to the SRM if the token entries are amended/re-ordered	Rarely	3.0STD0
deviceUid TokenIndex	Integer	The position within the token of the device uid. This is configurable to avoid changes to the SRM if the token entries are amended/re-ordered	Rarely	3.0STD0
userIdToken Index	Integer	The position within the token of the user uid. This is configurable to avoid changes to the SRM if the token entries are amended/re-ordered	Rarely	4.0STD4

#### 4.2.4 Configuration bean: vsConfig

This bean defines additional parameters for the connections to video servers.

This bean is located in file `/opt/srm/webapps/srm-deployer/WEB-INF/srm-conf.xml` .

**Caution!**

Properties marked "Sometimes" in the "Customize?" column may need to be changed to support more unusual configurations.

Properties marked "Rarely" typically do not need to be changed. Don't change these unless you know what you're doing.

Properties marked "Only when advised" can be dangerous to change. Support will not be provided for deployments where these values have been changed, unless they were changed on the specific instructions of Nagra support.

Property	Type	Description	Customize?	Version Available in Foreman
sdvVsSrmlp Address	String	The Address to be used by a SDV client for requesting a keep-alive. Generally this would be the address of a load-balancer	Always	3.1STD1

Property	Type	Description	Customize?	Version Available in Foreman
sdvVsSrmPort	Integer	The port to be used by a SDV client for requesting a keep-alive. Generally this would be the address of a load-balancer	Always	3.1STD1
effectiveSrmlp Address	String	In NGOD deployments, a successful SETUP response to a client includes an SDP (Session Description Protocol) block which provides additional metadata to the client. The SDP block includes this value - the effective IP address of the SRM, ie. the address which the client should use. The SRM itself cannot determine this in general: for example, if there is a firewall between the clients and the SRM the clients may need to contact the firewall rather than the SRM itself. Therefore it must be set manually.	Always	3.0STD0
timeout Recheck IntervalMins	Integer	The connections between the SRM and its VSs will time out after a while; if this happens, the VS will close the connection. The SRM is required to ensure that the connection stays open, so it must send keepalive messages periodically. The VS itself defines the connection timeout interval. When the SRM opens a connection to any VS, it queries the VS to discover the connection timeout interval, and uses this to determine the frequency with which it will send keepalive messages.  This property tells the SRM how frequently to re-query the connection timeout interval. Specified in minutes. <b>NB.</b> This is <b>not</b> the keepalive interval itself.	Rarely	3.0STD0
session Keepalive IntervalMins	Integer	<b>Undocumented.</b> How frequently to send session keepalive (PING) messages, in dialects which require this. NGOD R2 does not. Reserved for future expansion.	Never	3.0STD0

Property	Type	Description	Customize?	Version Available in Foreman
connection Keepalive IntervalSecs	Integer	How frequently in seconds to send connection keepalive messages; this is currently only supported for NAGRA VS.	Sometimes	3.0STD0
fullResync IntervalMins	Integer	How frequently to force a full resync, even if the connection is healthy. NGOD R2 only.	Rarely	3.0STD0
maxInFlight Messages	Integer	How many messages can be awaiting responses at once before the connection to the VS blocks. If set to zero, there is no upper limit. This limit may be needed in the case where overloading a VS causes it to malfunction or reject traffic ingracefully.	Rarely	3.0STD0
defaultSop Group	String	For deployments using NGOD R2 video servers, the default SOP group to use. If a system is also using NGOD S6 ERMs, and the ERM returns an <code>edge_input_group</code> declaration, then the <code>edge_input_group</code> will be sent as the SOP group, overriding the value given here.  <b>Note</b> NGOD R2 allows a SOP name to be specified rather than a SOP group; this allows the session manager to constrain the SOP more tightly. The SRM does not support SOP names.	Rarely	3.0STD0
responseCode FailoverList	List	A list of VS response codes that should result in a failover to an alternate VS	Rarely	3.0STD0
vsClientFormat	String	The format for the client-id string sent from the SRM to the VS.  E.g., where <code>smartcard_id = 170148375</code> , <code>TSID=64753</code> and <code>Service-Group = 201</code> :	Rarely	3.2STD1

Property	Type	Description	Customize?	Version Available in Foreman
		SMARTCARD_ID.TSIDSERVICE_GROUP01 will resolve as 170148375.6475320101		

## 4.3 stats-conf.xml

This Spring bean file is located at `/opt/srm/webapps/srm-deployer/WEB-INF/stats-conf.xml` .

### 4.3.1 Configuration bean: monitoredHistogramTemplates

This bean allows very advanced users to tune how the [Monitoring statistics](#) <sup>p.99</sup> API interprets data.

This bean is located in file `/opt/srm/webapps/srm-deployer/WEB-INF/stats-conf.xml` .

**Warning!** Do not alter this file without support from Nagra.

This bean defines a mapping from a [Monitored histogram](#) <sup>p.34</sup> template name to the template itself. The templates are defined in further beans in the same file.

By default, there are several cases where different template names map to the same template. You may split these out (ie add a new template, and remap one of the template names to use it.)

**Caution!** Don't add new template names to the map - the SRM will ignore them.  
Really don't delete any template names - the SRM will stop working if you do this.

#### Histogram template definitions

Underneath the monitoredHistogramTemplates bean itself, are beans which define the templates which monitored HistogramTemplates refers to.

Each bean contains 4 constructor-args. In order, these are:

1. The list of **bin initializer** strings. Each of these strings is in the form **name=limit**:

Name	Limit
The name of the category.	The upper limit of the category.

**Note**

The lower limit of the first category is zero. (The SRM does not support any quantities which can take negative values.)

The lower limit of every other category is one more than the upper limit of the previous category.

The supplied upper limit of the last category is a dummy value: the actual upper limit is effectively infinite. (Actually it's (263 - 1), since this is the largest value which can be represented efficiently.)

2. The range-description preposition. An infix used when constructing the range descriptions.
3. The unit. A suffix used when constructing the range descriptions.
4. The suffix used when constructing the names of the attributes which define the total sample range. For example, if this is **Time**, then these attributes will be called **minTime** and **maxTime**.

**Other beans**

After the histogram template beans are two final beans: **monitoredTypes** and **monitoredFixedObjects**. Leave them alone.

Nagra

## 5 Data ingest reference

### 5.1 Data ingest file format

SRM data ingest files are CSV-format text files, compatible with <http://www.ietf.org/rfc/rfc4180.txt>. It's recommended to create these files using a spreadsheet application. Microsoft® Excel 2010 is known to produce compatible output, but most other modern spreadsheets should be compatible as well.

The file consists of a series of data blocks, which will be processed in order. Each data block is a sequence of at least 3 lines (spreadsheet rows):

1. The first line is the **block heading**.
2. The second line is the **column heading**.
3. The third and subsequent lines are **data records**.

Any line whose first field starts with a # character is considered to be a comment, and is ignored.

Lines which are completely blank (or only contain spaces, tabs or other "whitespace" characters) are also ignored.

Comments or blank lines may occur at any point, including within a data block.

**Caution!** Fields may be delimited with double-quote characters ". Because of this, these characters are not permitted within any data field.

**Caution!** Every line in a well-formed CSV file must have the same number of fields, including block headings, comments and "blank" lines. A spreadsheet application will enforce this for you.

### Example

This block of CSV creates or updates two streaming servers.

```
Sts,override,,,,,,,,,,,,,  
Name,Spid,Address,TcpPort,CodecId,Enabled,,,,,,,,,,,,,  
,,,,,,,,,,,,,  
Vs1-Gobelins,3,172.22.6.94,5546,NgodR2,TRUE,,,,,,,,,,,,,  
Vs2-Gobelins,3,172.22.6.95,5546,NgodR2,FALSE,,,,,,,,,,,,,
```

### 5.2 Block headings

A block heading is a record whose first 2 columns only are populated. The first record of each block must be a block heading.

Column	Value
1	The type of data record ("object") this block contains. One of the valid <a href="#">Record types</a> <sup>p.64</sup>
2	The operation to perform. One of the valid <a href="#">Operations</a> <sup>p.65</sup>

**Caution!** The values of these columns must be specified exactly as written (case-sensitive.)

## 5.2.1 Record types

Record type	Description
(Block heading)	Starts a new block containing records of a particular type. This record type cannot be used within a block. See <a href="#">Block headings</a> <sup>p.63</sup>
ChannelBandwidth	Defines the bandwidth required for a given CU (Catch-Up) channel. See....
Erm	Edge Resource Manager. Represents the device which manages QAMs. See <a href="#">Data ingest: Erm device fields</a> <sup>p.70</sup>
Sts	Streaming Server. Represents the device which transmits video data to QAMs or clients. See <a href="#">Data ingest: Sts device fields</a> <sup>p.78</sup>
DeviceGroup	Represents a group of streaming servers which can stream to a Service Group. See <a href="#">Data ingest: DeviceGroup fields</a> <sup>p.68</sup>
ServiceGroup	Represents an RF output port from a QAM. Conceptually, a STB is connected directly to a single Service Group. See <a href="#">Data ingest: ServiceGroup fields</a> <sup>p.73</sup>



Record type	Description
ServiceGroupResource	<p>Represents of a single RF program / service, being transmitted via a Service Group.</p> <p>See <a href="#">Data ingest: ServiceGroupResource fields</a> <sup>p.75</sup></p> <p><b>Note</b></p> <p>ServiceGroupResources should only be ingested for Service Groups which are managed using table-based (static) bandwidth allocation.</p> <p>When dynamic bandwidth allocation is in use, you must ingest the ServiceGroup, but not any ServiceGroupResources, since these are managed by the ERM.</p>
SrmClient	<p>Defines an STB dialect supported by the deployment, and the workflows which the dialect uses.</p> <p>See <a href="#">Data ingest: SrmClient fields</a> <sup>p.76</sup></p>
SrmServer	<p>An <b>SrmServer</b> represents a connection to an SRM instance. Each SRM instance maintains a connection to every SRM instance in the same deployment (including itself). This is used to propagate ANNOUNCE messages to clients when sessions are torn down.</p> <p>See <a href="https://atlassian.hq.k.grp/confluence/display/SRM/Data+ingest%3A+SrmServer+device+fields">https://atlassian.hq.k.grp/confluence/display/SRM/Data+ingest%3A+SrmServer+device+fields</a> <a href="http://engwiki/display/SRM/Data+ingest%3A+SrmServer+device+fields">http://engwiki/display/SRM/Data+ingest%3A+SrmServer+device+fields</a> <a href="http://engwiki/display/SRM/Data+ingest%3A+SrmServer+device+fields">http://engwiki/display/SRM/Data+ingest%3A+SrmServer+device+fields</a></p>
Pav	<p>A Pav (Playout Authorization Verifier) represents an instance of the SDP. The SRM will use this to locate SDP instances which it can call to authorize a session setup.</p> <p>Typically the SRM shares a machine with an SDP instance. The SRM has a <b>primary</b> Pav, the name of which is stored in the SRM's configuration; it will always use this Pav if it's available. Otherwise, it will fail over to other Pavs, if available.</p> <p>See <a href="http://engwiki/display/SRM/Data+ingest%3A+Pav+device+fields">http://engwiki/display/SRM/Data+ingest%3A+Pav+device+fields</a></p>

## 5.2.2 Operations

Name	Description
Override	Update if present, or create.

Name	Description
Delete	Remove the record.
Deactivate	Only permitted for ServiceGroup blocks. Set the ServiceGroup to inactive, but don't change anything else.

### 5.3 Column headings

The column heading defines which values you intend to ingest, and the order in which you're specifying them. The second record in each block must be a column heading.

The valid entries in the column heading depend on which type of record the block is ingesting. See the reference section for each type.

**Caution!**

The column names must be specified in the ingest file exactly as written in the reference section (case-sensitive.)

### 5.4 Data records

After the first two records, each record in the block is a data record. It contains the values of the fields of a single object, in the order specified by the column heading.

#### Object identifiers

Each record has a field which uniquely identifies the corresponding object in the system. For operation **Override**, this field is required in order to determine whether the record is being created or updated. For operation **Delete** or **Deactivate**, this field is required in order to determine the record to delete.

Record type	Name of identifier field
Erm	Name
Qam	Name
Scs	Name
Sts	Name

Record type	Name of identifier field
Cryptocaster	Name
ServiceGroup	IngestName
	<p><b>Note</b></p> <p>The <b>Name</b> of a ServiceGroup is not always unique, for historical reasons, so this field cannot be used to identify the ServiceGroup. These objects have an additional <b>IngestName</b> which must be unique.</p>
ServiceGroupResource	Name
SrmClient	Name

### 5.4.1 Data ingest: ChannelBandwidth fields

Ideally, the PAV should specify the amount of bandwidth required for a CU channel, in the same way that it always does for a VOD asset. However, some PAV implementations do not support this. Therefore the SRM allows the operator to ingest ChannelBandwidth records. Each record defines the required bandwidth for a single CU channel.

Field	Type	Description	Required
Name	String	The CMS public id of the channel-version.	Yes
Bps	Integer	The required bandwidth, in bits per second.	Either Bps or Definition is required
Definition	SD, HD or 3D	The channel definition. If a Bps value is not specified, the definition will be used to look up the bandwidth from the SRM's internal configuration table. The same table is used for CU as for VOD.	Either Bps or Definition is required

**Note**

If the PAV response does define a bitRate or definition field for a given request, these will override any ChannelBandwidth value.

### 5.4.2 Data ingest: SrmServer device fields

An **SrmServer** represents a connection to an SRM instance. Each SRM instance maintains a connection to every SRM instance in the same deployment (including itself). This is used to propagate ANNOUNCE messages to clients when sessions are torn down.

Field	Type	Description	Required
Name	String	A unique name.  <b>Caution!</b> To avoid compatibility issues, this field should not contain spaces, semicolons, colons, or commas.	Yes
Address	String	IP address or DNS name for the device.  <b>Caution!</b> If you use a DNS name, a misconfigured or slow DNS configuration can cause severe performance problems. It may be safer - although less convenient - to use an IP address.	Yes
Port	Integer	TCP port to connect to.  <b>Note</b> Usually this should be 5541. Must be the management RTSP port, not the Client RTSP port.	Yes
Status	ACTIVE or DISABLED	The status of the SrmServer.	

### 5.4.3 Data ingest: DeviceGroup fields

A DeviceGroup represents a group of Sts (streaming servers), which can stream to a ServiceGroup.

Field	Type	Description	Required
Name	String	A meaningful (to the operator) name for this device group.	Yes
DeviceNames	String	The names of the Sts devices in the group. The names are separated by commas.	Yes

**Caution!** No spaces are permitted between the names.

See [Data ingest: Sts device fields](#) <sup>p.78</sup>

### 5.4.4 Data ingest: Range

A **Range** is an abstract concept which represents the 'range' of smartcards between a specified minimum and maximum smartcard ID.

The **Range Type** is used to create a specific range that can be utilised within the solution.

Field	Type	Description	Required
Name	String	A unique name.	Yes
Type	String	The range type. Allowable values are: <ul style="list-style-type: none"> <li>▶ EXTERNAL (default) <ul style="list-style-type: none"> <li>• Used to indicate a device which is not managed by MediaLive PAV. So, a product validation check, nor basic device/account cross-check, will be performed.</li> </ul> </li> <li>▶ UNENCRYPTED <ul style="list-style-type: none"> <li>• Used to indicate a device which should not utilise an encryption contributor.</li> </ul> </li> </ul>	Yes

**Caution!** To avoid compatibility issues, this field should not contain spaces, semicolons, colons, or commas.

Field	Type	Description	Required
		<ul style="list-style-type: none"> <li>▶ BANNED           <ul style="list-style-type: none"> <li>• Used to block a device from access to the SRM.</li> </ul> </li> </ul>	
MinValue	Long	First smartcard ID to be covered by the range.	Yes
MaxValue	Long	Last smartcard ID to be covered by the range.	Yes
Enabled	Boolean	Whether or not the range should be used	Yes

### 5.4.5 Data ingest: Erm device fields

An **ERM** - Edge Resource Manager - manages a set of QAMs.

**Note**

Conventionally "ERM" is capitalised, since it's an abbreviation. However, when creating a data ingest, you must use the mixed-case form, **Erm**.

Field	Type	Description	Required
Name	String	A unique name.	Yes
		<p><b>Caution!</b> To avoid compatibility issues, the "name" field should not contain spaces, semicolons, colons, or commas.</p>	
Address	String	IP address or DNS name for the device.	Yes
		<p><b>Caution!</b> If you use a DNS name, a misconfigured or slow DNS configuration can cause severe performance problems. It may be safer - although less convenient - to use an IP address.</p>	
Port	Integer	TCP port to connect to.	Yes

Field	Type	Description	Required
Codecd	Fixed value: <code>ngods6</code>	The RTSP dialect the ERM uses. Reserved for future expansion.	Yes

Status **ACTIVE, INACTIVE, or DISABLED**

The status of the ERM.

Status	Description
ACTIVE	The ERM is fully available.
INACTIVE	The ERM is going offline soon. Existing sessions are left alone, but no new sessions will be started.
DISABLED	The ERM is offline. The SRM will disconnect from it, and will not attempt to re-establish a connection until its status changes.

**Caution!**

In order to ensure that open sessions can be closed cleanly, some changes are forbidden. Any attempt to perform such a change will be rejected by the data ingest.

- ▶ An ERM may not be changed from ACTIVE to DISABLED. Set it to INACTIVE first, and wait for all sessions to finish.
- ▶ An ERM may not be changed from INACTIVE to DISABLED if there are active sessions on it.
- ▶ An ERM may not be deleted if it is ACTIVE, or if there are active sessions on it.

### 5.4.6 Data ingest: Pav device fields

A Pav (Playout Authorization Verifier) represents an instance of the SDP. The SRM will use this to locate SDP instances which it can call to authorize a session setup.

Typically the SRM shares a machine with an SDP instance. The SRM has a configurable **primary** Pav, the name of which is stored in the SRM's configuration; it will always use this Pav if it's available; otherwise it will fail-over to other configured Pavs, if available.

**Note**

In the most common arrangement where both the SRM and its primary PAV (a local SDP instance) are on the same machine, ingest a Pav device called "loopback" with an Address of 127.0.0.1 (for IPV4) and Port value of 80.

The SRM default configuration is to use the "loopback" PAV as its primary, but if required the name of this primary PAV device can be configured in the "sdpGatewayPrimaryPavDevice" value in the <http://engwiki/display/SRM/Configuration+bean%3A+srnCoreConfig> configuration file.

Field	Type	Description	Required
Name	String	A unique name.  <b>Caution!</b> To avoid compatibility issues, this field should not contain spaces, semicolons, colons, or commas.	Yes
Address	String	IP address or DNS name for the device.  <b>Caution!</b> If you use a DNS name, a misconfigured or slow DNS configuration can cause severe performance problems. It may be safer - although less convenient - to use an IP address.	Yes
Port	Integer	TCP port to connect to.  <b>Note</b> Usually this should be 8180. Must be the port which offers Web Services.	Yes
Status	ACTIVE or DISABLED	The status of the Pav.	Yes
Version	2_0 or 2_1 or 2_7	The WSDL version supported by this PAV. ▶ 2_0 is for MediaLive 2.2 (or compatible). It only supports NGOD S1 clients.	No (defaults to 2_0)



Field	Type	Description	Required
		<ul style="list-style-type: none"> <li>▶ 2_1 is for MediaLive 2.3 and above (or compatible). It supports NGOD S1 and Open TV clients.</li> <li>▶ 2_7 is for MediaLive 2.7 and above (or compatible). It supports NGOD S1 and Open TV clients for cable and Nagra OTT clients</li> </ul>	
		<p><b>Caution!</b> You must enter this exactly as shown. Variants such as "2.0" or "2-0", will not work.</p>	

### 5.4.7 Data ingest: ServiceGroup fields

A ServiceGroup in the SRM is a representation of an RF output port from a QAM. Conceptually, a STB is connected directly to a single ServiceGroup.

ServiceGroups must be defined and ingested even when ERMs are in use.

Field	Type	Description	Required
Spid	Integer	SPID. Defines whether the ServiceGroup is permitted to be used by a given client (business constraint).	Yes
Name	String	A meaningful (to the operator) name for this service group. Indexed. For compatibility reasons, this may be non-unique.	Yes
		<p><b>Caution!</b> To avoid compatibility issues, this field must not contain spaces, semicolons, colons, or commas.</p>	
ErmName	String	The unique name of the ERM which manages the QAM. To use <a href="#">Table-based bandwidth allocation</a> <sup>p.13</sup> , leave this blank.	No
StsDeviceGroup Name	String	The unique name of the device-group containing the set of VS which can stream to this service-group.	Yes

Status	<b>ACTIVE</b> or <b>INACTIVE</b>	New sessions will not be set up on an inactive service group, but pre-existing sessions are allowed to persist. This allows a service group to be taken down for maintenance without disturbing the end-users. NB. Values must be uppercase.	Yes
NgodS6Names	String	If this ServiceGroup is managed by an NGOD S6 ERM, the list of <b>qam-names</b> applicable to this port. In NGOD, each RF channel (frequency) output has its own name. The names are separated by commas.  <b>Caution!</b> No spaces are permitted between the names.	No
MaxBps	Integer	The total bandwidth available to the adaptor. Only used for table-based (internal) bandwidth management; ignored where the ServiceGroup is managed by an ERM.  <b>Note</b> If not supplied, a default value of 38,000,000 is set. This has been the value most commonly seen in practice so far.	No
DefaultRfModulation	Integer: a QAM modulation code	When using an ERM, the modulation to specify if the ERM does not. See <a href="#">QAM modulation codes</a> <small>p.324</small>	No
DefaultRfSymPerSec	Integer	When using an ERM, the symbol rate to specify if the ERM does not.	No
DefaultRfInnerFec	Integer	When using an ERM, the code for the inner Forward Error Correction method to specify, if the ERM does not.	No
DefaultRfOuterFec	Integer	When using an ERM, the code for the outer Forward Error Correction method to specify, if the ERM does not.	No
IngestName	String	A unique name for this ServiceGroup, used during ingest. May be the same as the name, if that happens to be unique.	Yes

**Caution!**

If this deployment uses OpenTV-compatible STBs, the IngestName is also used to identify the ServiceGroup during session setup. In this case, the IngestNames must be of the form `networkId.nodeId` .

NgodR2SopGroup	String	If an NGOD R2 Sts transmits to this ServiceGroup, the SOP group to specify when setting up a session on the Sts.	No
----------------	--------	--	----

**Note**

If this ServiceGroup is managed by an NGOD S6 ERM, this value is overridden by the edge input group returned from the ERM.

### 5.4.8 Data ingest: ServiceGroupResource fields

A ServiceGroupResource represents a single RF program / service, being transmitted via a ServiceGroup.

When a ServiceGroup is managed using table-based allocation, its ServiceGroupResources must be predefined and ingested.

When a ServiceGroup is managed by an ERM, its ServiceGroupResources must not be ingested into the SRM.

Field	Type	Description	Required
Name	String	A unique name.	Yes
SgIngestName	String	The ingest name of the ServiceGroup which owns this resource.	Yes
RfFreqHz	Integer	The centre frequency in Hertz.	Yes
RfDvbNetworkId	Integer	Original Network ID. The first part of the <b>DVB triplet</b> used to identify a resource.	Yes

**Caution!**

To avoid compatibility issues, the "name" field should not contain spaces, semicolons, colons, or commas.

RfDvbTsid	Integer	Transport Stream ID. The second part of the DVB triplet. Indexed.	Yes
RfSymPerSec	Integer	The symbol rate, in symbols per second.	No
RfModulation	Integer: a QAM modulation code	A code for the modulation technique used on this RF. See <a href="#">QAM modulation codes</a> <sup>p.324</sup>	Yes
RfInnerFec	Integer	A code for the inner Forward Error Correction method used on this RF.	No
RfOuterFec	Integer	A code for the outer Forward Error Correction method used on this RF.	No
DvbServiceId	Integer	The final part of the DVB triplet.	Yes
DestIpAddress	String	Destination IP address. The VS will send here, for VOD. For SDB, it will be joined to the multicast from the broadcast head-end. Indexed.	Yes
DestUdpPort	Integer	Destination UDP port. See comments for The VS will send here, for VOD. For SDB, it will be joined to the multicast from the broadcast head-end. Indexed.	Yes

### 5.4.9 Data ingest: SrmClient fields

An SrmClient defines an STB dialect supported by the deployment, and the workflows which the dialect uses.

Field	Type	Description	Required
TenantName	String	Name of Tenant. This key/Id is optional if 'default TenantName' is set in srm-conf.xml. Else, this will have to be added as part of data ingest.	No
Name	String	A meaningful (to the operator) name for this Srm Client. Must be unique.	Yes
UserAgent	String	Some dialects are specified by a combination of the <b>User-Agent</b> and <b>clabSessionGroup</b> headers (during setup). Required if this is one of those. If <b>clabSessionGroup</b> is null, then a <b>clabSessionGroup</b> must not be specified (any request which	No

		has a <code>clabSessionGroup</code> is not for this <code>SrmClient</code> . )	
Require	String	Some dialects are specified by the <b>Require</b> header. Required if this is one of those.	No
ClabSessionGroup	String	See UserAgent	No
KeepAlive	<b>true</b> or <b>false</b>	Whether the SRM is responsible for managing client keepalives in this workflow.	Yes
KeepAliveTimeout Secs	Integer	Where the SRM is responsible for managing client keepalives, the expected interval between keepalive calls, in seconds. Defaults to 30.	No
KeepaliveGrace PeriodSecs	Integer	Where the SRM is responsible for managing client keepalives, an additional interval after the expected keepalive interval to allow for network congestion or other stressors. Defaults to 5.	No
KeepaliveMisses	Integer	Where the SRM is responsible for managing client keepalives, how many keepalives the client is allowed to miss before the session is declared timed out. This allows for eg. network traffic spikes or transient network outages which may occasionally swallow a keepalive message. Defaults to 0.	No
RemoteAddress	String	<b>Reserved for future expansion.</b> For bandwidth resale (third-party SRM) only, the address to send RTSP traffic back to the third-party SRM.	No
RemotePort	Integer	<b>Reserved for future expansion.</b> For bandwidth resale (third-party SRM) only, the TCP port to send RTSP traffic back to the third-party SRM.	No
Description	String	Optional freeform text, useful to the operator.	No
SessionType	Fixed value: <b>VOD</b>	<b>Reserved for future expansion.</b> Must be <b>VOD</b> for now.	Yes
SessionVariant	<b>NGOD_s1</b> or <b>OTV</b>	The workflow required for this STB. Use <b>NGOD_s1</b> for NGOD S1 compatible STBs, <b>OTV</b> for OpenTV 2.4 compatible STBs.	Yes

EnableUser Limitation	boolean	Whether the value set against UserSessionLimit will be enforced when setting up the session. Defaults to false.	No
UserSessionLimit	int	Max number of users allowed per account. Defaults to 0	No

### 5.4.10 Data ingest: Sts device fields

An **STS** - STraming Server - sources video streams. Sometimes this is referred to as a VS (Video Server) or even a VSS.

**Note** | Conventionally "STS" is capitalised, since it's an abbreviation. However, when creating a data ingest, you must use the mixed-case form, **sts** .

Field	Type	Description	Required
Name	String	A unique name. <b>Caution!</b>   To avoid compatibility issues, this field should not contain spaces, semicolons, colons, or commas.	Yes
Address	String	IP address or DNS name for the device's management interface.	Yes
Port	Integer	TCP port to connect to.	Yes
CodeId	<b>NgodR2</b> or <b>Nagra</b> vs	The protocol to use when communicating with this STS. <b>Caution!</b>   This field is case sensitive and must be entered exactly as shown.	Yes
Status	<b>ACTIVE, INACTIVE, or DISABLED</b>	The status of the STS.	Yes

Status	Description
--------	-------------

Field	Type	Description	Required
		ACTIVE	The STS is fully available.
		INACTIVE	The STS is going offline soon. Existing sessions are left alone, but no new sessions will be started.
		DISABLED	The STS is offline. The SRM will disconnect from it, and will not attempt to re-establish a connection until its status changes.

**Caution!** In order to ensure that open sessions can be closed cleanly, some changes are forbidden. Any attempt to perform such a change will be rejected by the data ingest.

- ▶ An STS may not be changed from ACTIVE to DISABLED. Set it to INACTIVE first, and wait for all sessions to finish.
- ▶ An STS may not be changed from INACTIVE to DISABLED if there are active sessions on it.
- ▶ An STS may not be deleted if it is ACTIVE, or if there are active sessions on it.

### 5.4.11 Data ingest: Scs device fields

The **SimulCrypt Synchroniser (SCS)** is part of the Simulcrypt system to which the SRM communicates in order to support Session Based Scrambling (SBS) .

Typically the synchronizer is physically part of the QAM itself

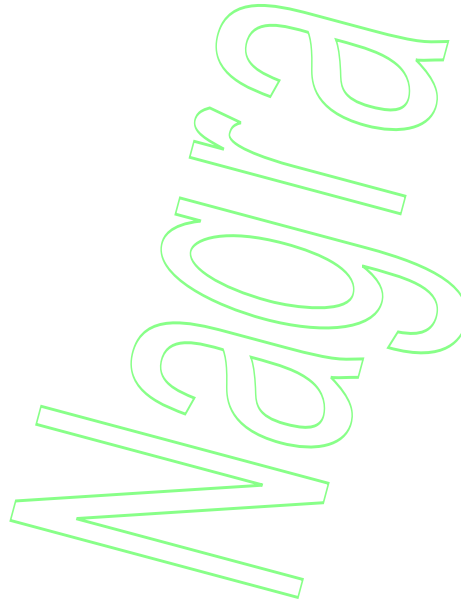
**Note**

Conventionally "SCS" is capitalised, since it's an abbreviation. However, when creating a data ingest, you must use the mixed-case form, **scs**.

Field	Type	Description	Required								
Name	String	A unique name.  <b>Caution!</b> To avoid compatibility issues, this field should not contain spaces, semicolons, colons, or commas.	Yes								
Address	String	IP address or DNS name for the device's management interface.	Yes								
Port	Integer	TCP port to connect to.	Yes								
CodeclId	String	The protocol to use when communicating with this SCS.  <b>Caution!</b> Currently not utilised	No								
Status	ACTIVE, INACTIVE, or DISABLED	The status of the SCS.	Yes								
		<hr/>									
		<table border="1"> <thead> <tr> <th>Status</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>ACTIVE</td> <td>The SCS is fully available.</td> </tr> <tr> <td>INACTIVE</td> <td>The SCS is going offline soon. Existing sessions are left alone, but no new sessions will be started.</td> </tr> <tr> <td>DISABLED</td> <td>The SCS is offline. The SRM will disconnect from it, and will not attempt to re-establish a connection until its status changes.</td> </tr> </tbody> </table>		Status	Description	ACTIVE	The SCS is fully available.	INACTIVE	The SCS is going offline soon. Existing sessions are left alone, but no new sessions will be started.	DISABLED	The SCS is offline. The SRM will disconnect from it, and will not attempt to re-establish a connection until its status changes.
		Status		Description							
ACTIVE	The SCS is fully available.										
INACTIVE	The SCS is going offline soon. Existing sessions are left alone, but no new sessions will be started.										
DISABLED	The SCS is offline. The SRM will disconnect from it, and will not attempt to re-establish a connection until its status changes.										
<hr/>											
		<b>Caution!</b> In order to ensure that open sessions can be closed cleanly, some changes are forbidden. Any attempt to perform such a change will be rejected by the data ingest.									



Field	Type	Description	Required
		<ul style="list-style-type: none"> <li>▶ An SCS may not be changed from ACTIVE to DISABLED. Set it to INACTIVE first, and wait for all sessions to finish.</li> <li>▶ An SCS may not be changed from INACTIVE to DISABLED if there are active sessions on it.</li> <li>▶ An SCS may not be deleted if it is ACTIVE, or if there are active sessions on it.</li> </ul>	



## 6 SRM API reference

### 6.1 Operator and Management APIs

#### 6.1.1 Overall URL structure for web UI

##### Overview

URLs on the SRM's web UI are arranged in a structural heirarchy, by URL path. The key divisions are as shown.

URL path prefix	Description
/srm/crud/	Data I/O
/srm/force-disconnect/	Forcibly tear down healthy sessions administratively
/srm/reporting/	Business reporting

#### 6.1.2 Bulk data ingest

##### Rationale

This API provides the way to ingest data into the SRM.

##### Request

###### HTTP Method

This API uses HTTP POST requests.

###### URI

Uses a standard URL:

`http://host:5580/srm/crud/bulk-ingest`

###### Additional request headers

Name	Format	Description	Required?
x-correlation-id	String	Externally-generated request identifier. Used to tie together the sections of a distributed request. If present, will be returned in the response. If missing, will be generated internally and returned in the response.	No

**Access control**

HTTP BASIC authentication is required. By default, the required username is **operations**.

**Request Body**

The request body is in **multipart/form-data** format.

Form element name	Format	Description	Required
csvdata	File	CSV data, compliant with RFC 4180. The format of the CSV data is given <a href="#">Data ingest file format</a> <sup>p.63</sup> .	Yes
version	String	The API version. Currently only version 1 is supported.	No - defaults to version 1
prettyPrint	String	Whether to pretty-print the response. If set to "true" or "prettyPrint" the output will be pretty-printed (ie. legible to humans). Otherwise a compact representation is returned.	No - defaults to empty
(Other elements)	n/a	Ignored	No

**Access control**

HTTP BASIC authentication is required. By default, the required username is **operations**.

**Success response**

On success or partial success, the response body is a JSON map.

Example:

```
HTTP/1.1 200 OK
[...]
Content-Type: application/json;charset=UTF-8
```

```
{
  "fileSummary" : {
    "fileName" : "guangzhou-delta-20130817.csv",
    "fileSize" : 10484,
    "ingestDate" : "2013-08-17T14:30:08+0800",
    "shalsum" : "4d7f47873452448e54a417fffe473463343f34d2"
  },
  "parseSummary" : {
    "Erm" : {
      "toCreate" : 1,
      "toUpdate" : 2,
      "toDelete" : 0
    },
    "ServiceGroup" : {
      "toCreate" : 2,
      "toUpdate" : 0,
      "toDeactivate" : 1,
      "toDelete" : 0
    }
  },
  "information" : [
    "Line 32: missing 4 data columns, using blank (your CSV encoder is nonstandard)"
  ],
  "warnings" : [
    "Line 31: ignoring 1 additional data columns"
  ],
  "errors" : [
    {
      "recordType" : "Erm",
      "ingestName" : "S6-Guangzhou-44",
      "status" : "ERROR",
      "errorCode" : "SRM-0003",
      "errorMessage" : "Invalid codecId NhodS6"
    }
  ],
}
```

**Entry**

**Description**

fileSummary

Information about the submitted file. Fields:

**Name**

**Description**

fileName

Filename as submitted in the ingest (if available).

Entry	Description								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>fileSize</td> <td>File size in bytes. May be useful for detecting transmission errors.</td> </tr> <tr> <td>ingestDate</td> <td>When the file was received. ISO date format. Timezone is the systemTimezone, as specified in <a href="#">Configuration bean: srmCoreConfig</a> <sup>p.48</sup>.</td> </tr> <tr> <td>sha1sum</td> <td>Hex-encoded SHA-1 hash ("checksum") of the file, as received. This is a very reliable way of detecting transmission errors.</td> </tr> </tbody> </table> <p><b>Note</b> Use the <code>sha1sum</code> command, or its equivalent on non-Linux machines, to determine the sha1sum of the file before sending. If the two don't match, the file was corrupted in transmission.</p>	Name	Description	fileSize	File size in bytes. May be useful for detecting transmission errors.	ingestDate	When the file was received. ISO date format. Timezone is the systemTimezone, as specified in <a href="#">Configuration bean: srmCoreConfig</a> <sup>p.48</sup> .	sha1sum	Hex-encoded SHA-1 hash ("checksum") of the file, as received. This is a very reliable way of detecting transmission errors.
Name	Description								
fileSize	File size in bytes. May be useful for detecting transmission errors.								
ingestDate	When the file was received. ISO date format. Timezone is the systemTimezone, as specified in <a href="#">Configuration bean: srmCoreConfig</a> <sup>p.48</sup> .								
sha1sum	Hex-encoded SHA-1 hash ("checksum") of the file, as received. This is a very reliable way of detecting transmission errors.								
parseSummary	For each known record type, the number of records of that type which were submitted for every applicable operation.								
information	A list of informative messages. These are typically generated in response to minor CSV formatting violations which the ingester was able to work around.								
warnings	A list of more serious problems, which were not serious enough to cause a record to be rejected.								
errors	<p>A list of structured error reports. The specified record has been rejected. The status, errorCode and errorMessage fields have the same meanings as for failure responses from other REST services.</p> <p>The usual list of <a href="#">Error Codes and status values</a> <sup>p.313</sup> applies.</p> <p>The example shows an <b>Erm</b> record which was rejected because of a typo in the codeId field (It should have been "NgodS6" not "NhodS6").</p>								

## Failure response

On a complete failure, the HTTP response code will be set to the most appropriate value (drawn from the set of standard HTTP response codes).

The response body is the JSON encoding of an Exception object which describes the failure. No records have been ingested.

### Example

```
HTTP/1.1 400 Bad Request
x-correlation-id:88bc514e-5425-4a23-92db-7b8ddedcd745
Transfer-Encoding:chunked
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:17:22 GMT
Content-Type:application/json;charset=UTF-8
Connection:close

{
  "errorResponse": {
    "status": "ERROR",
    "errorCode": "SRM-0002",
    "errorMessage": "Expected a .csv file, but the file you sent was called test.xml - did you send the correct file?"
  }
}
```

## 6.1.3 Force disconnect

### Rationale

#### Note

These APIs enable an operator to forcibly terminate sessions for the specified criteria. It's based on the MultiScreen <https://atlassian.hq.k.grp/confluence/display/MSARCH/REST+API+conventions>.

This request is asynchronous and the sessions will be queued - so session removal will not be immediate.

Purpose	URI
Terminate by asset into	<code>http://host:5580/srm/force-disconnect/asset</code>
Terminate by sessionuuid	<code>http://host:5580/srm/force-disconnect/sessionUuid</code>

Purpose	URI
Terminate by Account Uid	http://host:5580/srm/force-disconnect/account

### 6.1.3.1 Force Disconnect - Account

**Note**

Forced Disconnection is the ability to clear/teardown sessions based on the criteria specified. For example, removal of all sessions for account 1. This request is asynchronous and the sessions will be queued - so session removal will not be immediate.

#### Request

**HTTP Method**

This API uses HTTP POST requests.

**URI**

Uses a standard URL with query parameters:

http://host:5580/srm/force-disconnect/account?<parameters>

**Parameters**

Name	Format	Description	Required?
accountNumber	String	The account number of the account	No
accountUid	Number	The internal MediaLive UID of the account	No

**Note**

Either account number or account uid should be specified. If both are specified then accountUid will take preference.

At the moment accountNumber is not supported

**Additional request headers**

Name	Format	Description	Required?
x-correlation-id	String	Externally-generated request identifier. Used to tie together the sections of a distributed request.	No

Name	Format	Description	Required?
		If present, will be returned in the response. If missing, will be generated internally and returned in the response.	

**Access control**

HTTP BASIC authentication is required. By default, the required username is `nagra-agent`.

**Example**

```
POST /srm/force-disconnect/account?accountNumber=123
Host: nagra.com
Authorization: Basic bmFncmEtYWdlbnQ6YXJnYW5fJHRkZDI1NDAx
x-correlation-id: 4793w850-guangzhou
```

**Success response**

On success, the response body is a JSON object. This JSON object contains the total number of sessions which will be disconnected.

**Note**

NB. The call returns quickly. The actual disconnections happen asynchronously (ie. later). If there are a lot of sessions to disconnect, this may take a long time.

**Example**

```
200 OK HTTP/1.1
x-correlation-id:f1ad716d-0629-4632-b4ac-16dbd7bc63a1
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:14:32 GMT
Content-Type:application/json;charset=UTF-8
Content-Length:34

{ "disconnectedSessionCount":0 }
```

**Failure response**

On failure, the HTTP response code will be set to the most appropriate value (drawn from the set of standard HTTP response codes).

The response body is the JSON encoding of an Exception object which describes the failure.

**Example**

```
HTTP/1.1 400 Bad Request
```



```
x-correlation-id:88bc514e-5425-4a23-92db-7b8ddedcd745
Transfer-Encoding:chunked
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:17:22 GMT
Content-Type:application/json;charset=UTF-8
Connection:close

{
  "errorResponse": {
    "status": "ERROR",
    "errorCode": "SRM-0002",
    "errorMessage": "Missing mandatory parameter: accountId"
  }
}
```

### 6.1.3.2 Force Disconnect - Asset

#### Request

##### HTTP Method

This API uses HTTP POST requests.

##### URI

Uses a standard URL with query parameters:

`http://host:5580/srm/force-disconnect/asset?<parameters>`

##### Note

In the future, we may need to be able to force disconnect by other criteria. For example it may be necessary to disconnect all sessions which are using a particular device, in order to perform emergency maintenance on that device.

##### Parameters

Name	Format	Description	Required?
providerId	String	PAID	Yes
assetId	String	PAID	Yes
version	String	API version. Currently only version 1 is supported.	No - defaults to 1

##### Additional request headers

Name	Format	Description	Required?
x-correlation-id	String	Externally-generated request identifier. Used to tie together the sections of a distributed request. If present, will be returned in the response. If missing, will be generated internally and returned in the response.	No

**Access control**

HTTP BASIC authentication is required. By default, the required username is `nagra-agent`.

**Example**

```
POST /srm/force-disconnect/asset?providerId=EuroTV&assetId=Cyprus-2013-03-19
Host: srm04.guangzhou.example.tv
Authorization: Basic bmFncmEtYWdlbnQ6YXJnYW5fJHRkZDI1NDAx
x-correlation-id: 4793w850-guangzhou
```

**Success response**

On success, the response body is a JSON object. This JSON object contains the total number of sessions which will be disconnected.

**Note**

NB. The call returns quickly. The actual disconnections happen asynchronously (ie. later). If there are a lot of sessions to disconnect, this may take a long time.

**Example**

```
200 OK HTTP/1.1
x-correlation-id:f1ad716d-0629-4632-b4ac-16dbd7bc63a1
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:14:32 GMT
Content-Type:application/json;charset=UTF-8
Content-Length:34

{ "disconnectedSessionCount":0 }
```

**Failure response**

On failure, the HTTP response code will be set to the most appropriate value (drawn from the set of standard HTTP response codes).

The response body is the JSON encoding of an Exception object which describes the failure.

**Example**

```
HTTP/1.1 400 Bad Request
x-correlation-id:88bc514e-5425-4a23-92db-7b8ddedcd745
Transfer-Encoding:chunked
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:17:22 GMT
Content-Type:application/json;charset=UTF-8
Connection:close

{
  "errorResponse": {
    "status": "ERROR",
    "errorCode": "SRM-0002",
    "errorMessage": "Missing mandatory parameter: providerId"
  }
}
```

### 6.1.3.3 Force Disconnect - Session Uuid

This request allows the operator to terminate a specific request via the session-uuid

#### Request

##### HTTP Method

This API uses HTTP POST requests.

##### URI

Uses a standard URL with query parameters:

<http://host:5580/srm/force-disconnect/sessionUuid<parameters>>

##### Parameters

Name	Format	Description	Required?
uuid	String	The UUID number of the session	Yes

##### Additional request headers

Name	Format	Description	Required?
x-correlation-id	String	Externally-generated request identifier. Used to tie together the sections of a distributed request. If present, will be returned in the response.	No

Name	Format	Description	Required?
		If missing, will be generated internally and returned in the response.	

### Access control

HTTP BASIC authentication is required. By default, the required username is **nagra-agent**.

### Example

```
POST /srm/force-disconnect/sessionUuid?uuid=9aaa29bc-6cf2-4655-962c-dc862d21e94e
Host: nagra.com
Authorization: Basic bmFncmEtYWdlbnQ6YXJnYW5fJHRkZDI1NDAx
x-correlation-id: 4793w850-guangzhou
```

### Success response

On success, the response body is a JSON object. This JSON object contains the total number of sessions which will be disconnected.

### Note

NB. The call returns quickly. The actual disconnections happen asynchronously (ie. later). If there are a lot of sessions to disconnect, this may take a long time.

### Example

```
200 OK HTTP/1.1
x-correlation-id:f1ad716d-0629-4632-b4ac-16dbd7bc63a1
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:14:32 GMT
Content-Type:application/json;charset=UTF-8
Content-Length:34

{ "disconnectedSessionCount":0 }
```

### Failure response

On failure, the HTTP response code will be set to the most appropriate value (drawn from the set of standard HTTP response codes).

The response body is the JSON encoding of an Exception object which describes the failure.

### Example

```
HTTP/1.1 400 Bad Request
x-correlation-id:88bc514e-5425-4a23-92db-7b8ddedcd745
```

```
Transfer-Encoding:chunked
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:17:22 GMT
Content-Type:application/json;charset=UTF-8
Connection:close

{
  "errorResponse": {
    "status": "ERROR",
    "errorCode": "SRM-0002",
    "errorMessage": "Missing mandatory parameter: uuid"
  }
}
```

## 6.1.4 Machine diagnostics

The monitoring system has a separate module for monitoring the host and OS. However, in situations where this module is not readily available (eg. testing), it can be useful to expose some of this information via the SRM itself.

### Note

This API is beyond the scope of the [Monitoring statistics](#) <sup>p.99</sup> API (since the monitoring system should not use it).

### Values

The API exposes three values. These are instantaneous readings, taken at the time of the request (they are not polled in the way that a true [Monitored quantity](#) <sup>p.33</sup> is).

Value	Description
freeMemory	The amount of free memory available to the JVM, in KB.  <b>Note</b> This may vary in an irregular or "saw-tooth" fashion due to the action of the garbage collector.
loadAverage	The system <a href="http://en.wikipedia.org/wiki/Load_%28computing%29">http://en.wikipedia.org/wiki/Load %28computing%29</a> over the last minute.
cpuCount	The number of CPU cores available to the system. Typically this remains constant. However, it could vary in a virtualized system.

## Request

### HTTP Method

This API uses HTTP GET requests.

### URI

Uses a standard URL with optional query parameters:

`http://host:5580/srm/diagnostics/machine[?parameters]`

### Query parameters

All query parameters are optional - they have default values.

Parameter	Description	Default value
<code>version</code>	API version. Only version 1 is supported.	1
<code>prettyPrint</code>	Whether to pretty-print the response. <code>true</code> or <code>false</code> .	<code>false</code>

### Request body

The request should not contain a body. Any body it does contain is ignored.

## Success response

The response is a JSON map containing all the values, in a similar format to the [Monitoring statistics](#) <sup>p.99</sup> API.

For example:

```
HTTP/1.1 200 OK
[...]
Content-Type: application/json;charset=UTF-8

{
  "id": "SRM:/srm/diagnostics/machine",
  "apiVersion": 1,
  "freeMemory": 32477,
  "loadAverage": 3.4,
  "cpuCount": 8
}
```

### Note

Layout shown is for readability.

## Failure response

On failure, the HTTP response code will be set to the most appropriate value (drawn from the set of standard HTTP response codes).

The response body is a JSON map containing error information.

```
HTTP/1.1 500 Internal Server Error
[...]
Content-Type:application/json;charset=UTF-8
Connection:close

{
  "errorResponse" : {
    "status": "INTERNAL_ERROR",
    "errorCode": "SRM-9999",
    "errorMessage": "SecurityException: No access to system MXBeans"
  }
}
```

## 6.1.5 Monitoring device health

This API exposes the current health status of any active device. (Inactive devices are not available).

### Request

#### HTTP Method

This API uses HTTP GET requests.

#### URI

Uses a standard URL with optional query parameters:

`http://host:5580/srm/monitoring/health[/<device-name>][?parameters]`

If the device name is provided, only results for the specified device will be returned.

#### Query parameters

All query parameters are optional - they have default values.

Parameter	Description	Default value
<code>version</code>	API version. Only version 1 is supported.	1
<code>prettyPrint</code>	Whether to pretty-print the response. <code>true</code> or <code>false</code> .	<code>false</code>

#### Request body

The request should not contain a body. Any body it does contain is ignored.

#### Success response

If no device-name is specified in the request URI, the response is a JSON list containing an [Monitored device health status](#) <sup>p.40</sup> for each device.

For example:

```
HTTP/1.1 200 OK
[...]
Content-Type: application/json;charset=UTF-8

[
  {
    "id": "SRM:/srm/monitoring/health/AppServer_Myself",
    "apiVersion": 1,
    "status": "VALID",
    "faultDuration": 0,
    "faultReason": "",
    "recoverAfter": ""
  },
  {
    "id": "SRM:/srm/monitoring/health/Erm_Ermest_1_Gobelins",
    "apiVersion": 1,
    "status": "VALID",
    "faultReason": "",
    "faultDuration": 0,
    "recoverAfter": ""
  },
  {
    "id": "SRM:/srm/monitoring/health/Sts_Streamer_1_Gobelins",
    "apiVersion": 1,
    "status": "FAULT",
    "faultReason": "ConnectException: Request timed out",
    "faultDuration": 134,
    "recoverAfter": "2013-06-05T20:13:00+0800"
  },
  {
    "id": "SRM:/srm/monitoring/health/Pav_Local",
    "apiVersion": 1,
    "status": "VALID",
    "faultReason": "",
    "faultDuration": 0,
    "recoverAfter": ""
  }
]
```

**Note**

Layout shown is for readability.

The devices are sorted by type: AppServer first, then Erm, Sts, and finally Pav. The devices are sorted by name within each type.



If a device-name is specified, only the single entry is returned (rather than an outer list which only contains one entry).

### Failure response

On failure, the HTTP response code will be set to the most appropriate value (drawn from the set of standard HTTP response codes).

The response body is a JSON map containing error information.

```
HTTP/1.1 404 Not Found
[...]
Content-Type:application/json;charset=UTF-8
Connection:close

{
  "errorResponse" : {
    "status": "ERROR",
    "errorCode": "SRM-0003",
    "errorMessage": "No such device"
  }
}
```

## 6.1.6 Monitoring device list

Returns the list of devices. This list varies, but typically only quite slowly. It only changes in response to operator actions.

### Request

#### HTTP Method

This API uses HTTP GET requests.

#### URI

Uses a standard URL with optional query parameters:

```
http://host:5580/srm/monitoring/device-list[<device type>][?parameters]
```

To obtain the list of devices which are available for [Monitoring device health](#) <sup>p.95</sup>, omit the device type. (All devices will be returned).

Otherwise, a filtered list will be returned. Use filtered lists for [Monitoring statistics](#) <sup>p.99</sup>: the device type defines which quantities (and hence, which counters) are available.

Device type	Returns
contributor	All <a href="#">Contributor device</a> <sup>p.10</sup> records

Device type	Returns
pav	All <a href="#">Pav device</a> <sup>p.11</sup> records

**Note** | There is no need to enumerate [SrmServer device](#) <sup>p.11</sup> records - they don't have any monitored quantities.

**Query parameters**

All query parameters are optional - they have default values.

Parameter	Description	Default value
version	API version. Only version 1 is supported.	1
prettyPrint	Whether to pretty-print the response, true or false.	false

**Request body**

The request should not contain a body. Any body it does contain is ignored.

**Success response**

The response is a JSON list containing an entry for each known object. This is contained in an outer wrapper which also specifies the API version.

**Note** | Devices which were marked as DISABLED when they were ingested are not returned in the list.

Field	Description
name	The name of the object. Use this to specify the object in other monitoring APIs.

For example:

```
HTTP/1.1 200 OK
[...]
Content-Type: application/json;charset=UTF-8
{
  "id" : "SRM:/srm/monitoring/device-list"
  "apiVersion" : 1,
  "deviceNames" : [
    "Erm_Ermest_1_Gobelins",
```

```

    "Sts_Streamer_1_Gobelins",
    "AppServer_Myself",
    "Pav_Local"
  ]
}

```

**Note**

Layout shown is for readability.

### Failure response

On failure, the HTTP response code will be set to the most appropriate value (drawn from the set of standard HTTP response codes).

The response body is a JSON map containing error information.

```

HTTP/1.1 404 Not Found
[...]
Content-Type:application/json;charset=UTF-8
Connection:close

{
  "errorResponse" : {
    "status": "ERROR",
    "errorCode": "SRM-0003",
    "errorMessage": "No such device type"
  }
}

```

## 6.1.7 Monitoring statistics

This API exposes statistics from any [Monitored object](#)<sup>p.35</sup> which offers a [Monitored quantity](#)<sup>p.33</sup>.

### Request

#### HTTP Method

This API uses HTTP GET requests.

#### URI

Uses a standard URL with optional query parameters:

`http://host:5580/srm/monitoring/statistics[/<object-name>][/<period-name>][?parameters]`

If an object name is provided, only results for the specified object will be returned. Valid object names are:

- ▶ The name of any [Contributor device](#) <sup>p.10</sup> or [Pav device](#) <sup>p.11</sup>. Use the [Monitoring device list](#) <sup>p.97</sup> API to obtain these names.
- ▶ The name of a [Monitored interface](#) <sup>p.36</sup>.
- ▶ The name of the [Monitored session manager](#) <sup>p.37</sup> or [Monitored connection manager](#) <sup>p.38</sup>.

If a period name is provided, only results for the specified viewing period will be returned. The period name is one of the [Monitored viewing periods](#) <sup>p.33</sup>.

### Query parameters

All query parameters are optional - they have default values.

Parameter	Description	Default value
version	API version. Only version 1 is supported.	1
prettyPrint	Whether to pretty-print the response. <b>true</b> or <b>false</b> .	<b>false</b>

### Request body

The request should not contain a body. Any body it does contain is ignored.

## Success response

### No filtering

The response is a JSON list containing a JSON structure for each [Monitored object](#) <sup>p.35</sup>.

For example:

```
HTTP/1.1 200 OK
[...]
Content-Type: application/json;charset=UTF-8

[
  {
    "id" : "SRM:/srm/monitoring/statistics/Erm_Ernest_1_Gobelins",
    "apiVersion" : 1,
    "lastMinute" : {
      "setupExecution" : {
        "successCounter" : 79,
        "failedCounter" : 3,
        "totalCounter" : 82,
        "minTime" : 38,
        "maxTime" : 414,
        "histogram" : {
          "category" : [ "normal", "acceptable", "warning", "critical" ],
          "counter" : [ 42, 35, 5, 0 ],
          "range" : [ "<50ms", "50to149ms", "150to499ms", ">500ms" ]
        }
      }
    }
  }
]
```

```

    },
    "teardownExecution" : {
      [...]
    }
  },
  "lastHour" : { [...] },
  "lastDay" : { [...] }
},
{
  "id" : "SRM:/srm/monitoring/statistics/Pav_Local",
  "apiVersion" : 1,
  "lastMinute" : {
    "pavExecution" : {
      "successCounter" : 294,
      "failedCounter" : 5,
      "totalCounter" : 299,
      "minTime" : 41,
      "maxTime" : 570,
      "histogram" : {
        "category" : [ "fast", "normal", "slow", "warning", "critical" ],
        "counter" : [ 7, 206, 84, 2, 0 ],
        "range" : [ "<100ms", "100to199ms", "200to399ms", "400to999ms",
">1000ms" ]
      }
    }
  },
  "lastHour" : { [...] },
  "lastDay" : { [...] }
}
]

```

**Note**

Layout shown is for readability.

**Single object**

If an object is specified, then only one list entry is shown, and the outer list is omitted. For example:

```

HTTP/1.1 200 OK
[...]
Content-Type: application/json;charset=UTF-8

{
  "id" : "SRM:/srm/monitoring/statistics/Erm_Ernest_1_Gobelins",
  "apiVersion" : 1,
  "lastMinute" : {
    "setupExecution" : {
      "successCounter" : 79,
      "failedCounter" : 3,

```

```

    "totalCounter" : 82,
    "minTime" : 38,
    "maxTime" : 414,
    "histogram" : {
      "category" : [ "normal", "acceptable", "warning", "critical" ],
      "counter" : [ 42, 35, 5, 0 ],
      "range" : [ "<50ms", "50to149ms", "150to499ms", ">500ms" ]
    }
  },
  "teardownExecution" : {
    [...]
  }
},
"lastHour" : { [...] },
"lastDay" : { [...] }
}

```

**Note**

Layout shown is for readability.

**Single period**

If only a single period is wanted, the structure and returned id are adjusted accordingly. For example:

```

HTTP/1.1 200 OK
[...]
Content-Type: application/json;charset=UTF-8
{
  "id" : "SRM:/srm/monitoring/statistics/Erm_Ermest_1_Gobelins/lastMinute",
  "apiVersion" : 1,
  "setupExecution" : {
    "successCounter" : 79,
    "failedCounter" : 3,
    "totalCounter" : 82,
    "minTime" : 38,
    "maxTime" : 414,
    "histogram" : {
      "category" : [ "normal", "acceptable", "warning", "critical" ],
      "counter" : [ 42, 35, 5, 0 ],
      "range" : [ "<50ms", "50to149ms", "150to499ms", ">500ms" ]
    }
  },
  "teardownExecution" : {
    [...]
  }
}

```

**Note**

Layout shown is for readability.

## Failure response

On failure, the HTTP response code will be set to the most appropriate value (drawn from the set of standard HTTP response codes).

The response body is a JSON map containing error information.

```
HTTP/1.1 404 Not Found
[...]
Content-Type:application/json;charset=UTF-8
Connection:close

{
  "errorResponse" : {
    "status": "ERROR",
    "errorCode": "SRM-0003",
    "errorMessage": "No such monitored object"
  }
}
```

## 6.1.8 Session Discovery

### Rationale

There is a requirement for operators to retrieve specific session information.

These APIs provides an automated way of retrieving sessions for the specified criteria. It's based on the MultiScreen <https://atlassian.hq.k.grp/confluence/display/MSARCH/REST+API+conventions>.

Purpose	URI
Retrieve by Account UID	<code>http://host:5580/&lt;tenant-name&gt;/srm/session-discovery/account?accountUid=</code>
Retrieve by Account Number	<code>http://host:5580/&lt;tenant-name&gt;/srm/session-discovery/account?accountNumber=</code>
Retrieve by Smartcard	<code>http://host:5580/&lt;tenant-name&gt;/srm/session-discovery/session?smartcard=</code>
Retrieve session based on vsSessionId	<code>http://host:5580/&lt;tenant-name&gt;/srm/session-discovery/session?vsSessionId=</code>
Retrieve sessions sharing the network resource based on the vs-session	<code>http://host:5580/&lt;tenant-name&gt;/srm/session-discovery/session?vsSessionId=&amp;shared=TRUE</code>

**Note**

The SRM is currently only aware of the account UID. In a future release this api will extend to perform the lookup based on account number also.

### 6.1.8.1 Session Discovery - Account

#### Request

##### HTTP Method

This API uses HTTP GET requests.

##### URI

Uses a standard URL with query parameters:

http://host:5580/srm/session-discovery/account?<parameters>

##### Parameters

Name	Format	Description	Required?
accountNumber	String	The account number of the account	No
accountUid	Number	The internal MediaLive UID of the account	No

**Note**

Either account number or account uid should be specified. If both are specified then accountUid will take preference.

At the moment accountNumber is not supported

##### Additional request headers

Name	Format	Description	Required?
x-correlation-id	String	Externally-generated request identifier. Used to tie together the sections of a distributed request. If present, will be returned in the response. If missing, will be generated internally and returned in the response.	No

##### Access control

HTTP BASIC authentication is required. By default, the required username is **nagra-agent**.



### Example

```
GET /srm/force-disconnect/account?accountUid=1341
Host: nagra.com
Authorization: Basic bmFncmEtYWdlbnQ6YXJnYW5fJHRkZDI1NDAx
x-correlation-id: 4793w850-guangzhou
```

### Success response

On success, the response body is a JSON object. This JSON object contains the sessions specific to that account.

#### Example

```
200 OK HTTP/1.1
x-correlation-id:flad716d-0629-4632-b4ac-16dbd7bc63a1
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:14:32 GMT
Content-Type:application/json;charset=UTF-8
Content-Length:34

[{"uuid":"2b6a1ba5-5082-4b29-b96d-f82dae26ac38","sessionType":"VOD","sessionVariant":"NAGRA_OTT","accountUid":1341,"deviceUid":1433},{ "uuid":"95a3bf0e-161c-4391-8e03-0b71b6cb198d","sessionType":"VOD","sessionVariant":"NAGRA_OTT","accountUid":1341,"deviceUid":1433}]
```

### Failure response

On failure, the HTTP response code will be set to the most appropriate value (drawn from the set of standard HTTP response codes).

The response body is the JSON encoding of an Exception object which describes the failure.

#### Example

```
HTTP/1.1 400 Bad Request
x-correlation-id:88bc514e-5425-4a23-92db-7b8ddedcd745
Transfer-Encoding:chunked
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:17:22 GMT
Content-Type:application/json;charset=UTF-8
Connection:close

{
  "errorResponse": {
    "status": "ERROR",
    "errorCode": "SRM-0002",
```

```
"errorMessage": "Missing mandatory parameter: accountUid"
  }
}
```

## 6.1.8.2 Session Discovery - smartcard

### Request

#### HTTP Method

This API uses HTTP GET requests.

#### URI

Uses a standard URL with query parameters:

`http://host:5580/srm/session-discovery/session?<parameters>`

#### Parameters

Name	Format	Description	Required?
smartcard	String	Smartcard ID of a Client Device	Yes

#### Additional request headers

Name	Format	Description	Required?
x-correlation-id	String	Externally-generated request identifier. Used to tie together the sections of a distributed request. If present, will be returned in the response. If missing, will be generated internally and returned in the response.	No

#### Access control

HTTP BASIC authentication is required. By default, the required username is `nagra-agent`.

#### Example

```
GET /srm/session-discovery/session?smartcard=123456789
Host: nagra.com
Authorization: Basic bmFncmEtYWdlbnQ6YXJnYW5fJHRkZDI1NDAx
x-correlation-id: 4793w850-guangzhou
```

## Success response

On success, the response body is a JSON object. This JSON object contains the session specific to that smartcard.

Example

```
200 OK HTTP/1.1
x-correlation-id:flad716d-0629-4632-b4ac-16dbd7bc63a1
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:14:32 GMT
Content-Type:application/json;charset=UTF-8
Content-Length:34

[{"uuid":"2b6a1ba5-5082-4b29-b96d-f82dae26ac38","sessionType":"VOD","sessionVariant":"NCSDV","accountUid":1341,"deviceUid":1433},{ "uuid":"95a3bf0e-161c-4391-8e03-0b71b6cb198d","sessionType":"VOD","sessionVariant":"NCSDV","accountUid":1341,"deviceUid":1433}]
```

## Failure response

On failure, the HTTP response code will be set to the most appropriate value (drawn from the set of standard HTTP response codes).

The response body is the JSON encoding of an Exception object which describes the failure.

Example

```
HTTP/1.1 400 Bad Request
x-correlation-id:88bc514e-5425-4a23-92db-7b8ddedcd745
Transfer-Encoding:chunked
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:17:22 GMT
Content-Type:application/json;charset=UTF-8
Connection:close

{
  "errorResponse": {
    "status": "ERROR",
    "errorCode": "SRM-0002",
    "errorMessage": "Missing mandatory parameter: smartdcard"
  }
}
```

### 6.1.8.3 Session Discovery - vsession

## Request

### HTTP Method

This API uses HTTP GET requests.

### URI

Uses a standard URL with query parameters:

`http://host:5580/srm/session-discovery/session?<parameters>`

### Parameters

Name	Format	Description	Required?
vsSessionId	String	Session ID of a videosever session	Yes
shared	Boolean	Specify TRUE to retrieve all sessions that share the same network resource. If not specified this will default to FALSE	No

### Additional request headers

Name	Format	Description	Required?
x-correlation-id	String	Externally-generated request identifier. Used to tie together the sections of a distributed request. If present, will be returned in the response. If missing, will be generated internally and returned in the response.	No

### Access control

HTTP BASIC authentication is required. By default, the required username is `nagra-agent`.

### Example

```
GET /srm/session-discovery/session?vsSessionId=123456789&shared=TRUE
Host: nagra.com
Authorization: Basic bmFncmEtYWdlbnQ6YXJnYW55fjRkZDI1NDAx
x-correlation-id: 4793w850-guangzhou
```

## Success response

On success, the response body is a JSON object. This JSON object contains the session specific to that vsSessionId.

Example

```
200 OK HTTP/1.1
x-correlation-id:flad716d-0629-4632-b4ac-16dbd7bc63a1
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:14:32 GMT
Content-Type:application/json;charset=UTF-8
Content-Length:34

[{"uuid":"2b6a1ba5-5082-4b29-b96d-f82dae26ac38","sessionType":"VOD","sessionVariant":"NCSDV","accountUid":1341,"deviceUid":1433},{"uuid":"95a3bf0e-161c-4391-8e03-0b71b6cb198d","sessionType":"VOD","sessionVariant":"NCSDV","accountUid":1341,"deviceUid":1433}]
```

## Failure response

On failure, the HTTP response code will be set to the most appropriate value (drawn from the set of standard HTTP response codes).

The response body is the JSON encoding of an Exception object which describes the failure.

### Example

```
HTTP/1.1 400 Bad Request
x-correlation-id:88bc514e-5425-4a23-92db-7b8ddedcd745
Transfer-Encoding:chunked
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:17:22 GMT
Content-Type:application/json;charset=UTF-8
Connection:close

{
  "errorResponse": {
    "status": "ERROR",
    "errorCode": "SRM-0002",
    "errorMessage": "Missing mandatory parameter: vsSessionId"
  }
}
```

## 6.1.9 Data Discovery

### Rationale

There is a requirement for operators to retrieve specific information from the database.

These APIs provides an way of retrieving data for the specified criteria. It's based on the <https://engwiki.hq.k.grp/pages/viewpage.action?pagelid=118276042>.

Purpose	URI	Output
Retrieve by channel name	<code>http://host:5580/srm/data-discovery/channel/&lt;name&gt;</code>	channel + associated channel resources

### 6.1.9.1 Data Discovery - Channel

#### Request

##### HTTP Method

This API uses HTTP GET requests.

##### URI

Uses a standard URL with query parameters:

`http://host:5580/srm/data-discovery/channel/<parameter>`

##### Parameters

Name	Format	Description	Required?
name	String	The unique Channel name	yes

##### Additional request headers

Name	Format	Description	Required?
x-correlation-id	String	Externally-generated request identifier. Used to tie together the sections of a distributed request. If present, will be returned in the response. If missing, will be generated internally and returned in the response.	No

##### Access control

HTTP BASIC authentication is required.

##### Example

```
GET /srm/data-discovery/channel/<name>
```

```
Host: nagra.com
Authorization: Basic bmFncmEtYWdlbnQ6YXJnYW5fJHRkZDI1NDAx
x-correlation-id: 4793w850-guangzhou
```

## Success response

On success, the response body is a JSON object. This JSON object contains the sessions specific to that Channel + associated ServiceGroupResources.

Example

```
200 OK HTTP/1.1
x-correlation-id:f1ad716d-0629-4632-b4ac-16dbd7bc63a1
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:14:32 GMT
Content-Type:application/json;charset=UTF-8
Content-Length:34
```

```
{
  "Resources": [
    {
      "rfNgodS6Name": "default-qam",
      "rfFreqNo": 103,
      "rfFreqHz": 51000000,
      "rfDvbNetworkId": 64753,
      "rfDvbTsid": 61223,
      "rfSymPerSec": 6400000,
      "rfModulation": 64,
      "rfInnerFec": 3,
      "rfOuterFec": 7,
      "dvbServiceId": 1,
      "destIpAddress": "127.0.0.1",
      "destUdpPort": 6100,
      "allocBps": 0,
      "sgIngestName": "Gobelins-201",
      "chanName": "Gobelins-20101",
      "status": "AVAILABLE"
    },
    {
      "rfNgodS6Name": "default-qam",
      "rfFreqNo": 101,
      "rfFreqHz": 6100000000,
      "rfDvbNetworkId": 49723,
      "rfDvbTsid": 61230,
      "rfSymPerSec": 6400000,
      "rfModulation": 3,
      "rfInnerFec": 3,
      "rfOuterFec": 7,
      "dvbServiceId": 62,

```

```

        "destIpAddress": "10.200.0.3",
        "destUdpPort": 6105,
        "allocBps": 0,
        "sgIngestName": "65753.202",
        "chanName": "Gobelins-20101",
        "name": "Q2Gob-20201-62",
        "status": "AVAILABLE"
    },
    ],
    "name": "Gobelins-20101",
    "sgIngestName": "Gobelins-201",
    "maxBps": 38000000,
    "currentBps": 0
}

```

## Failure response

On failure, the HTTP response code will be set to the most appropriate value (drawn from the set of standard HTTP response codes).

The response body is the JSON encoding of an Exception object which describes the failure.

### Example

```

HTTP/1.1 400 Bad Request
x-correlation-id:88bc514e-5425-4a23-92db-7b8ddedcd745
Transfer-Encoding:chunked
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:17:22 GMT
Content-Type:application/json;charset=UTF-8
Connection:close

{
  "errorResponse": {
    "status": "ERROR",
    "code": "SRM-0002",
    "errorMessage": "Missing mandatory parameter: channel name"
  }
}

```

## 6.1.10 User-behavior reporting

### Rationale

This API reports user-behavior information - who has watched what.



## ISO Date-time

This API uses date and time specifiers in the **extended** ISO format. "Extended" means that extra separators are included to improve legibility. The date is in reverse order of size (YYYY-MM-DD) with hyphen separators, eg. 2013-03-20. The time is in 24-hour clock with colon separators (HH:MM:SS). The date and time components are separated by a single uppercase 'T'.

A timezone must be specified. This can either be a single suffix 'Z' (for UTC - recommended) or an offset from UTC. This offset is either a two-digit hour or a 4-digit HHMM (or even HH:mm, although this is unusual).

### Examples:

Expression	Description
2013-03-20T16:30:05Z	The time of writing, in the UK
2013-03-21T00:30:05+0800	The same moment, expressed using Beijing local time

## Request

### HTTP Method

This API uses HTTP GET requests.

### URI

Uses a standard URL with query parameters:

```
http://host:5580/srm/reporting/user-behavior?<parameters>
```

### Parameters

Name	Format	Description	Required?
sessionClient	String	Only return the sessions for given session client (e.g OTV, NAGRA_OTT)	No
contentType	String	Only return sessions of a particular ContentType (e.g COD, NPVR, TS, CU, LIVE)	No - default to return all sessions since epoch, 1970-01-01T00:00:00Z
notStartedBefore	ISO date-time	Only return sessions which started at or after the specified date / time.	No - defaults to the epoch, 1970-01-01T00:00:00Z

Name	Format	Description	Required?
notEndedAfter	ISO date-time	Only return sessions which ended before the specified date / time. If this date is in the future, the response will include sessions which are still open.	No - defaults to a date in the far future
pageOffset	Non-negative integer ( >= 0)	Omit this many results, in result sort order.	No - defaults to 0
maxResults	Non-negative integer (>= 0)	Return at most this many results. 0 means no limit, return all the results.	No - defaults to 100
version	String	API version. Currently only version 1 is supported.	No - defaults to 1

**Additional request headers**

Name	Format	Description	Required?
x-correlation-id	String	Externally-generated request identifier. Used to tie together the sections of a distributed request. If present, will be returned in the response. If missing, will be generated internally and returned in the response.	No

**Access control**

HTTP BASIC authentication is required. By default, the required username is **nagra-agent**.

**Example**

Time-based request

```
GET /srm/reporting/user-behavior?notStartedBefore=2013-05-20T15:30:00Z&pageOffset=50&maxResults=50
Host: srm04.guangzhou.gdc.net.cn
Authorization: Basic bmFncmEtYWdlbnQ6YXJnYW5fJHRkZDI1NDAx
x-correlation-id: 4793w847-guangzhou
```

Content-Type request

```
GET http://cwm-sdp-vdev38:5580/srm/reporting/user-behavior?contentType=COD HTTP/1.1
Accept-Encoding: gzip,deflate
```

**x-correlation-id: 1**

## Success response

On success, the response body is a JSON object. This JSON object is a list of behavior records. Each such record is a JSON map, containing the following fields.

Field name	Format	Description
number	String	A unique identifier for this record.  <b>Note</b> This is the MongoDB internal identifier for the record.
accountId	String	Account ID.
providerId	String	Provider ID. The providerId and assetId form the PAID. VOD only - left blank for CU.
assetId	String	Asset ID. The providerId and assetId form the PAID. VOD only - left blank for CU.
deviceId	String	Device ID.  <b>Note</b> At the time of writing, only smartcard IDs are supported. In future deployments, MAC addresses may be supported as well.
sessionId	String	S1 session ID.
contentType	String	ContentType derived from the Pav (currently supported "COD", "TS", "CU", "NPVR", "LIVE" )
assetUri	String	Incoming request URI
sessionStartTime	ISO date-time	Session start time (same TZ as request).
sessionEndTime	ISO date-time	Session end time (same TZ as request).

Field name	Format	Description						
sessionEndReason	String	Session end reason (user stop / fault / forced disconnect due to VOD lease expiry). Initially blank.						
clientPrivateString	String	Business-layer free string (provided in S1 / E2 Purchase Token). By default, it contains a JSON-formatted map, containing business-layer data. The content depends on whether this is a VOD or CU record. All field values are strings.  <b>VOD:</b>						
<table border="1"> <thead> <tr> <th>Field name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>productId</td> <td>Unique identifier for the product which was purchased.</td> </tr> <tr> <td>catalogueId</td> <td>Unique identifier for the catalogue from which the purchase was made.</td> </tr> </tbody> </table>			Field name	Description	productId	Unique identifier for the product which was purchased.	catalogueId	Unique identifier for the catalogue from which the purchase was made.
Field name	Description							
productId	Unique identifier for the product which was purchased.							
catalogueId	Unique identifier for the catalogue from which the purchase was made.							
<b>CU:</b>								
<table border="1"> <thead> <tr> <th>Field name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>eventId</td> <td>Unique identifier for the event being watched</td> </tr> <tr> <td>productId</td> <td>Unique identifier for the product which was purchased.</td> </tr> </tbody> </table>			Field name	Description	eventId	Unique identifier for the event being watched	productId	Unique identifier for the product which was purchased.
Field name	Description							
eventId	Unique identifier for the event being watched							
productId	Unique identifier for the product which was purchased.							
serverBusinessString	String	Business-layer free string. Contains additional information provided by the PAV server. By default, this is blank.						
srmClientName	String	The name of the session client which setup this session e.g. OTV_VOD, NAGRA_OTT.						
contentID	String	The CMS ID for the content being consumed (this is only applicable for OTT sessions).						

**Caution!** The following fields, which were originally part of the UBR record, have been removed (by agreement):

- ▶ assetName
- ▶ catalogueName
- ▶ eventName
- ▶ channelName

**Sort order**

The list of sessions is sorted by session start time.

**Caution!**

The session start times are stored internally with millisecond precision; this is used to sort as well.

**Example**

Response of a Time-based request

```
200 OK HTTP/1.1
x-correlation-id:f1ad716d-0629-4632-b4ac-16dfd7bc63a1
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:14:32 GMT
Content-Type:application/json;charset=UTF-8
Content-Length:2200

[
  {
    "number":"51caf8772d5d5070445c4fa3",
    "accountId":"Caesar,J",
    "providerId":"ImperialMovies",
    "assetId":"IdesOfMarch",
    "assetUri":"SUI_ASSET4_NOPROF.ts",
    "deviceId":"21f5",
    "sessionId":"1937900175",
    "sessionStartTime":"2013-03-19T20:05:07+0200",
    "sessionEndTime":"2013-03-19T22:05:07+0200",
    "sessionEndReason":"200 User stop",
    "clientPrivateString":{"productId":"LYS-543","catalogueId":"LYS-4322"},
    "serverBusinessString":"",
    "contentID":"","
    "contentType":"COD",
    "srmClientName":"OTV_VOD"
  },
  {
    "number":"51caf8772d5d5070445c4fa4",
    "accountId":"2223334",
    "providerId":"ImperialMovies",
    "assetId":"","
    "assetUri":"","
    "deviceId":"MP_123",
    "sessionId":"1937900189",
```

```

        "sessionStartTime": "2013-03-19T20:06:0+0200",
        "sessionEndTime": "2013-03-19T22:06:30+0200",
        "sessionEndReason": "200 User stop",
        "clientPrivateString": "",
        "serverBusinessString": "",
        "contentID": "LYS_12345678_VOD",
        "contentType": "COD",
        "srmClientName": "NAGRA_OTT"
    },
    {
        "number": "51caf8772d5d5070445c4fa4", "accountId": "Caesar,J", "providerId": "Imperial
Movies", "assetId": "IdesOfMarch", "assetUri": "SUI_ASSET4_NOPROF.ts", "deviceId": "21f4",
        "sessionId": "1503755078", "sessionStartTime": "2013-03-19T20:05:08+0200",
        "sessionEndTime": "2013-03-19T22:05:08+0200", "sessionEndReason": "200 User stop", "business
String": "{ \"assetName\": \"ADI_IDESMARCH\", \"productId\": \"LYS-543\", \"catalogueId\":
\"LYS-4322\" }", "serverBusinessString": "", "contentID": "", "contentType": "COD",
        "srmClientName": "OTV_VOD",
        "number": "51caf8772d5d5070445c4fa5", "accountId": "Caesar,J", "providerId": "Imperial
Movies", "assetId": "IdesOfMarch", "assetUri": "SUI_ASSET4_NOPROF.ts", "deviceId": "21f3",
        "sessionId": "626318356", "sessionStartTime": "2013-03-19T20:05:09+0200",
        "sessionEndTime": "2013-03-19T22:05:09+0200", "sessionEndReason": "200 User stop", "business
String": "{ \"assetName\": \"ADI_IDESMARCH\", \"productId\": \"LYS-543\", \"catalogueId\":
\"LYS-4322\" }", "serverBusinessString": "", "contentID": "", "contentType": "COD",
        "srmClientName": "OTV_VOD",
        "number": "51caf8772d5d5070445c4fac", "accountId": "Caesar,J", "providerId": "Imperial
Movies", "assetId": "IdesOfMarch", "assetUri": "SUI_ASSET4_NOPROF.ts", "deviceId": "21f2",
        "sessionId": "278655455", "sessionStartTime": "2013-03-19T20:05:10+0200",
        "sessionEndTime": "2013-03-19T22:05:10+0200", "sessionEndReason": "200 User stop", "business
String": "{ \"assetName\": \"ADI_IDESMARCH\", \"productId\": \"LYS-543\", \"catalogueId\":
\"LYS-4322\" }", "serverBusinessString": "", "contentID": "", "contentType": "COD",
        "srmClientName": "OTV_VOD",
        "number": "51caf8772d5d5070445c4faf", "accountId": "Caesar,J", "providerId": "Imperial
Movies", "assetId": "IdesOfMarch", "assetUri": "SUI_ASSET4_NOPROF.ts", "deviceId": "21f1",
        "sessionId": "291043737", "sessionStartTime": "2013-03-19T20:05:11+0200",
        "sessionEndTime": "2013-03-19T22:05:11+0200", "sessionEndReason": "200 User stop", "business
String": "{ \"assetName\": \"ADI_IDESMARCH\", \"productId\": \"LYS-543\", \"catalogueId\":
\"LYS-4322\" }", "serverBusinessString": "", "contentID": "", "contentType": "COD",
        "srmClientName": "OTV_VOD"
    }
]

```

Response of a Content-type request

```

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Cache-Control: private
Expires: Thu, 01 Jan 1970 01:00:00 GMT
x-correlation-id: 56dc8bfc-4c57-4ed4-95f0-141ad2fc4c88
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked

```

```
Date: Fri, 14 Nov 2014 12:27:41 GMT
[{"number": "5464bd56e4b0b0e8bf1af715", "accountId": "SUI-N5XS72", "providerId": null, "asset
Id": null, "deviceId": "10001", "sessionId": "2967464414034061751", "contentType": "COD", "asset
Uri": "SUI_ASSET4_NOPROF.ts", "sessionStartTime": null, "sessionEndTime": null, "sessionEnd
Reason": "204 No user activity", "clientPrivateString": "1234", "serverBusinessString": ""}]
```

**Note**

Example based on dummy data. "Pretty printing" applied manually here to make the example easier to read. The real data is not pretty-printed; all the results are returned on a single line.

## Failure response

On failure, the HTTP response code will be set to the most appropriate value (drawn from the set of standard HTTP response codes).

The response body is the JSON encoding of an Exception object which describes the failure.

**Example**

```
HTTP/1.1 400 Bad Request
x-correlation-id:88bc514e-5425-4a23-92db-7b8ddedcd745
Transfer-Encoding:chunked
Server:Apache-Coyote/1.1
Date:Thu, 16 May 2013 13:17:22 GMT
Content-Type:application/json;charset=UTF-8
Connection:close
```

```
{
  "errorResponse": {
    "status": "ERROR",
    "errorCode": "SRM-0001",
    "errorMessage": "Unsupported API version: 2"
  }
}
```

## 6.2 OTT Session Management

This section outlines the inbound HTTP APIs provided by the Nagra SRM for the management of OTT (Over The Top) sessions.

## Revision history

Version	Date	Description
1.0	14 Jan 2015	Initial version

## Connection Management

Although persistent connections are supported in HTTP 1.1, the assumption is that the session manager will not receive all requests for a successfully setup OTT session over a single connection.

### 6.2.1 Session Setup

This API should be used to setup a new OTT session for a client device.

When used, the SRM will verify that the device and associated account are enabled and that they have access to the content requested (via a PAV interface); if so creating a session and returning the details for this.

To use this API, send a HTTP POST request to the below URL:-

`http://<host>:<port>/srm/sessions` (uses the latest version)  
`http://<host>:<port>/srm/v1/sessions` (uses v1)

### HTTP Headers

Name	Type	Description	Required
Client-Token	String	Authentication token gained from a positive sign on response	Yes

### Request body

Name	Type	Description	Required
content	JSON object (Content, see below)	Identifier for the content for which playback is being requested.	Yes

#### Content JSON object

The Content object must contain the below fields.



Name	Type	Description	Required
contentID	String	Identifies the content being consumed <ul style="list-style-type: none"> <li>▶               <ul style="list-style-type: none"> <li>• For VOD this should be a valid CMS Origin Key value identifying a COD Asset</li> <li>• For TimeShift/StartOver, Live Broadcast this should be a valid CMS Origin Key for a BTV Channel</li> <li>• For Catchup and NPVR this should be a valid Broadcast Event CMS Origin Key</li> </ul> </li> </ul>	Yes
contentType	String	Identifies the type of content being requested out of the list of values below:- <ul style="list-style-type: none"> <li>▶               <ul style="list-style-type: none"> <li>• <b>COD</b> - for VOD content</li> <li>• <b>CU</b> - for Catch-up content</li> <li>• <b>TS</b> - for Timeshift / StartOver content</li> <li>• <b>NPVR</b> - for NPVR content</li> <li>• <b>LIVE</b> - for live broadcast content</li> </ul> </li> </ul>	Yes

```
{
  "content": {
    "contentID": "100001V3D",
    "contentType": "COD"
  }
}
```

```
{
  "content": {
    "contentID": "BTV_CH1_V",
    "contentType": "LIVE"
  }
}
```

### Success response

HTTP response code of "200" with the relevant body content will be returned, see <http://engwiki/display/SRM/Error+Codes+and+status+values> page for further details.

Name	Type	Description	Always Present
sessionId	String	A unique identifier returned for the newly created session; this will be in the form of a UUID (Universally unique identifier).	Yes
timeout	Integer	This is the number of seconds in which the session manager expects a keep-alive request to be received for the session.  If this request is not received in the specified grace period, the session will be cleaned up by the session manager.  If the timeout is not specified then the keep-alive feature has been disabled for this session.	No

```
{
  "successResponse": {
    "sessionId": "cfd48f8a-4c33-4c0e-ba52-6632e8810d3e",
    "timeout": 30
  }
}
```

## Error response

Name	Type	Description	Always Present
code	String	The HTTP response code.  For a full list of status values see the <a href="http://engwiki/display/SRM/Error+Codes+and+status+values">http://engwiki/display/SRM/Error+Codes+and+status+values</a> page	Yes
status	String	The categorisation of the failure to process the request.  For a full list of status values see the <a href="http://engwiki/display/SRM/Error+Codes+and+status+values">http://engwiki/display/SRM/Error+Codes+and+status+values</a> page	Yes
errorCode	String	The specific error code.	Yes

Name	Type	Description	Always Present
		For a full list of response codes see the <a href="http://engwiki/display/SRM/Error+Codes+and+status+values">http://engwiki/display/SRM/Error+Codes+and+status+values</a> page	
errorMessage	String	A human readable error message.	Yes

```
{
  "errorResponse": {
    "code": "400",
    "status": "ERROR",
    "errorCode": "SRM-0002",
    "errorMessage": "Malformed request or mandatory parameters missing from request"
  }
}
```

### 6.2.1.1 Session Setup Example Responses

#	As a	Given that	But	Then	SRM Code
1	Client	I have subscribed to content and my session usage is within my account/device/user limits		I receive a session id and timeout value set according to configured value and SRM saves my session information with expiry time set according to configured Keep Alive values.	SRM-0000
2	Client	I have subscribed to content and my session usage is within my account/device/user limits	I supply an expired token	I receive a not Authorised response from HTTP Router	

#	As a	Given that	But	Then	SRM Code
3	Client	My session usage is within my account/device limits	I have not subscribed to the content	I receive a not Authorised response from SRM	SRM-0004
4	Client	I have subscribed to content and my session usage is within my account/device/user limits	Mongo DB service is not running	I receive a failure notification from SRM	SRM-999n
5	Client	I have subscribed to content	My session usage is already at its limit	I receive a not Authorised response from SRM	SRM-0006
6	Client	I have subscribed to content and my session usage is within my account/device/user limits	I supply an invalid token	I receive a not Authorised response from HTTP Router	
7	Client	I have subscribed to content and my session usage is within my account/device/user limits	I have previously requested and received a session id for this content	I receive another session id and timeout value set according to configured value and SRM saves my session information with expiry time set according to configured Keep Alive values.	SRM-0000
8	Client	I have subscribed to content and my session usage is within my account/device/user limits	I have previously requested and received a session id for other content	I receive another session id and timeout value set according to configured value and SRM saves	SRM-0000

#	As a	Given that	But	Then	SRM Code
			and that session is still current	my session information with expiry time set according to configured Keep Alive values.	
10	Client	I have subscribed to content and my session usage is within my account/device/user limits	I have failed to provide content id	I receive an error response from SRM - Bad Request	SRM-0002
11	Client	I have subscribed to content and my session usage is within my account/device/user limits	The PAV is not available	I receive an error response from SRM	SRM-0003
12	Client	I have subscribed to content and my session usage is within my account/device/user limits	The PAV returns an error	I receive an error response from SRM	SRM-999n
13	Client	I have subscribed to content and my session usage is within my account/device/user limits	I have failed to provide Content Type	I receive an error response from SRM - Bad Request	SRM-0002
14	Client	I have subscribed to content and my session usage is within my account/device/user limits	I provided an invalid Content Type	I receive an error response from SRM - Bad Request	SRM-0002

#	As a	Given that	But	Then	SRM Code
15	Client	I have subscribed to content	My session usage is already above its limit	I receive a not Authorised response from SRM and the failure is recorded	SRM-0006
16	Client	I have subscribed to content and my session usage is within my account/device/user limits	SRM is configured to exclude KeepAlive functionality	I receive a session id but no timeout value and SRM saves my session information. The session has no expiry time - the session will be deleted when the start time plus configured duration expires (not part of this test).	SRM-0000
17	Client	I have subscribed to 2 items and my session usage is within my account/device/user limits	The timeout value configured on SRM is updated after I request a session for content id 1 and before I request a session for content id 2	I receive both session ids but the timeout values reflect the configuration current at the time of each request	SRM-0000
18	Client	I have subscribed to content and my session usage is within my account/device/user limits	I request two sessions for the content on 2 different device types which have been configured with different timeout values	I receive both session ids and the timeout values reflect the configuration current for each device type at the time of each request  <b>Not possible at present, we can only use Mp devices</b>	SRM-0000

#	As a	Given that	But	Then	SRM Code
				<b>and a single client type on SRM</b>	
19	Client	I have subscribed to content and my session usage is within my account/device/user limits	I request session discovery at various times	Responses to my session discovery requests are accurate as I add and remove sessions	
20	Client	I have subscribed to content and my session usage is within my account/device/user limits	I request forced session disconnection	My sessions are cancelled and counts of disconnections returned are accurate	
21	Client	I have subscribed to content and my session usage is not within my user limits		My session setup request fails with error ...	SRM-0002
22	Client	I have subscribed to content and my session usage is not within my account/device limits		My session setup request fails with error ...	SRM-0009
23	Client				
24	Client				

## 6.2.2 Session Teardown

This API should be used to close an active OTT session for a client device.

When used, the SRM will mark the session as ended and ready for removal.

To use this API, send a HTTP DELETE request to the below URL:-

`http://<host>:<port>/srm/sessions/{sessionId}` (uses the latest version)

`http://<host>:<port>/srm/v1/sessions/{sessionId}` (uses v1)

## HTTP Headers

Name	Type	Description	Required
Client-Token	String	Authentication token gained from a positive sign on response	Yes

It is assumed that the client will maintain a valid authentication Token to be included in this request.

## Request parameters

Name	Type	Description	Required
sessionId	String	A unique identifier for the session; this will be in the form of a UUID (Universally unique identifier).	Yes

## Success response

HTTP response code of "200" will be returned with no body content, see <https://atlassian.hq.k.grp/confluence/display/SRM/Error+Codes+and+status+values> page for further details.

## Error response

Name	Type	Description	Always Present
code	String	The HTTP response code. For a full list of status values see the <a href="https://atlassian.hq.k.grp/confluence/display/SRM/Error+Codes+and+status+values">https://atlassian.hq.k.grp/confluence/display/SRM/Error+Codes+and+status+values</a> page	Yes
status	string	The categorisation of the failure to process the request.	Yes



Name	Type	Description	Always Present
		For a full list of status values see the <a href="https://atlassian.hq.k.grp/confluence/display/SRM/Error+Codes+and+status+values">https://atlassian.hq.k.grp/confluence/display/SRM/Error+Codes+and+status+values</a> page	
errorCode	string	The specific error code.	Yes
		For a full list of response codes see the <a href="https://atlassian.hq.k.grp/confluence/display/SRM/Error+Codes+and+status+values">https://atlassian.hq.k.grp/confluence/display/SRM/Error+Codes+and+status+values</a> page	
errorMessage	string	A human readable error message.	Yes

```
{
  "errorResponse": {
    "code": "400",
    "status": "ERROR",
    "errorCode": "SRM-0002",
    "errorMessage": "Malformed request or mandatory parameters missing from request"
  }
}
```

## Failure handling

Case	Disposal
Failure to receive response/timeout	<ul style="list-style-type: none"> <li>▶ Client sends teardown request for session to the session manager.</li> <li>▶ Client does not receive a response from session manager               <ul style="list-style-type: none"> <li>• If the session timeout period has not yet been reached the client is free to re-attempt the teardown using a new request within half the configured timeout period.</li> <li>• If the session timeout period has been exceeded the client should assume the session to be closed.</li> <li>• If the session timeout period is not set then the client is free to re-attempt the teardown using a new request N times within T period;</li> </ul> </li> </ul>

Case	Disposal
	after this period the session can be assumed to be closed.
Error response	<ul style="list-style-type: none"> <li>▶ Client sends teardown request for session to the session manager.</li> <li>▶ Client receives an error response from the session manager               <ul style="list-style-type: none"> <li>• The client should assume the session to be closed.</li> </ul> </li> </ul>

### 6.2.2.1 Session Teardown Example Responses

#	As a	Given that	But	Then	SRM Response Code
1	Client	I have a session and I make the call before it has expired		I receive a successful response and my session is removed on SRM	
2	Client	I have a session and I make the call after it has expired but within my grace period		I receive a successful response and my session is removed on SRM	
3	Client	I have a session and I make the call after it has expired but within my grace period	KeepAlive is not configured	I receive a successful response even though my session has already been removed by SRM	
4	Client	I have a session and I make the call before it has expired	I fail to include the session id in my call	I receive an error response and my session	SRM-0002

#	As a	Given that	But	Then	SRM Response Code
				remains on SRM	
5	Client	I have a session and I make the call before it has expired	I include an unknown session id in my call	I receive a successful response	
6	Client	I have a session and I make the call before it has expired	I include the id of a session which belongs to my account but not to my current device	I receive a success response and the session remains on SRM	
7	Client	I have a session and I make the call before it has expired	I include the id of a session which belongs to another account/device	I receive a success response and the session remains on SRM	
8	Client	I have a session and I make the call before it has expired	My token has expired	I receive an error response from HTTP-Router	
9	Client	I have a session and I make the call before it has expired	I repeat the call	I receive a successful response even though my session has already been removed by SRM	
10	Client	I have a session and I make the call after it has expired and after my grace period		I receive a successful response even though my session has already been removed by SRM	

## 6.2.3 Session Keepalive

This API should be used to maintain an active OTT session for a client device.

When used, the SRM will check the session state, if still valid it will identify the session as being active returning a positive response to the client.

Where a client does not request a keepalive for a session within a configured time period, that session will become a candidate for removal, with further client access to that session resulting in an error response.

The time period within which a keepalive is expected by the session manager for the session, is included in the "timeout" field of a successful SETUP response message.

If no "timeout" value has been specified the client should assume that keepalive requests are not required for this session and should not be sent.

To use this API, send a HTTP PUT request to the below URL:-

`http://<host>:<port>/srm/sessions/{sessionId}`  
`http://<host>:<port>/srm/v1/sessions/{sessionId}` (uses v1)

### HTTP Headers

Name	Type	Description	Required
Client-Token	String	Authentication token gained from a positive sign on response	Yes

It is assumed that the client will maintain an valid authentication Token to be included in this request.

### Request parameters

Name	Type	Description	Required
sessionId	String	A unique identifier for the session; this will be in the form of a UUID (Universally unique identifier).	Yes

Success response

HTTP response code of "200" will be returned, see <http://engwiki/display/SRM/Error+Codes+and+status+values> page for details.

## Error response

Name	Type	Description	Always Present
code	String	The HTTP response code. For a full list of status values see the <a href="http://engwiki/display/SRM/Error+Codes+and+status+values">http://engwiki/display/SRM/Error+Codes+and+status+values</a> page	Yes
status	string	The categorisation of the failure to process the request. For a full list of status values see the <a href="http://engwiki/display/SRM/Error+Codes+and+status+values">http://engwiki/display/SRM/Error+Codes+and+status+values</a> page	Yes
errorCode	string	The specific error code. For a full list of response codes see the <a href="http://engwiki/display/SRM/Error+Codes+and+status+values">http://engwiki/display/SRM/Error+Codes+and+status+values</a> page	Yes
errorMessage	string	A human readable error message.	Yes

```
{
  "errorResponse": {
    "code": "400",
    "status": "ERROR",
    "errorCode": "SRM-0002",
    "errorMessage": "Malformed request or mandatory parameters missing from request"
  }
}
```

```
{
  "errorResponse": {
    "code": "404",
    "status": "ERROR",
    "errorCode": "SRM-0008",
    "errorMessage": "Session Not Found"
  }
}
```

## Failure handling

Case	Disposal
Failure to receive response/timeout	<ul style="list-style-type: none"> <li>▶ Client sends keepalive request for session to the session manager.</li> <li>▶ Client does not receive a response from session manager.                             <ul style="list-style-type: none"> <li>• If the session timeout period has not yet been reached the client is free to re-attempt the session keepalive using a new request within half the configured timeout period.</li> <li>• If the session timeout period has been exceeded the client should send a TEARDOWN request for the session and attempt to setup a new session if required (i. e. OTT stream is still active) .</li> </ul> </li> </ul>
Error response	<ul style="list-style-type: none"> <li>▶ Client sends a keepalive request for the session to the session manager</li> <li>▶ Client receives an error response from the session manager                             <ul style="list-style-type: none"> <li>• Client should send a TEARDOWN request for the session and attempt to setup a new session if required (i.e. OTT stream is still active).</li> </ul> </li> </ul>

### 6.2.3.1 Session Keepalive Example Responses

#	As a	Given that	But	Then	SRM Response Code
1	Client	I have a session and I make the call before it has expired		I receive a successful response and the expiry date of my session is updated on SRM (need KeepaliveBefore Millis) using configured Keep Alive values	SRM-0000

#	As a	Given that	But	Then	SRM Response Code
2	Client	I have a session and I make the call before it has expired	My token has expired	I receive an error response from HTTP-Router	
3	Client	I have a session and I make the call before it has expired	KeepAlive is not implemented in SRM	I receive a successful response and the expiry date of my session is not updated on SRM	SRM-0000
4	Client	I have a session and I make the call before it has expired	the session id I supply is incorrect, it does not exist	I receive an error response from SRM	SRM-0008
5	Client	I have a session and I make the call before it has expired	the session id I supply is incorrect, it belongs to another account/device	I receive an error response from SRM and the referenced session is not updated on SRM	SRM-0008
6	Client	I have a session and I make the call before it has expired	I do not provide a session id	I receive an error response from SRM	SRM-002
7	Client	I have a session and I make the call before it has expired	the session id I supply is incorrect, it belongs to my account but not my device	I receive an error response from SRM and the referenced session is not updated on SRM	SRM-0008
8	Client	I have a session and KeepAlive Misses has not been configured (and defaults to zero)	I make the call after the timeout has expired but before the grace period has expired - KeepAlive	I receive a successful response and the expiry date of my session is updated on SRM using	SRM-0000

#	As a	Given that	But	Then	SRM Response Code
9	Client	I have a session and KeepAlive Misses has not been configured (and defaults to zero)	Misses has not been configured (and defaults to zero) I make the call after the timeout has expired and after the grace period has expired	configured Keep Alive values I receive an error response from SRM and the referenced session has been removed from SRM	SRM-0008

## 6.3 RTSP Gateway

This section details the various RTSP methods and dialects that are supported by the SRM.

### 6.3.1 SETUP

#### Overview

RTSP SETUP requests are sent from the Client (STB or a third-party SRM) to the SRM to reserve network bandwidth. It is the first of the RTSP methods to use. The client must issue this SETUP request and the server must issue a SETUP response before the client or the server can utilize other methods.

#### Note

The SRM is flexible and can support a multitude of different clients which can be configured via the SRM Ingest.

Within the SRM each client is converted into a dialect - and this dialect determines the workflow to be used (for example whether the client requires Product Validation, an external resource manager etc..)

#### Request Format

To establish a session, the SRM shall send an RTSP SETUP request to the SRM. This is a regular RTSP request, with no body. In (quasi-)BNF the format of this request shall be

```
SETUP SP <uri> SP RTSP/1.0 CRLF
1*<header> CRLF
<body>
```



**Request URI**

The request <uri> shall have the form:

```
rtsp://<logicalSrmAddress>:<logicalSrmPort>/<pathString> RTSP/1.0
```

Field	Description								
<logicalSrmAddress>	The DNS name or IP address which the Client uses to contact the SRM.								
<logicalSrmPort>	The TCP port which the Client sends RTSP request to on the SRM.								
<pathString>	Used in the case of VOD and SDV to identify the content/channel being consumed. Only required for OpenTV								
	<table border="1"> <thead> <tr> <th>Content type</th> <th>&lt;pathString&gt; format</th> </tr> </thead> <tbody> <tr> <td>SDV</td> <td>?SDV=&lt;channel_number&gt; &lt;channel_number&gt; is a channel key, which uniquely identifies the channel.</td> </tr> <tr> <td></td> <td> <p><b>Caution!</b></p> <p>The <a href="http://www.ietf.org/rfc/rfc2326.txt">http://www.ietf.org/rfc/rfc2326.txt</a> mandates use of UTF-8 to encode requests, and therefore we do as well. However, we recommend that channel number be restricted to US-ASCII, in order to avoid any potential compatibility issues / internationalization bugs in the software chain.</p> </td> </tr> <tr> <td>VOD</td> <td>&lt;content_name&gt;&lt;request_params&gt;</td> </tr> </tbody> </table>	Content type	<pathString> format	SDV	?SDV=<channel_number> <channel_number> is a channel key, which uniquely identifies the channel.		<p><b>Caution!</b></p> <p>The <a href="http://www.ietf.org/rfc/rfc2326.txt">http://www.ietf.org/rfc/rfc2326.txt</a> mandates use of UTF-8 to encode requests, and therefore we do as well. However, we recommend that channel number be restricted to US-ASCII, in order to avoid any potential compatibility issues / internationalization bugs in the software chain.</p>	VOD	<content_name><request_params>
Content type	<pathString> format								
SDV	?SDV=<channel_number> <channel_number> is a channel key, which uniquely identifies the channel.								
	<p><b>Caution!</b></p> <p>The <a href="http://www.ietf.org/rfc/rfc2326.txt">http://www.ietf.org/rfc/rfc2326.txt</a> mandates use of UTF-8 to encode requests, and therefore we do as well. However, we recommend that channel number be restricted to US-ASCII, in order to avoid any potential compatibility issues / internationalization bugs in the software chain.</p>								
VOD	<content_name><request_params>								

Field	Description									
<b>Content type</b>	<p><b>&lt;pathString&gt; format</b></p> <p>&lt;content_name&gt; is the asset uri as required for product validation, and the video-server contributor.</p> <p>&lt;request_params&gt; is a list of HTTP request params, preceded with a ? and separated via an ampersand. Valid params follow:</p> <table border="1"> <thead> <tr> <th>Req Param Name</th> <th>Description</th> <th>Required</th> </tr> </thead> <tbody> <tr> <td>session</td> <td>Session ID when communicating for a pre-existing session</td> <td>No</td> </tr> <tr> <td>bitrate</td> <td>The bandwidth to reserve in bytes. This value will only be utilised for deployments with a no-op PAV configuration.</td> <td>No</td> </tr> </tbody> </table>	Req Param Name	Description	Required	session	Session ID when communicating for a pre-existing session	No	bitrate	The bandwidth to reserve in bytes. This value will only be utilised for deployments with a no-op PAV configuration.	No
Req Param Name	Description	Required								
session	Session ID when communicating for a pre-existing session	No								
bitrate	The bandwidth to reserve in bytes. This value will only be utilised for deployments with a no-op PAV configuration.	No								

**Request Headers**

Header	Format	Definition	Mandatory?
CSeq	Integer	Sequence number, which is incremented by one for each distinct request transmitted. It is a 64-bit decimal integer	Yes

Header	Format	Definition	Mandatory?
User-Agent	String	This identifies the type of client which sent the request;.	Yes
Require	String	The Require header is used by clients to query the server about options that it may or may not support. The server MUST respond to this header by using the Unsupported header to negatively acknowledge those options which are NOT supported.	No
		<p><b>Note</b></p> <p>The RTSP standard [RFC 2326] can be extended by using additional headers objects. For that purpose a Require tag is introduced to handle special functionality additions (see [RFC 2326], 1.5 Extending Rtsp and 12.32 Require).</p>	
Transport	Multipart	clab-DOCSIS/QAM ;unicast	Yes for 3rd Party

Header	Format	Definition	Mandatory?										
		<pre> ;bit_rate=&lt;bit_rate&gt; ;depi_mode=&lt;depi_mode&gt; ;qam_name=&lt;qam_name&gt; ;source=&lt;source&gt; </pre>											
		<table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>unicast</td> <td>Means that the streaming will be made in unicast. Multicast mode is not used</td> </tr> <tr> <td>bit_rate</td> <td>Integer used to allocate resource in the Cable Network. It's the content TS bitrate in bps.</td> </tr> <tr> <td>depi_mode</td> <td>String Used for ERMI-3 compatibility, its value is always "docsis_mpt"</td> </tr> <tr> <td>qam_name</td> <td>String Used by the Cable Network to identify</td> </tr> </tbody> </table>	Parameter	Description	unicast	Means that the streaming will be made in unicast. Multicast mode is not used	bit_rate	Integer used to allocate resource in the Cable Network. It's the content TS bitrate in bps.	depi_mode	String Used for ERMI-3 compatibility, its value is always "docsis_mpt"	qam_name	String Used by the Cable Network to identify	
Parameter	Description												
unicast	Means that the streaming will be made in unicast. Multicast mode is not used												
bit_rate	Integer used to allocate resource in the Cable Network. It's the content TS bitrate in bps.												
depi_mode	String Used for ERMI-3 compatibility, its value is always "docsis_mpt"												
qam_name	String Used by the Cable Network to identify												

Header	Format	Definition	Mandatory?
		<p><b>Parameter Description</b></p> <p>an EQAM. It contains the NIT and the TSID separate by a dot, i.e. NIT.TSID</p> <p><b>Syntax example:</b> qam_name=64753.20101</p> <p><b>NIT</b> The Digit ID. E.g. 64753</p> <p><b>TSID</b> Transport ID. or 5 E. Digits 20101.</p> <p><b>sourceIP</b> This is <b>optional</b> and corresponds to the IP of the streaming server. It can be used by</p>	

Header	Format	Definition	Mandatory?
		<p><b>Parameter Description</b></p> <p>the edge qam to verify the source of the incoming stream</p>	
Transport	Multipart	<p>MP2T/DVBC/QAM;unicast;client=&lt;client Id&gt;</p> <p><b>Parameter Description</b></p> <p>client &lt;Smartcard information Id&gt; - Network Id - SRM Media Group - Channel Id - smartcard and network identifiers. It gives the device via use of the smartcard Network ID. E.g. 64753</p>	Yes for OTV

Header	Format	Definition	Mandatory?
		<p><b>Service Group</b> service-group for this network.</p> <p><b>QAM Port</b> QAM Port which the STB expects to receive the Transport stream (this maps to the tuning frequency)</p>	
clab-ClientSessionId	String	<p>A unique session identifier made up of Client-ID + Session-ID</p> <p>Example for an RTSP Client with a MAC address of 00:AF:12:34:56:DE and a Session-ID of 00000001:</p>	Yes for cablelabs

Header	Format	Definition	Mandatory?
		clab-ClientSessionId: 00AF123456DE00000001	
clab-SessionGroup	String	This indicates to which group the session will be associated. The parameter will be sent into GET_PARAMETER request in order to fetch only the session belonging to our SessionGroup  Default value is "SMARTVISION_CRM_GROUP" for Thomson/Bytel	Yes for cablelabs
sessionId	String	This indicates a pre-existing session for which the SRM is completing the session. Typically used when the original session was established via the HTTP Router	No

**Warning!**

Headers which are not specified as mandatory may still be required in order to identify a unique client. For example third-party clients require the clab related headers.

**Request Body**

An optional body can be specified. However no inbound validation is performed on the format of the body.

Where a body is specified it will be proxied onto any outbound Video-Server request, should the outbound dialect support a body

**Response Format**

In (quasi-)BNF the format of the response to this request shall be:

```
RTSP/1.0 <responseCode> SP <reasonPhrase> CRLF
1*<header> CRLF
```

**Response Headers**



If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format	Definition	Mandatory?
CSeq	Integer	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	Client Specific	<p>The ID of the RTSP session. This is standard and part of the RTSP protocol</p> <p><code>&lt;session Id&gt;[;timeout=&lt;delta&gt;]</code></p> <p>In the nominal case of an OTV <b>Second Configuration</b> deployment this contains session information for the VOD Server.</p> <p>In an OTV <b>First Configuration</b> deployment this contains the session information for the Management server.</p>	Yes

**Field Description Mandatory**

Field	Description	Mandatory
<code>&lt;session Id&gt;</code>	alphanumeric session ID. Max 20 characters (for compatibility with Open TV). The format rules of the ID are dependent	Yes

Header	Format	Definition	Mandatory?
		<p>on the client.</p> <p><b>Note</b></p> <p>Both SRM and Open TV support alphanumeric session IDs. However, all the examples in the <a href="http://www.ietf.org/rfc/rfc2326.txt">http://www.ietf.org/rfc/rfc2326.txt</a> are purely numeric. Some STBs may not expect to see letters in a session ID.</p> <p>&lt;delta&gt; the (optional)</p>	<p><b>Field Description Mandatory</b></p>

Header	Format	Definition	Mandatory?
		<p><b>Field Description Mandatory</b></p> <p>timeout clause is provided, the maximum interval, in seconds, which the client can leave between keepalive requests to the SRM for this session. If this limit is exceeded, the session shall be terminated. The client should typically send at an interval which is between 50% and 80% of this value.</p> <p><b>Note</b></p> <p>The timeout interval for the STB-to-</p>	

Header	Format	Definition	Mandatory?
		<p style="text-align: center;"><b>Field Description Mandatory</b></p> <hr/> <p style="text-align: center;">SRM client is defined by the SRM, not by this value.</p> <hr/>	
Transport	Multipart	<pre> clab-DOCSIS/QAM ;unicast ;bit_rate=&lt;bit_rate&gt; ;depi_mode=&lt;depi_mode&gt; ;qam_name=&lt;qam_name&gt; ;source=&lt;source&gt; ;qam_destination=&lt;qam_destination&gt; ;modulation=&lt;modulation&gt; ;symbol_rate=&lt;symbol_rate&gt; ;destination=&lt;destination&gt; ;destination_port=&lt;destination_port&gt; </pre>	No
		<p style="text-align: center;"><b>Parameter Description</b></p> <hr/> <p style="text-align: center;">unicast</p>	

Header	Format	Definition	Mandatory?
		<p><b>Param</b></p> <p><b>Form</b></p> <p><b>Description</b></p>	
		<p>Means that the streaming will be made in unicast. Multicast mode is not used</p>	
		<p>bit_rate Integer Used to allocate resource in the Cable Network. It's the content TS bitrate in bps.</p>	
		<p>depi_mode String Used for ERMI-3 compatibility, its value is always "docsis_mpt"</p>	
		<p>qam_name String Used by the Cable Network to identify an EQAM. It contains the NIT and the TSID separate by a dot.</p>	

Header	Format	Definition	Mandatory?
<b>Parameter</b>	<b>Format</b>	<b>Description</b>	
		<p><b>Syntax example:</b> qam_ name= 64753.20101</p>	
		<p>1. The 2. Digits 3. ID.  4. E. 5. 9. 6. 64753</p>	
		<p>1. Transport 2. ID. 3. or 4. E. 5. Digits 6. 2010 7. or 8. 20101.</p>	
		<p>sourceString This is <b>optional</b> and corresponds to the IP of the streaming server. It can be used by the edge qam to verify the source of the incoming stream</p>	

Header	Format	Definition	Mandatory?
--------	--------	------------	------------

**Parameter Description**

qam\_ destination The QAM Channel information. It is composed of the Frequency and the Program Number separated by a "."

modulation The modulation type of the QAM channel. It can take values from 3 to 7. Each value description is in the table below.

**Modulation Description**

3 64-QAM

4 256-QAM

5 128-QAM

Header	Format	Definition	Mandatory?						
		<p><b>Parameters</b></p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>6</td> <td>512-QAM</td> </tr> <tr> <td>7</td> <td>1024-QAM</td> </tr> </tbody> </table> <p><b>symbol_rate</b> The symbol rate of the channel in bps</p> <p><b>destination_ip</b> The IP of the Edge QAM channel where the resource has been reserved.</p> <p><b>destination_port</b> port number in range 0-65535</p>	Parameter	Description	6	512-QAM	7	1024-QAM	
Parameter	Description								
6	512-QAM								
7	1024-QAM								
Location		rtsp://<phys_addr>:<phys_port>	Yes for OTV						



Header	Format	Definition	Mandatory?
		<b>Field Description Mandatory?</b>	
		<p>&lt;physThe DNS addr&gt;name or IP address which the STB should use to contact the VS. In systems where the physical VS and logical- VS / orchestrator are different, this will usually be the address of a physical VS.</p>	Yes
		<p>&lt;physThe TCP port&gt; port on which the STB should contact the VS.</p>	Yes

**Note**

As the SRM serves traffic for both VOD and Management servers

Header	Format	Definition	Mandatory?
		for SDV sessions, the location details will be the address and port of an SRM server.	
ControlSession		<p>&lt;session id&gt;;timeout=&lt;timeout&gt;</p> <p>In an OTV <b>First Configuration</b> deployment this contains session information for the VOD Server.</p> <p>Same format as session</p>	No
Tuning		<p>frequency=&lt;frequency&gt;;symbol_rate=&lt;symbol_rate&gt;;modulation=&lt;modulation&gt;;fec_inner=&lt;fec_inner&gt;;fec_outer=&lt;fec_outer&gt;</p>	Yes
<b>Field Description Mandatory</b>			
		<p>frequency denotes the signal frequency for tuning to the QAM channel; it is an 8 digit value in 4-bit BCD format.</p>	Yes

Header	Format	Definition	Mandatory?
<b>Field Description Mandatory</b>			
symbols_ rate		<p>is the symbol rate for the QAM channel in symbols per second; it is a 7 digit value in 4-bit BCD.</p>	Yes
modulation		<p>represents the QAM Modulation scheme used to convey the data. Possible values are 1, 2, 3, 4 or 5.</p>	Yes
fec_inner		<p>Forward Error Correction (FEC) which provides error conversion rates for bad data in a transport stream. Its value ranges from 1 - 15</p>	Yes

Header	Format	Definition	Mandatory?
		<b>Field Description Mandatory</b>	
		fec_ It outer provides more precise error conversion than using fec_ inner. Its values are either 1 or 2.	Yes
Channel		Tsid=<Tsid>; Svcid =<Svcid>	Yes for OTV
		<b>Field Description Mandatory</b>	
		Tsid Uniquely identifies the MPEG Transport Stream Identifier (TSID) being outputted on the QAM Channel allocated.	Yes
		<b>Format</b>	
		4 digits	

Header	Format	Definition	Mandatory?
		<p style="text-align: center;"><b>Field Description Mandatory</b></p> <hr/> <p style="text-align: center;"><b>Format</b></p> <hr/> <p style="text-align: center;">1 - 9999</p> <p style="text-align: center;">5 1 digits 65535</p> <hr/> <p>Svcid The DVB Service ID/ Program Number on the Transport Stream, it identifies the individual program on a multiplexed stream which a client has access to. It's value ranges form 1 - 31.</p> <hr/>	Yes
clab-ClientSessionId	String	A unique session identifier made up of Client-ID + Session-ID	No

Header	Format	Definition	Mandatory?
		<p>Example for an RTSP Client with a MAC address of 00:AF:12:34:56:DE and a Session-ID of 00000001:</p> <p><a href="http://clab-clientsessionid:00AF123456DE00000001">http://clab-clientsessionid:00AF123456DE00000001</a></p>	

**Response Codes**

Code / phrase	Reason
200 OK	The request succeeded
400 Bad request	The request was incorrectly formatted.
401 Unauthorized	
402 Payment Required	
404 Not found	The requested session does not exist.
405 Method Not Allowed	
408 Request Time-out	
415 Unsupported Media Type	
451 Parameter not understood	A header or URI field was invalid. (Implementations may return 400 Bad request instead.)
453 Not Enough Bandwidth	There is not enough bandwidth to service the request.
454 Session Not Found	
461 Unsupported Transport	
462 Destination Unreachable	

Code / phrase	Reason
500 Internal error	There is an unspecified fault within the SRM.
503 Service Unavailable	

### 6.3.1.1 RTSP Setup Examples

```
SETUP rtsp://test-apps4:8184/?SDV=1 RTSP/1.0
User-Agent: OpenTV VOD 1
Transport: MP2T/DVBC/QAM;unicast;client=170148375.6475320101
CSeq: 90512
```

```
RTSP/1.0
200 OK
CSeq: 90512
Session: NVS821130592
Location: rtsp://172.16.9.4:8184
Tuning: frequency=04000000;symbol_rate=0068750;modulation=3;fec_inner=15
Channel: Tsid=20101;Svcid=68
```

```
SETUP rtsp://test-apps4:8184/MyVideo.ts RTSP/1.0
User-Agent: NC-VOD
Transport: MP2T/DVBC/QAM;unicast;client=64753.201.170148375
CSeq: 90512
```

```
SETUP rtsp://test-apps4:8184/MyVideo.ts?bitrate=3500000 RTSP/1.0
User-Agent: NC-VOD
Transport: MP2T/DVBC/QAM;unicast;client=64753.201.170148375
CSeq: 90512
Content-Length: 114
```

```
playlist_item=WebUI-1502723389760;ordinal=0;trickmode=0
playlist_item=WebUI-1502723359578;ordinal=1;trickmode=0
```

```
SETUP rtsp://test-apps4:8184/MyVideo.ts?session=123456 RTSP/1.0
User-Agent: OpenTV VOD 1
Transport: MP2T/DVBC/QAM;unicast;client=170148375.6475320101
CSeq: 90512
```

```
RTSP/1.0
200 OK
CSeq: 90512
Session: 821130592
Location: rtsp://172.16.9.4:8184
Tuning: frequency=04000000;symbol_rate=0068750;modulation=3;fec_inner=15
Channel: Tsid=20101;Svcid=68
```

```
SETUP rtsp://test-apps4:8184 RTSP/1.0
User-Agent: Playcast
clab-SessionGroup: PLAYCAST_CRM_GROUP
Transport: clab-DOCSIS/QAM;unicast;bit_rate=1250000;depi_mode=docsis_mpt;qam_
name=64753.20101;source=127.7.0.2
clab-ClientSessionId: 2e226d142ae4be12
CSeq: 90514
```

```
RTSP/1.0
200 OK
CSeq: 90514
Session: 821130593;timeout:900
Transport: clab-DOCSIS/QAM;unicast;bit_rate=1250000;depi_mode=docsis_mpt;qam_
name=64753.20101;source=127.7.0.2;qam_destination=400000000.69;modulation=3;symbol_
rate=6875000;destination=172.16.5.80;destination_port=44100
clab-ClientSessionId: 2e226d142ae4be12
```

```
SETUP rtsp://test-apps4:8184 RTSP/1.0
User-Agent: Thomson
clab-SessionGroup: SMARTVISION_CRM_GROUP
Transport: clab-DOCSIS/QAM;unicast;bit_rate=1250000;depi_mode=docsis_mpt;qam_
name=64753.20101;source=127.7.0.2
clab-ClientSessionId: 59f94da91e23607c
CSeq: 90516
```

```
RTSP/1.0
200 OK
CSeq: 90516
Session: 821130594;timeout:360
Transport: clab-DOCSIS/QAM;unicast;bit_rate=1250000;depi_mode=docsis_mpt;qam_
name=64753.20101;source=127.7.0.2;qam_destination=400000000.70;modulation=3;symbol_
rate=6875000;destination=172.16.5.80;destination_port=44200
clab-ClientSessionId: 59f94da91e23607c
```

```
SETUP rtsp://test-apps4:8184/video-01.mpg RTSP/1.0
User-Agent: Nagra VOD
```



Transport: MP2T/DVBC/QAM;unicast;client=170148375.6475320101  
CSeq: 90512

RTSP/1.0  
200 OK  
CSeq: 90512  
Session: 821130592  
Location: rtsp://172.16.9.4:8184  
Tuning: frequency=04000000;symbol\_rate=0068750;modulation=3;fec\_inner=15  
Channel: Tsid=20101;Svcid=68

SETUP rtsp://test-apps4:8184/video-01.mpg RTSP/1.0  
User-Agent: Nagra UGC  
Transport: MP2T/DVBC/QAM;unicast;client=170148375.6475320101  
CSeq: 90512

RTSP/1.0  
200 OK  
CSeq: 90512  
Session: 821130592  
Location: rtsp://172.16.9.4:8184  
Tuning: frequency=04000000;symbol\_rate=0068750;modulation=3;fec\_inner=15  
Channel: Tsid=20101;Svcid=68

SETUP rtsp://srm:554/;purchaseToken=53c7eeeb-5c30-456f-a79c-4893c27cd226;serverID=1.1.1.1  
RTSP/1.0  
CSEQ: 123  
Require: com.comcast.ngos.s1  
Transport: MP2T/DVBC/QAM;unicast;client=00AF123456DE;qam\_name=Chicago.Southbend.5;  
ClientSessionId: 00AF123456DE00000001

RTSP/1.0 200 OK  
CSEQ: 123  
Session 716195834  
Transport: MP2T/DVBC/QAM;unicast;destination=24.000000.23  
OnDemandSessionId: be074250cc5a11d98cd50800200c9a66  
Content-type: application/sdp  
Content-length: 149  
v=0000000o=- 777 2890842817 IN IP4 1.2.3.4  
s=  
t=0 0  
a=control:rtsp://videosever:554/9876  
c=IN IP4 0.0.0.000000m= video 0 udp MP2T

## 6.3.2 TEARDOWN

### Overview

TEARDOWN RTSP requests are sent from the Client to the SRM to end a session. It is the last method called in the RTSP lifecycle. The client must have issued a SETUP request and the server must issue a SETUP response before the client can utilize this method.

### Request Format

To terminate a session and release the resource, the client shall send an RTSP TEARDOWN request to the SRM. This is a regular RTSP request, with no body. In (quasi-)BNF the format of this request shall be

```
TEARDOWN SP <uri> SP RTSP/1.0 CRLF
1*<header> CRLF
```

### Request URI

The TEARDOWN request uses an asterisk \* in place of a particular URI to match the current presentation URI . Alternatively the request <uri> shall have the form:

```
rtsp://<logicalSrmAddress>:<logicalSrmPort>/ RTSP/1.0
```

### Request Headers

Header	Format	Definition	Mandatory?
CSeq		Sequence number, which is incremented by one for each distinct request transmitted. It is a 64-bit decimal integer	Yes
Require		The Require header is used by clients to query the server about options that it may or may not support. The server MUST respond to this header by using the Unsupported header	Yes for Cablelabs

Header	Format	Definition	Mandatory?
--------	--------	------------	------------

to negatively acknowledge those options which are NOT supported.

**Note**

The RTSP standard [RFC 2326] can be extended by using additional headers objects. For that purpose a Require tag is introduced to handle special functionality additions (see [RFC 2326], 1.5 Extending Rtp and 12.32 Require).

clab-Reason

String

The reason why a TEARDOWN is being requested. One of:

Yes for Cablelabs

Code	Description
------	-------------

200	User stopped
-----	--------------

204	No user activity
-----	------------------

205	STB capability mismatch
-----	-------------------------

Header	Format	Definition	Mandatory?																												
		<table border="1"> <thead> <tr> <th>Code</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>206</td> <td>Insufficient priority</td> </tr> <tr> <td>207</td> <td>Network delivery failure</td> </tr> <tr> <td>400</td> <td>Fail to tune</td> </tr> <tr> <td>401</td> <td>Loss of tune</td> </tr> <tr> <td>402</td> <td>Loss of tune</td> </tr> <tr> <td>403</td> <td>RTSP failure</td> </tr> <tr> <td>404</td> <td>Channel failure</td> </tr> <tr> <td>405</td> <td>No RTSP server</td> </tr> <tr> <td>408</td> <td>Unknown</td> </tr> <tr> <td>409</td> <td>Network resource failure</td> </tr> <tr> <td>420</td> <td>STB heartbeat timeout</td> </tr> <tr> <td>421</td> <td>STB inactivity timeout</td> </tr> <tr> <td>422</td> <td>Content unavailable</td> </tr> </tbody> </table>	Code	Description	206	Insufficient priority	207	Network delivery failure	400	Fail to tune	401	Loss of tune	402	Loss of tune	403	RTSP failure	404	Channel failure	405	No RTSP server	408	Unknown	409	Network resource failure	420	STB heartbeat timeout	421	STB inactivity timeout	422	Content unavailable	
Code	Description																														
206	Insufficient priority																														
207	Network delivery failure																														
400	Fail to tune																														
401	Loss of tune																														
402	Loss of tune																														
403	RTSP failure																														
404	Channel failure																														
405	No RTSP server																														
408	Unknown																														
409	Network resource failure																														
420	STB heartbeat timeout																														
421	STB inactivity timeout																														
422	Content unavailable																														

Header	Format	Definition	Mandatory?																		
		<table border="1"> <thead> <tr> <th>Code</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>423</td> <td>Streaming failure</td> </tr> <tr> <td>424</td> <td>QAM failure</td> </tr> <tr> <td>425</td> <td>Volume failure</td> </tr> <tr> <td>426</td> <td>Stream control error</td> </tr> <tr> <td>427</td> <td>Stream control timeout</td> </tr> <tr> <td>428</td> <td>Session list mismatch</td> </tr> <tr> <td>502</td> <td>QAM parameter mismatch</td> </tr> <tr> <td>550</td> <td>Session timeout</td> </tr> </tbody> </table> <p><b>The SmartVision CDN will use clab-Reason code 200, 206, 403, 409 or 550</b></p>	Code	Description	423	Streaming failure	424	QAM failure	425	Volume failure	426	Stream control error	427	Stream control timeout	428	Session list mismatch	502	QAM parameter mismatch	550	Session timeout	
Code	Description																				
423	Streaming failure																				
424	QAM failure																				
425	Volume failure																				
426	Stream control error																				
427	Stream control timeout																				
428	Session list mismatch																				
502	QAM parameter mismatch																				
550	Session timeout																				
Session	Integer	<p>The ID of the RTSP session, as returned from the SETUP request. This is standard and part of the RTSP protocol</p> <p>For OpenTV this is an alphanumeric session ID, Max 20 characters</p>	Yes																		

Header	Format	Definition	Mandatory?
		which will be prefixed with "NVS". in the case of SDV	
clab-ClientSessionId	String	A unique session identifier made up of Client-ID + Session-ID  Example for an RTSP Client with a MAC address of 00:AF:12:34:56:DE and a Session-ID of 00000001:  <a href="http://clab-clientsessionid:00AF123456DE00000001">http://clab-clientsessionid:00AF123456DE00000001</a>	Yes for Cablelabs

**Response Format**

In (quasi-)BNF the format of the response to this request shall be:

```
RTSP/1.0 <responseCode> SP <reasonPhrase> CRLF
1*<header> CRLF
```

**Response Headers**

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	The ID of the RTSP session. This is standard and part of the RTSP protocol	Yes for Comcast
clab-ClientSessionId	A unique session identifier made up of Client-ID + Session-ID  Example for an RTSP Client with a MAC address of 00:AF:12:34:56:DE and a Session-ID of 00000001:  <a href="http://clab-clientsessionid:00AF123456DE00000001">http://clab-clientsessionid:00AF123456DE00000001</a>	Yes for Comcast

**Response Codes**

Code / phrase	Reason
200 OK	The request succeeded
400 Bad request	The request was incorrectly formatted.
401 Unauthorized	
402 Payment Required	
404 Not found	The requested session does not exist.
405 Method Not Allowed	
408 Request Time-out	
415 Unsupported Media Type	
451 Parameter not understood	A header or URI field was invalid. (Implementations may return 400 Bad request instead.)
453 Not Enough Bandwidth	There is not enough bandwidth to service the request.
454 Session Not Found	
461 Unsupported Transport	
462 Destination Unreachable	
500 Internal error	There is an unspecified fault within the SRM.
503 Service Unavailable	

**6.3.2.1 RTSP TEARDOWN Examples**

```
TEARDOWN * RTSP/1.0
CSeq: 5
```

```
Session: NVS118582287615149
```

```
RTSP/1.0 200 OK  
CSeq: 5
```

```
TEARDOWN rtsp://192.0.2.2/ RTSP/1.0  
CSeq: 315  
Require: com.cablelabs.ermi  
clab-Reason: 200 "User stop"  
Session: 47112345  
clab-clientSessionId: 8cd50800200c00000000
```

```
RTSP/1.0 200 OK  
CSeq: 315  
Session: 47112345  
clab-clientSessionId: 8cd50800200c00000000
```

### 6.3.3 GET\_PARAMETER

#### Overview

GET\_PARAMETER RTSP requests are sent from the Client to the SRM to either act as a keep-alive mechanism or to retrieve some information from the SRM.

The request can be distinguished via the headers of content of the request body:

- ▶ Update expiry date of the session (**Session HEADER**)
- ▶ Retrieve a list of sessions (**clab-session-list BODY and clab-SessionGroup HEADER**)
- ▶ Retrieve the value for the session timeout (**clab-connection-timeout BODY and clab-SessionGroup HEADER**)

#### Request Format

To fetch the value of a parameter, the Bytel SRM shall send an RTSP GET\_PARAMETER request to the SRM. This is a regular RTSP request, with a body that specifies the parameter of interest. In (quasi-)BNF the format of this request shall be

```
GET_PARAMETER SP <uri> SP RTSP/1.0 CRLF  
1*<header> CRLF  
CRLF  
?<body>
```



**Request URI**

The request <uri> shall have the form:

```
rtsp://<logicalSrmAddress>:<logicalSrmPort>/ RTSP/1.0
```

**Request Headers**

Header	Format	Definition	Mandatory?
CSeq	Integer	Sequence number, which is incremented by one for each distinct request transmitted. It is a 64-bit decimal integer	Yes
Session	String	Session ID	No
Require	String	???? e.g.com.cablelabs.ermi	Yes
Content-Type	String	Type of content, e.g. text/parameters	Yes
Content-Length	Integer	Length of the body.	Yes if there's a body
clab-SessionGroup	String	For clab-session-list requests, only sessions belonging to the requested SessionGroup will be returned.	Yes for clab-session-list requests

**Response Format**

In (quasi-)BNF the format of the response to this request shall be:

```
RTSP/1.0 <responseCode> SP <reasonPhrase> CRLF
1*<header> CRLF
CRLF
?<body>
```

**Response Headers**

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format	Definition	Mandatory?
CSeq	Integer	64-bit decimal integer. An exact copy of the value in the request.	Yes
Content-Type	String	Type of content	Yes
Content-Length	Integer	Length of body	Yes

**Response Codes**

Code / phrase	Reason
200 OK	The request succeeded
400 Bad request	The request was incorrectly formatted.
401 Unauthorized	
402 Payment Required	
404 Not found	The requested session does not exist.
405 Method Not Allowed	
408 Request Time-out	
415 Unsupported Media Type	
451 Parameter not understood	A header or URI field was invalid. (Implementations may return 400 Bad request instead.)
453 Not Enough Bandwidth	There is not enough bandwidth to service the request.
454 Session Not Found	
461 Unsupported Transport	

Code / phrase	Reason
462 Destination Unreachable	
500 Internal error	There is an unspecified fault within the SRM.
503 Service Unavailable	

### 6.3.3.1 RTSP GET\_PARAMETER Examples

```
GET_PARAMETER * RTSP/1.0
CSeq: 5
Session: 118582287615149
Accept: text/parameters
```

```
RTSP/1.0 200 OK
CSeq: 4
Session: 118582287615132
Content-Type: text/parameters
Content-Length: 0
```

```
GET_PARAMETER rtsp://172.16.9.4:8184 RTSP/1.0
User-Agent: Thomson
CSeq: 002
Require: com.cablelabs.ermi
Content-Type: text/parameters
Content-Length: 17
clab-SessionGroup: SMARTVISION_CRM_GROUP

clab-session-list
```

```
RTSP/1.0
200 OK
CSeq: 002
Content-Type: text/parameters
Content-Length: 42

clab-session-list:
12345:00AF123456DE00000001;12346:00BD123456C200000021;12347:00CE123456AA00000A01
```

```
GET_PARAMETER rtsp://192.0.2.2 RTSP/1.0
```

```
CSeq: 314
Require: com.cablelabs.ermi
Content-Type: text/parameters
Content-Length: 25

clab-connection-timeout
```

```
RTSP/1.0 200 OK
CSeq: 314
Content-Type: text/parameters
Content-Length: 29

clab-connection-timeout:180
```

## 6.3.4 SET\_PARAMETER

### Overview

SET\_PARAMETER RTSP requests are sent from the Client to the SRM to act as a keep-alive mechanism.

### Request Format

To fetch the value of a parameter, the Bytel SRM shall send an RTSP GET\_PARAMETER request to the SRM. This is a regular RTSP request, with a body that specifies the parameter of interest. In (quasi-)BNF the format of this request shall be

```
SET_PARAMETER SP <uri> SP RTSP/1.0 CRLF
1*<header> CRLF
```

### Request URI

The request <uri> shall have the form:

```
rtsp://<logicalSrmAddress>:<logicalSrmPort>/ RTSP/1.0
```

### Request Headers

Header	Format	Definition	Mandatory?
CSeq	Integer	Sequence number, which is incremented by one	Yes

Header	Format	Definition	Mandatory?
		for each distinct request transmitted. It is a 64-bit decimal integer	
Require	String	e.g.com.cablelabs.ermi	Yes
Session	String	ID of the session	Yes

### Response Format

In (quasi-)BNF the format of the response to this request shall be:

```
RTSP/1.0 <responseCode> SP <reasonPhrase> CRLF
1*<header> CRLF
CRLF
```

### Response Headers

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format	Definition	Mandatory?
CSeq	Integer	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	String	Session ID	Yes
Content-Type	String	Type of content	Yes

### Response Codes

Code / phrase	Reason
200 OK	The request succeeded
400 Bad request	The request was incorrectly formatted.

Code / phrase	Reason
401 Unauthorized	
402 Payment Required	
404 Not found	The requested session does not exist.
405 Method Not Allowed	
408 Request Time-out	
415 Unsupported Media Type	
451 Parameter not understood	A header or URI field was invalid. (Implementations may return 400 Bad request instead.)
453 Not Enough Bandwidth	There is not enough bandwidth to service the request.
454 Session Not Found	
461 Unsupported Transport	
462 Destination Unreachable	
500 Internal error	There is an unspecified fault within the SRM.
503 Service Unavailable	

### 6.3.4.1 RTSP SET\_PARAMETER Examples

```
SET_PARAMETER rtsp://192.0.2.2/ RTSP/1.0
CSeq: 314
Require: com.cablelabs.ermi
Session: 47112344
```

```
RTSP/1.0 200 OK
CSeq: 314
Session: 47112344
```

Content-Type: text/parameters

## 6.3.5 ANNOUNCE handling

### Overview

In RTSP, ANNOUNCE messages serve three purposes:

1. Interrogative: a component may send an announce to query its upstream manager. For example, some types of ERM send announce messages to the SRM, to query whether a given session is still alive.
2. Session lifecycle announcements: Sometimes a component must change the behavior of a valid session on the fly.
3. Session failure announcements: If a session fails for some reason and must be torn down, the component which discovers the failure will usually send an ANNOUNCE to the session manager (ie. the SRM) to tell it to tear down the session.

In general, the SRM handles these 3 types of request as follows:

- ▶ The SRM handles interrogative announce messages internally and does not forward them on to other components.
- ▶ The SRM does not generate lifecycle announcements and does not expect to receive them: typically these are only used on STB to VS links.
- ▶ When the SRM receives a session failure announcement, it forwards it to the STB and initiates a session teardown. (The STB may also request a teardown of the failed session, and the SRM will respond 200 OK even if it has already torn down the session.) The Announce code and message is not remapped - it's sent verbatim. Other headers (eg. **Require:**) will be adjusted as required to conform to the STB dialect.
- ▶ When a session is torn down administratively on the SRM (forced-disconnect), the SRM issues a **5402** announce to the STB and initiates a session teardown. (The STB may also request a teardown of the killed session, and the SRM will respond 200 OK even though it has already started to tear down the session.)

Because the announce codes are forwarded verbatim, announce handling is a cross-cutting concern across dialects. This is why it gets this separate section.

### API-specific considerations

#### STB-to-VS dialects

These are beyond the scope of this guide.

#### NGOD 2.0 S1

NGOD defines a large number of valid ANNOUNCE codes. A conformant client should tolerate receiving them all, and should implement sensible defaults for codes it does not recognize. We recommend the following strategy for unrecognized codes:

Code range	Action
1xxx	Ignore

Code range	Action
2xxx	Ignore
4xxx	Treat as session failure
5xxx	Treat as session failure
6xxx or higher	Treat as session failure

**NagraVS (SRM to VS)**

This protocol does not define any announce codes. Therefore, in a deployment which uses it, the only ANNOUNCE codes which the STB may expect are 5402 (forced disconnect) and any announce codes generated by the ERM (if the deployment includes them).

**Note** The SRM's internal bandwidth manager does not generate announce messages.

**OpenTV 2.4 (STB to SRM)**

The OpenTV 2.4 specification defines (in Appendix A) a smaller list of defined ANNOUNCE codes. It also mandates a strategy for dealing with unknown codes (ignore any announce whose code is below 4000, tear down for any above).

The set of announce codes used by OpenTV 2.4 overlaps the set used by NGOD - it's neither a proper superset nor a proper subset. However, all codes which appear in both, have the same meaning in both. Combined with the mandated defaulting strategy, this fact means that OpenTV 2.4 STBs will interoperate correctly with NGOD upstream devices, with no remapping of announce codes required.

The SRM also observes a few additional restrictions:

- ▶ The SRM will not send 2401 (ticket expired) - it sends 5402 (session terminated) instead.
- ▶ The SRM will not send 5200 (server resources unavailable) unless this was sent by an upstream device. (The SRM does not shed load by killing active sessions.)
- ▶ The SRM will not send 5403 (server shutting down). In normal operation, an SRM instance which needs to be shut down, will be taken out of operation in a controlled manner which does not disconnect active sessions. Therefore the only situation in which this would be applicable is an emergency shutdown procedure. In this case, the SRM will not be able to wait for sessions to acknowledge the announce message, but must terminate abruptly instead.

**6.3.5.1 ANNOUNCE**

**Overview**

An ANNOUNCE is issued by the video-server following termination of a stream. The SRM uses this ANNOUNCE to determine that a session has been terminated and commence the reaping process.



## Request Format

An ANNOUNCE request is a regular RTSP request, with no body. In (quasi-)BNF the format of this request shall be:

```
ANNOUNCE SP <uri> SP RTSP/1.0 CRLF
1*<header> CRLF
```

## Request URI

The request <uri> shall have the form:

```
rtsp://<logicalSrmAddress>:<logicalSrmPort> RTSP 1.0
```

Field	Description
<logicalSrmAddress>	The DNS name or IP address which the Client uses to contact the SRM.
<logicalSrmPort>	The TCP port which the Client sends RTSP request to on the SRM.

## Request Headers

Header	Format / definition	Mandatory?
CSeq	Sequence number, which is incremented by one for each distinct request transmitted. It is a 64-bit decimal integer	Yes
Session	It is the vs sessionId that was generated by the Cisco VS.	Yes
Notice	A string message from the VS describing the reason for the ANNOUNCE	No

## Response Format

In (quasi-)BNF the format of the response to this request shall be:

```
RTSP/1.0 <responseCode> SP <reasonPhrase> CRLF
1*<header> CRLF
```

## Response Headers

If the request succeeds, the required / permitted headers are as defined in the table below. If the request fails for any reason, only the CSeq header will be included.

Header	Format / definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes

## Response Codes

Code / phrase	Reason
200 OK	The request succeeded
454 Not found	The requested session does not exist.
500 Internal error	There is an unspecified fault within the SRM. This also occurs when there is a bad CSeq or Session header

```
ANNOUNCE * RTSP/1.0^M
CSeq: 304^M
Session: 19924022^M
Notice: 5402 Event-Date=20130419T094406.965Z "Client Session Terminated"^M
RTSP/1.0 454 Session Not Found
CSeq: 304
x-qsp-exception-class: tv.quative.service.gateway.rtsp.RtspException
x-qsp-exception-message: No active session
RTSP/1.0 200 OK
CSeq: 307
```

### 6.3.5.2 ANNOUNCE Examples

```
ANNOUNCE * RTSP/1.0
CSeq: 307
Session: 1251233973
```

```
RTSP/1.0 200 OK
CSeq: 307
```

```
RTSP/1.0 454 Session Not Found
CSeq: 304
x-qsp-exception-class:tv.quative.service.gateway.rtsp.RtspException
x-qsp-exception-message: No active session
```

## 6.3.6 SDV

### 6.3.6.1 SDV SETUP, DESCRIBE, PLAY

#### Overview

SDV is basically 'broadcast over VOD'. This feature is used to allow service providers to save bandwidth, where QAM would only modulate on-demand-channels streams into the medium. In addition, an ERM resource for a channel is used among multiple clients.

This API is featured when a client sends a SETUP, PLAY or DESCRIBE RTSP requests to the SRM, in order to watch a particular channel.

#### Client-to-SRM Protocol

The protocol is a dialect of RTSP, which is [https://atlassian.hq.k.grp/confluence/download/attachments/24215713/RTSP\\_iopguide\\_06may2011\\_%5B1%5D.pdf?api=v2](https://atlassian.hq.k.grp/confluence/download/attachments/24215713/RTSP_iopguide_06may2011_%5B1%5D.pdf?api=v2) for the SDV. It is based on 3 commands, i.e. SETUP, PLAY and DESCRIBE.

#### Overall Flow Diagram

Clients should follow this use case to SETUP an SDV session on SRM, and call DESCRIBE and PLAY.

```
@startuml
actor User as Userparticipant
actor Client as Clientparticipant
actor SRM as SRMparticipant
actor PAV as PAVparticipant
actor ERM as ERMgroup
actor OTV as OTV
Client->>SRM: [RTSP] SETUP (URI, headers, devId)
SRM->>PAV: [SOAP] isAuthorized (chan_originKey, devId)
SRM<-<-PAV: authorized / notAuthorized
SRM->>ERM: [RTSP] SETUP
SRM<-<-ERM: ERM sessionId, tuning details
Client->>SRM: SRM sessionId, tuning details
endgroup
OTV DESCRIBE on Client
Client->>SRM: [RTSP] DESCRIBE (sessionId, headers)
Client<-<-SRM: 200 OK
endgroup
OTV PLAY on Client
Client->>SRM: [RTSP] PLAY (sessionId, headers)
Client<-<-SRM: 200 OK
end
```

Reference	<TBD>
Main actor	SRM
Secondary actors	PAV, ERM, Client application

Pre-Conditions	SRM, PAV and ERM are running and operational.
Trigger	Client sends a SETUP, PLAY or DESCRIBE RTSP request to the SRM

---

## SETUP Steps

1. Client sends a SETUP RTSP request to the SRM, using the API when switching to watch a channel.
2. SRM validates incoming request URI and headers.
3. SRM sends a request to the PAV for authorization, using the API <TBD>.
4. SRM acquires authorization from PAV.
5. SRM sends a SETUP rtsp request to ERM, using the API SETUP.
6. SRM acquires an ERM sessionId with tuning details.
7. SRM returns SRM sessionId with tuning details to the client.

## SETUP Extensions

Ref	From step	Description
A	2	If URI is not available or a header(s) is missing, SRM returns an error response indicating the failure.
B	3	If PAV is unreachable, SRM returns an error response indicating the error.
C	4	If PAV returns unauthorized, SRM returns an error response indicating the error.
D	5	If ERM is unreachable, SRM returns an error response indicating the error.
E	6	If ERM does not return a resource, SRM returns an error response indicating the error.

---

## DESCRIBE Steps

1. Client sends a DESCRIBE RTSP request to the SRM, using the API [DESCRIBE](#) <sup>p.181</sup> after SETUP has been initialised.
2. SRM validates incoming request headers.
3. SRM returns presentation length information to the client.

## DESCRIBE Extensions

Ref	From step	Description
A	2	If sessionId is missing / invalid, SRM returns an error response indicating the failure.
B	2	If a header is missing, SRM returns an error response indicating the error.

## PLAY Steps

1. Client sends a PLAY RTSP request to the SRM, using the API [PLAY](#) <sup>p.184</sup> after SETUP has been initialised.
2. SRM validates incoming request headers.
3. SRM returns response to the client to tune to the network and receive the content.

## PLAY Extensions

Ref	From step	Description
A	2	If sessionId is missing / invalid, SRM returns an error response indicating the failure.
B	2	If a header is missing, SRM returns an error response indicating the error.

### 6.3.6.2 DESCRIBE

*This page is still under review*

#### Overview

DESCRIBE RTSP request is sent from Client to SRM to request presentation information for the channel being viewed; this is normally sent after a SETUP request.

The client must issue this DESCRIBE request and the server must issue a DESCRIBE response in NPT format that contains the requested information. As the SRM serves traffic for both VOD and Management servers in this case, the response to these requests will be static.

#### Request Format

A DESCRIBE request is a regular RTSP request, with no body. In (quasi-)BNF the format of this request shall be:

```
DESCRIBE SP <uri> SP RTSP/1.0 CRLF
1*<header> CRLF
CRLF
?<body>
```

## Request URI

The request <uri> will have the form:

rtsp://<logicalSrmAddress>:<logicalSrmPort> RTSP 1.0

Field	Description
<logicalSrmAddress>	The DNS name or IP address which the Client uses to contact the SRM.
<logicalSrmPort>	The TCP port which the Client sends RTSP request to on the SRM.

## Request Headers

Header	Format / definition	Mandatory?
CSeq	Sequence number, which is incremented by one for each distinct request transmitted. It is a 64-bit decimal integer	Yes
Session	It is the client sessionId that was returned by the SRM in the SETUP response. An alphanumeric session ID, Max 20 characters (for compatibility with OpenTV).	Yes
Accept	The Accept header contains an application/sdp value for the media type.	Yes

## Response Format

In (quasi-)BNF the format of the response to this request shall be:

```
RTSP/1.0 <responseCode> SP <reasonPhrase> CRLF
```

```
1*<header> CRLF
CRLF
<body>
```

## Response Headers

If the request succeeds, the required / permitted headers are as defined in the below table. If the request fails for any reason, only the CSeq header will be included.

Header	Format / definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	An alphanumeric session ID; Max 20 characters (for compatibility with OpenTV).	Yes
Content-Type	Contains a "application/sdp" value for the media type.	Yes
Content-Length	It denotes the total length of parameters and their values returned in the content body of the response.  The Content-Length value must account for each 2-byte CRLF contained within the content body.	Yes
<body>	It contains information about the current presentation using the following parameters: i=, a=type:, and a=range:npt.	Yes

## Response Codes

Code / phrase	Reason
200 OK	The request succeeded
400 Bad request	The request was incorrectly formatted.

Code / phrase	Reason
404 Not found	The requested session does not exist.
451 Parameter not understood	A header or URI field was invalid. (Implementations may return 400 Bad request instead.)
500 Internal error	There is an unspecified fault within the SRM.

### 6.3.6.2.1 RTSP DESCRIBE Examples

```
DESCRIBE rtsp://192.168.0.99:1999 RTSP/1.0
CSeq: 1
Session: 118582287615149
Accept: application/sdp
```

```
RTSP/1.0 200 OK
CSeq: 1
Session: 118582287615149
Content-Type: application/sdp
Content-Length: 163
```

```
v=0
o=-3486817297 3486817297 IN IP4 172.16.7.27
s=RTSP Session
t=0 0
m=video 0 udp M2T
c=IN IP4 0.0.0.0
i=video12.mpg
a=type:vod
a=range:npt=0.0-10207.320
```

### 6.3.6.3 PLAY



## Overview

PLAY RTSP request is sent from Client to SRM to start a presentation stream for the current session. The response from the SRM server must contain the expected headers and information. This request is usually sent after a DESCRIBE request for the SDV Channel being consumed.

As the SRM serves traffic for both VOD and Management servers in this case, the response to these requests will be static.

## Request Format

A PLAY request is a regular RTSP request, with no body. In (quasi-)BNF the format of this request shall be:

```
PLAY SP <uri> SP RTSP/1.0 CRLF
1*<header> CRLF
CRLF
?<body>
```

## Request URI

The request <uri> shall have the form:

```
rtsp://<logicalSrmAddress>:<logicalSrmPort> RTSP 1.0
```

Field	Description
<logicalSrmAddress>	The DNS name or IP address which the Client uses to contact the SRM.
<logicalSrmPort>	The TCP port which the Client sends RTSP request to on the SRM.

## Request Headers

Header	Format / definition	Mandatory?
CSeq	Sequence number, which is incremented by one for each distinct request transmitted. It is a 64-bit decimal integer	Yes
Session	It is the client sessionId that was generated by the SRM in the SETUP call. An alphanumeric session ID. Max 20 characters (for compatibility with OpenTV).	Yes

Header	Format / definition	Mandatory?
Range	<p>Sets the starting point for playing a presentation; this will always be "npt=0-" to represent the beginning of the stream.</p> <p>The format of this value is defined by the OTV2.4 specification:-</p> <p>["Range" ":" SP "npt=" npt-time ["-" npt-time] CRLF] npt-time = now   npt-sec   npt-hhmmss now = "now" ; in first range only to denote current time npt-sec = 1*DIGIT ["." *DIGIT] npt-hhmmss = npt-hh ":" npt-mm ":" npt-ss ["." *DIGIT] npt-hh = 1*2DIGIT ; any positive number npt-mm = 1*2DIGIT ; 0-59 npt-ss = 1*2DIGIT ; 0-59</p>	Yes
Scale	Denotes the speed of play back, 1 denotes normal playback speed.	Yes

## Response Format

In (quasi-)BNF the format of the response to this request shall be:

```
RTSP/1.0 <responseCode> SP <reasonPhrase> CRLF
1*<header> CRLF
CRLF
?<body>
```

## Response Headers

If the request succeeds, the required / permitted headers are as defined in the table below. If the request fails for any reason, only the CSeq header will be included.

Header	Format / definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	An alphanumeric session ID. Max 20 characters (for compatibility with OpenTV).	Yes

Header	Format / definition	Mandatory?
Range	<p>Sets the starting point for playing a presentation, "npt=0-" represents the beginning of the stream.</p> <p>The format of this value is defined by the OTV2.4 specification:-</p> <p>["Range" ":" SP "npt=" npt-time ["-" npt-time] CRLF] npt-time = now   npt-sec   npt-hhmmss now = "now" ; in first range only to denote current time npt-sec = 1*DIGIT ["." *DIGIT] npt-hhmmss = npt-hh ":" npt-mm ":" npt-ss ["." *DIGIT] npt-hh = 1*2DIGIT ; any positive number npt-mm = 1*2DIGIT ; 0-59 npt-ss = 1*2DIGIT ; 0-59</p>	Yes
Scale	Denotes the speed of play back, 1 denotes normal playback speed.	Yes

## Response Codes

Code / phrase	Reason
200 OK	The request succeeded
400 Bad request	The request was incorrectly formatted.
404 Not found	The requested session does not exist.
451 Parameter not understood	A header or URI field was invalid. (Implementations may return 400 Bad request instead.)
500 Internal error	There is an unspecified fault within the SRM.

### 6.3.6.3.1 RTSP PLAY Examples

```
PLAY rtsp://192.168.0.99:1999 RTSP/1.0
CSeq: 1
Session: 118582287615149
```

```
Range: npt=0-
Scale: 1
```

```
RTSP/1.0 200 OK
CSeq: 1
Session: 118582287615149
Range: npt=0-
Scale: 1
```

```
:
Range: npt=0-
Scale: 1
```

## 6.3.7 OpenTV 2.4

### Overview

The [https://atlassian.hq.k.grp/confluence/download/attachments/24215713/RTSP\\_iopguide\\_06may2011\\_%5B1%5D.pdf?api=v2](https://atlassian.hq.k.grp/confluence/download/attachments/24215713/RTSP_iopguide_06may2011_%5B1%5D.pdf?api=v2) defines protocols for an STB to interact with a VS and a session manager. The SRM supports the OpenTV session manager role, based on the **First Configuration** deployment architecture defined in that spec.

### Limitations and clarifications

1. In SETUP requests, the client string in the Transport header must have the form `servicegroupName.smartcardId`, where `servicegroupName` is the unique ingest name of the ServiceGroup - see [Data ingest: ServiceGroup fields](#)<sup>p.73</sup>. In many deployments, the `servicegroupName` is of the form `networkId.nodeId`, giving a client string of the form `networkId.nodeId.smartcardId`. This form is encouraged.
2. In SETUP requests for VOD and persistent-CU, the RTSP URI must have the form `rtsp://srmHost:srmPort/filename`
  - `srmHost` is the DNS name or IP address used by the STB to make the request.
  - `srmPort` is the TCP port used by the STB to make the request.
  - `filename` is a unique filename used to identify the Content-Version's (VOD) / BTV Event's (persistent-CU) Media.
3. In SETUP requests for buffered-CU the RTSP URI must have the form `rtsp://srmHost:srmPort/sContentId?begin=beginTime&end=endTime`
  - `sContentId` identifies the channel. It shall be the CMS public id.
  - `startTime` and `endTime` identify the interval to play back. Both shall be in compact ISO time format, with the exception that the effective timezone shall always be UTC, and the timezone identifier shall be omitted.
  - NB. the begin and end clauses **must** be in the order shown.
4. In SETUP responses, the Session and ControlSession IDs may be the same. However, for the sake of robustness, STBs should not assume this.

5. The STB must also implement a suitable protocol with which to talk to the VS. Typically it will use the OpenTV 2.4 dialect (NB. this is different from the STB-to-SRM protocol described here) but the SRM does not mandate this.
6. OpenTV "Second configuration" is not supported. Although at first glance it's usefully simpler, it cannot be used here because it does not enable the SRM to discover when a session terminates; there is thus no way to free allocated QAM bandwidth in a timely manner. (This could work where VOD is delivered via an over-provisioned IP network instead of via QAM.)
7. OpenTV "Third configuration" is not supported because the SRM and VS are separate servers.

## 6.3.8 NGOD C1 Interface

### Overview

The On Demand Clients interact with the Streaming Server with the Trick Play Requests through SRM. SRM acts as proxy server for all trick play requests.

#### Interaction Diagram

The below diagram depicts the interaction between the On Demand Client, SRM and the Streaming Server to control playback of a stream.

@startuml  
 participant STB as participant STB  
 participant SRM as participant SRM  
 participant StreamingServer as StreamingServer  
 STB->>SRM: [RTSP] TRICK Request  
 SRM-->>StreamingServer: [RTSP] TRICK Request  
 StreamingServer-->>SRM: [RTSP] Response  
 SRM-->>STB: [RTSP] Response  
 @enduml

### 6.3.8.1 C1 ANNOUNCE

#### Overview

On some occasions the Streaming Server will send unsolicited messages to the On Demand Clients that have active sessions running. These messages shall be sent via RTSP Announce Request and Response messages as detailed below.

#### Interaction Diagram

The below diagram depicts the interaction between the Streaming Server and the On Demand Client to notify it of information relating to the session.

#### Request Format

In (quasi-) BNF the format of this request shall be

---

```
ANNOUNCE SP <uri> SP RTSP/1.0 CRLF
1 *<header> CRLF
```

---

**Request URI**

The request <uri> shall have the form:

---

```
rtsp: //<StreamingServerHost>:<StreamingServerPort>/
```

---

Field	Description
<StreamingServerHost>	The DNS name or IP address of Streaming Server.
<StreamingServerPort>	The TCP port of Streaming Server.

**RTSP Request Headers**

The required / permitted headers in the request are as defined in the below table.

Header	Format / Definition	Mandatory
CSeq	64-bit decimal integer identifying the request.	Yes
Require	The Require header defines a mechanism by which the client indicates in a request that certain extensions are required to fulfill this request. The syntax of the Require header is as follows:-  Require: com.comcast.ngod.c1	
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.  "Session" ":" session-id  Session header example Session: 10000105	Yes

Header	Format / Definition	Mandatory
Notice	<p>The Notice header defines information sent from a RTSP server to a RTSP client within an ANNOUNCE message.</p> <p>The syntax of the Notice header is as follows. Notice = "Notice" HCOLON event *(COMMA event)</p> <p>HCOLON = * ( SP / HT ) ":"            SWSSP = %X20HT = %X09SWS            = [ LWS ]LWS = [ CRLF ] 1 * ( SP / HT )CRLF = CR LFCR = %X0DLF = %X0A</p> <p>event = announce-code SP text-description SP event-date SP npt</p> <p>announce-code = ; For Announce Notice Codes please refer to <a href="#">NGOD Response And Notice Codes</a> <sup>p.262</sup> text-description = quoted-stringquoted-string = ( DQ * qdtext DQ )qdtext = %X20-21 / %X23-%X7E / %80-%XFFDQ = %X22</p> <p>event-date = "event-date" EQUAL utc-time</p> <p>EQUAL = SWS "=" SWSutc-time = utc-date "T" utc-clock "Z"utc-date = 8DIGIT ; Year (4) Month (2) Date (2)utc-clock = 6DIGIT [ "." fraction ] ; Hour (2) Minute (2) Second (2)fraction = 1 * DIGITDIGIT = %X30 - %X39</p> <p>npt = "npt" EQUAL [ npt - sec ] ; only if value is known</p> <p>npt-sec = 1 * DIGIT [ "." * DIGIT ] ; position as decimal sec.msec</p> <p>The NPT must be specified in decimal seconds.milliseconds format. This format is consistent with the [RFC 2326] definition. Where Notice is used but npt is not known, &lt;npt-sec&gt; (but not the attribute) will be omitted.</p> <p>Example of the Notice header for end of stream</p>	Yes

Header	Format / Definition	Mandatory
	<p>(2314.223 sec is actual end of content): Notice: 2101 "End-of-Stream Reached" event-date=20150316T064707.735Z npt=2314.223</p> <p>Example of the Notice header for beginning of stream: Notice: 2101 "Start-of-Stream Reached"</p> <p>For Announce Notice Codes please refer to <a href="#">NGOD Response And Notice Codes</a> <sup>p.262</sup></p>	

**Request Example**

```
ANNOUNCE rtsp://videosever234.nagra.com:554/ RTSP/1.0
CSeq: 456Require: com.comcast.ngod.c1Session: 77Notice: 2101 "End-of-Stream Reached" event-date=20040316T064707.735Z npt=2314.223
```

**C1 ANNOUNCE Response**

**Response Format**

```
RTSP/ 1.0 <responseCode> SP <reasonPhrase> CRLF
1 *<header> CRLF
```

**Response Headers**

The required / permitted headers in the response are as defined in the below table.

Header	Format / Definition	Mandatory
CSeq	64-bit decimal integer indentifying the response; this will include an exact copy of the value in the request.	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response.The session	Yes



Header	Format / Definition	Mandatory
	<p>identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.</p> <p>"Session" ":" session-id</p> <p>Session header example Session: 10000105</p>	

**Response Example**

RTSP/1.0 200 OKCSeq: 456Session: 85377815835034221373

For Response Codes please refer to [NGOD Response And Notice Codes](#) <sup>p.262</sup>

**6.3.8.2 C1 GET\_PARAMETER**

**Warning!** NOT currently supported.

**Overview**

The On Demand Clients interact with the Streaming Server with the RTSP Get\_Parameter Request and Response messages to retrieve information about the state of the stream.

**Interaction Diagram**

The below diagram depicts the interaction between the On Demand Client and the Streaming Server for a Get\_Parameter interaction.

**Request Format**

In (quasi-) BNF the format of this request shall be

```
GET_PARAMETER SP <uri> SP RTSP/ 1.0 CRLF
1 *<header> CRLF
CRLF
<body>
```

**Request URI**

The request <uri> shall have the form:

---

rtsp: //<StreamingServerHost>:<StreamingServerPort>/

---

Field	Description
<StreamingServerHost>	The DNS name or IP address of Streaming Server.
<StreamingServerPort>	The TCP port of Streaming Server.

### RTSP Request Headers

The required / permitted headers in the request are as defined in the below table.

Header	Format / Definition	Mandatory
CSeq	64-bit decimal integer identifying the request.	Yes
Require	The Require header defines a mechanism by which the client indicates in a request that certain extensions are required to fulfill this request. The syntax of the Require header is as follows:-  Require: com.comcast.ngod.c1	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.  "Session" ":" session-id  Session header example Session: 10000105	Yes
Content-type	The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient	Yes

Header	Format / Definition	Mandatory
	<p>Content-Type = "Content-Type" ":" media-type</p> <p>An example is</p> <p>Content-Type: text/parameters</p>	
Content-length	<p>The Content-Length entity-header field indicates the size of the entity-body, in decimal number of OCTETs, sent to the recipient.</p> <p>Content-Length = "Content-Length" ":" 1*DIGIT</p> <p>An example is</p> <p>Content-Length: 3495</p>	Yes

**Request Example**

```
GET_PARAMETER rtsp://videosever234.nagra.com:554/ RTSP/1.0CSeq: 456Require: com.comcast.ngod.
c1Session: 72767478Content-Type: text/parameterContent-Length: 16
presentation_state
```

**Parameter Options**

Parameter Type	Description
connection_timeout	<p>Timeout setting for activity on a connection.</p> <p>The RTSP server shall support a parameter of connection_timeout. The return value from a get_parameter of connection_timeout shall be as follows</p> <p>connection_timeout: &lt;timeout&gt;</p> <p>where</p> <p>&lt;timeout&gt; is an integer representing seconds.</p> <p>for example: connection_timeout: 300</p>
session_list	<p>List of current active sessions.</p> <p>The RTSP server shall support a parameter of session_list. As an optional header the RTSP server shall support SessionGroup. The return value conveys a list of IDs for active sessions that were setup within the specified Session Group (if provided). If the SessionGroup header is omitted, all active session IDs will be returned.</p>

Parameter Type	Description
	<p>The syntax for the return value for a GET_PARAMETER session_list is as follows.            session_list: [&lt;rtsp-session-id&gt;:&lt;on-demand-session-id&gt;]&lt;CRLF&gt;[&lt;rtsp-session-id&gt;:&lt;on-demand-session-id&gt;]*where            &lt;rtsp-session-id&gt; is the RTSP server's session ID            &lt;on-demand-session-id&gt; is the OnDemandSessionId generated by the Session Manager.            An example return value for a GET_PARAMETER session_list follows            session list: 12345:b50557b            Ofecc11d98cd60800200c9a6612346:dec1b300fecc11d98cd60800200c9a6612347:0257ce80fecc</p>
position	<p>The current stream position. The NPT values are in seconds, consistent with the ntp-sec as defined in RFC 2326bis07</p>
presentation_state	<p>Current state of the stream            The list of possible return values for presentation_state is as follows.            init            ready            play            pause            Fast forward will be represented with a presentation_state of "play" and a scale greater than 1.0. Rewind will be represented with a presentation_state of "play" and a negative scale value</p>
scale	<p>The current play scale. E.g. 1.0, 7.0, -7.0            Please refer to RTSP RFC2326 for full definition of Scale header.</p>

**C1 GET\_PARAMETER Response**

**Response Format**

```
RTSP/ 1.0 <responseCode> SP <reasonPhrase> CRLF
1 *<header> CRLF
CRLF
<body>
```

**Response Headers**

The required / permitted headers in the response are as defined in the below table.

Header	Format / Definition	Mandatory
CSeq	64-bit decimal integer indentifying the response; this will include an exact copy of the value in the request.	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.  "Session" ":" session-id Session header example Session: 10000105	Yes
Content-type	The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient  Content-Type = "Content-Type" ":" media-type An example is Content-Type: text/ parameters	Yes
Content-length	The Content-Length entity-header field indicates the size of the entity-body, in decimal number of OCTETs, sent to the recipient.  Content-Length = "Content-Length" ":" 1*DIGIT An example is Content-Length: 3495	Yes

**Response Example**

RTSP/1.0 200 OK  
CSeq: 456Session: 72767478Content-Type: text/parameterContent-Length: 23  
presentation\_state: play

For Response Codes please refer to <https://atlassian.hq.k.grp/confluence/display/SRM/NGOD+Response+And+Notice+Codes>

### 6.3.8.3 C1 PAUSE

**Warning!** NOT currently supported.

#### Overview

The On Demand Clients interact with the Streaming Server with the RTSP Pause Request and Response messages to pause the video play-out.

#### Interaction Diagram

The below diagram depicts the interaction between the On Demand Client and the Streaming Server to exact a pause.

#### Request Format

In (quasi-) BNF the format of this request shall be

```
PAUSE SP <uri> SP RTSP/ 1.0 CRLF
1 *<header> CRLF
```

#### Request URI

The request <uri> shall have the form:

```
rtsp: //<StreamingServerHost>:<StreamingServerPort>/
```

Field	Description
<StreamingServerHost>	The DNS name or IP address of Streaming Server.
<StreamingServerPort>	The TCP port of Streaming Server.

#### RTSP Request Headers

The required / permitted headers in the request are as defined in the below table.

Header	Format / Definition	Mandatory
CSeq	64-bit decimal integer identifying the request.	Yes
Require	The Require header defines a mechanism by which the client indicates in a request that certain extensions are required to fulfill this request. The syntax of the Require header is as follows:- Require: com.comcast.ngod.c1	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.  "Session" ":" session-id Session header example Session: 10000105	Yes

**Request Example**

PAUSE rtsp://videosever234.nagra.com:554/ RTSP/1.0CSeq: 456Require: com.comcast.ngod.c1Session: 77

**C1 PAUSE Response**

**Response Format**

RTSP/ 1.0 <responseCode> SP <reasonPhrase> CRLF  
1 \*<header> CRLF

**Response Headers**

The required / permitted headers in the response are as defined in the below table.

Header	Format / Definition	Mandatory
CSeq	64-bit decimal integer identifying the response; this will include an exact copy of the value in the request.	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.  "Session" ":" session-id Session header example Session: 10000105	Yes
Range	<range> is the playout range in the form: [<start-npt>]-[<stop-npt>]  If not provided the stream will start from beginning. Please refer to RTSP RFC2326 for full definition of Range header.	Yes

**Response Example**

RTSP/1.0 200 OKCSeq: 836Session: 77Range: npt=1.742-

For Response Codes please refer to <https://atlassian.hq.k.grp/confluence/display/SRM/NGOD+Response+And+Notice+Codes>

**6.3.8.4 C1 PLAY**

**Overview**

The On Demand Clients interacts with the Streaming Server with the RTSP Play Request and Response messages. This enable playback at normal (1x) speed, rewinding and fast-forwarding the stream.

**Interaction Diagram**

The below diagram depicts the interaction between the On Demand Client and the Streaming Server to exact a pause.



### Request Format

In (quasi-) BNF the format of this request shall be

```
PLAY SP <uri> SP RTSP/1.0 CRLF
1 *<header> CRLF
```

### Request URI

The request <uri> shall have the form:

```
rtsp://<StreamingServerHost>:<StreamingServerPort>/
```

Field	Description
<StreamingServerHost>	The DNS name or IP address of Streaming Server.
<StreamingServerPort>	The TCP port of Streaming Server.

### RTSP Request Headers

The required / permitted headers in the request are as defined in the below table.

Header	Format / Definition	Mandatory
CSeq	64-bit decimal integer identifying the request.	Yes
Require	The Require header defines a mechanism by which the client indicates in a request that certain extensions are required to fulfill this request. The syntax of the Require header is as follows:- Require: com.comcast.ngod.c1	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media	Yes

Header	Format / Definition	Mandatory
	<p>server. Once a client receives a Session identifier, it MUST return it for any request related to that session.</p> <p>"Session" ":" session-id</p> <p>Session header example Session: 10000105</p>	
Scale	<p>The current play scale. E.g. 1.0, 7.0, -7.0</p> <p>Please refer to RTSP RFC2326 for full definition of Scale header.</p>	No
Range	<p>&lt;range&gt; is the playout range in the form: [&lt;start-npt&gt;]-&lt;stop-npt&gt;</p> <p>If not provided the stream will start from beginning. Please refer to RTSP RFC2326 for full definition of Range header.</p>	No

**Request Example**

An example of a normal play follows. `PLAY rtsp://videosever234.nagra.com:554/ RTSP/1.0CSeq: 456Require: com.comcast.ngod.c1Session: 77Range: npt=0.0-`

An example of a rewind -5x play follows. `PLAY rtsp://videosever234.nagra.com:554/ RTSP/1.0CSeq: 456Require: com.comcast.ngod.c1Session: 77Scale: -5.0`  
 An example of a fast-forward at 5x play follows. `PLAY rtsp://videosever234.nagra.com:554/ RTSP/1.0CSeq: 456Require: com.comcast.ngod.c1Session: 77Scale: 5.0`

**C1 PLAY ResponseResponse Format**

`RTSP/ 1.0 <responseCode> SP <reasonPhrase> CRLF`  
`1 *<header> CRLF`

**Response Headers**

The required / permitted headers in the response are as defined in the below table.

Header	Format / Definition	Mandatory
CSeq	64-bit decimal integer indentifying the response; this will include an exact copy of the value in the request.	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.  "Session" ":" session-id Session header example Session: 10000105	Yes
Range	<range> is the playout range in the form: [<start-npt>]-[<stop-npt>]  If not provided the stream will start from beginning. Please refer to RTSP RFC2326 for full definition of Range header.	No

**Response Example**

RTSP/1.0 200 OKCSeq: 456Session: 77Range: npt=0.0-

For Response Codes please refer to <https://atlassian.hq.k.grp/confluence/display/SRM/NGOD+Response+And+Notice+Codes>

**6.3.8.5 C1 SET\_PARAMETER**

**Warning!** NOT currently supported.

**Overview**

An On Demand Client interacts with the Streaming Server via the RTSP SET\_PARAMETER request and response messages to provide the Streaming Server with a list of sessions being managed by a single On Demand Client connection.

**Interaction Diagram**

The below diagram depicts the interaction between the On Demand Client and the Streaming Server for a Set\_Parameter interaction.

**Request Format**

In (quasi-) BNF the format of this request shall be

```
SET_PARAMETER SP <uri> SP RTSP/1.0 CRLF
1 *<header> CRLF
CRLF
<body>
```

**Request URI**

The request <uri> shall have the form:

```
rtsp://<StreamingServerHost>:<StreamingServerPort>/
```

Field	Description
<StreamingServerHost>	The DNS name or IP address of Streaming Server.
<StreamingServerPort>	The TCP port of Streaming Server.

**RTSP Request Headers**

The required / permitted headers in the request are as defined in the below table.

Header	Format / Definition	Mandatory
CSeq	64-bit decimal integer identifying the request.	Yes
Require	The Require header defines a mechanism by which the client indicates in a request that certain extensions are required to fulfill this request. The syntax of the Require header is as follows:- Require: com.comcast.ngod.c1	Yes

Header	Format / Definition	Mandatory
Content-type	The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient  Content-Type = "Content-Type" ":" media-type  An example is Content-Type: text/ parameters	Yes
Content-length	The Content-Length entity-header field indicates the size of the entity-body, in decimal number of OCTETs, sent to the recipient.  Content-Length = "Content-Length" ":" 1*DIGIT  An example is Content-Length: 3495	Yes

**Request Example**

```
SET_PARAMETER rtsp://videosever234.nagra.com:554/ RTSP/1.0CSeq: 456Require: com.comcast.ngod.
c1Content-Type: text/parameterContent-Length: 52
session_list: 12345:b50557bOfecclld98cd60800200c9a66
```

**Parameter Options**

When using the SET\_PARAMETER method in C1 the following parameters must be supported.

Parameter	Description
session_list	Normally the connection between the RTSP client and server is persistent and ANNOUNCE methods can be sent across it.  However, if the connection is broken due to an unexpected failure, there needs to be a way for an RTSP client to indicate which connection will be used for session communications. An RTSP client MAY send a list of sessions using the parameter session_list. This is a signal to the RTSP server that these sessions will be controlled via the connection over which the SET_PARAMETER method was received. Should an ANNOUNCE method be required for any session in the session list, the RTSP server will use the connection over which the session_list was received for sending the ANNOUNCE method.  The syntax for session_list is as follows. session_list: [<rtsp-session-id>:<on-demand-session-id>]<CRLF>[<rtsp-session-id>:<on-demand-session-id>]*where

Parameter	Description
	<p>&lt;rtsp-session-id&gt; is the RTSP server's session ID</p> <p>Example:</p> <pre>SET_PARAMETER rtsp://videosever234.comcast.com:554 RTSP/1.0 CSeq: 2 Require: com.comcast.ngod.c1 Content-Type: text/parameters Content-Length: 52 session_list: 12345:b50557b0fecc11d98cd60800200c9a66</pre>

### C1 SET\_PARAMETER Response

#### Response Format

```
RTSP/1.0 <responseCode> SP <reasonPhrase> CRLF
1 *<header> CRLF
```

#### Response Headers

The required / permitted headers in the response are as defined in the below table.

Header	Format / Definition	Mandatory
CSeq	64-bit decimal integer identifying the response; this will include an exact copy of the value in the request.	Yes

#### Response Example

```
RTSP/1.0 200 OK
CSeq: 456
```

For Response Codes please refer to <https://atlassian.hq.k.grp/confluence/display/SRM/NGOD+Response+And+Notice+Codes>

## 6.3.9 NGOD R2 Interface

## Overview

NGOD R2 is the interface through which the SRM communicates with streaming servers (video servers). The SRM acts as an NGOD ODRM (On-Demand Resource Manager). The protocol for R2 communication is Real-time Streaming Protocol (RTSP). Sessions are established and torn down via interactions over R2 between the SRM and Streaming Server. The Streaming Server may inform the SRM regarding session state changes.

### 6.3.9.1 R2 ANNOUNCE

#### R2 ANNOUNCE Request

Occasionally the Streaming Server will send unsolicited messages to the SRM regarding active sessions. These messages shall be sent via RTSP Announce Request and Response messages as detailed below.

##### Interaction Diagram

The diagram below depicts the interaction between the SRM and the Streaming Server to notify the SRM of information relating to the session.

##### Request Format

In (quasi-) BNF the format of this request shall be

```
ANNOUNCE SP <uri> SP RTSP/ 1.0 CRLF
1 *<header> CRLF
```

##### Request URI

The request <uri> shall have the form:

```
rtsp: //<SRMServerHost>:<SRMServerPort>/
```

Field	Description
<SRMServerHost>	The DNS name or IP address of SRM Server.
<SRMServerPort>	The TCP port of SRM Server.

##### RTSP Request Headers

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / Definition	Mandatory
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Notice	<p>The Notice header defines information sent from a RTSP server to a RTSP client within an ANNOUNCE message.</p> <p>The syntax of the Notice header is as follows. Notice = "Notice" HCOLON event *(COMMA event)</p> <p>HCOLON = * ( SP / HT ) ":"            SWSSP = %X20HT = %X09SWS            = [ LWS ]LWS = [ CRLF ] 1 * ( SP / HT )CRLF = CR LFCR = %X0DLF = %X0A</p> <p>event = announce-code SP text-description SP event-date SP npt</p> <p>announce-code = ; see Table for the list of possible values            text-description = quoted-string            quoted-string = ( DQ * qdtext DQ )            qdtext = %X20-21 / %X23—%X7E / %80—%XFF            DQ = %X22</p> <p>event-date = "event-date" EQUAL utc-time</p> <p>EQUAL = SWS "=" SWS            utc-time = utc-date "T" utc-clock "Z"            utc-date = 8DIGIT ; Year (4) Month (2) Date (2)            utc-clock = 6DIGIT [ "." fraction ] ; Hour (2) Minute (2) Second (2)            fraction = 1 * DIGITDIGIT = %X30 - %X39</p> <p>npt = "npt" EQUAL [ npt - sec ] ; only if value is known            npt-sec = 1 * DIGIT [ "." * DIGIT ] ; position as decimal sec.msec</p> <p>The NPT must be specified in decimal seconds.milliseconds format. This format is consistent with the [RFC 2326] definition. Where Notice is used but npt is not known, &lt;npt-sec&gt; (but not the attribute) will be omitted.</p> <p>Example of the Notice header for end of stream</p>	Yes



Header	Format / Definition	Mandatory
	<p>(2314.223 sec is actual end of content): Notice: 2101 "End-of-Stream Reached" event-date=20150316T064707.735Z npt=2314.223</p> <p>Example of the Notice header for beginning of stream: Notice: 2101 "Start-of-Stream Reached"</p> <p>Please see Section ANNOUNCE Codes below for a full list of the valid options for Notice-code.</p>	
OnDemandSessionId	<p>The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.</p> <p>The syntax of the OnDemandSessionId header is as follows. OnDemandSessionId: &lt;onDemandSessionId&gt;where&lt;onDemandSessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemandSessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemandSessionId header follows. OnDemandSessionId: be074250cc5a1d98cd50800200c9a66</p>	Yes
Require	<p>The Require header defines a mechanism by which the client indicates in a request that certain extensions are required to fulfill this request. The syntax of the Require header is as follows.</p> <p>Require: com.comcast.nqod.&lt;interface-id&gt;where&lt;interface-id&gt; is</p>	

Header	Format / Definition	Mandatory
	<p>the NGOD identi#er of the interface, e.g. "s1", "c1", "r2".</p> <p>The speci#c NGOD interfaces that use this feature will de#ne the exact usage. An example of the Require header follows.</p> <p>Require: com.comcast.ngod.r2</p>	
Session	<p>This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.</p> <p>"Session" ":" session-id</p> <p>Session header example Session: 10000105</p>	Yes

**Request Example**

```
ANNOUNCE rtsp://sessionmanager.comcast.com:554/ RTSP/1.0
CSeq: 315
Require: com.comcast.ngod.r2
Session: 47112345
Notice: 5402 "Client Session Terminated" event-date=20160401T023735.13Z npt=342.554
OnDemandSessionId: be074250cc5a11d98cd50800200c9a66
```

**R2 ANNOUNCE Response**

**Response Format**

```
RTSP/1.0 <responseCode> SP <reasonPhrase> CRLF
1 *<header> CRLF
```

**Response Headers**

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	<p>This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.</p> <p>"Session" ":" session-id</p> <p>Session header example Session: 10000105</p>	Yes
OnDemandSessionId	<p>The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.</p> <p>The syntax of the OnDemandSessionId header is as follows. OnDemandSessionId: &lt;onDemandSessionId&gt; where &lt;onDemandSessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemandSessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemandSessionId header follows. OnDemandSessionId: be 074250cc5a1d98cd50800200c9a66</p>	Yes

**Response Example**

```
RTSP/1.0 200 OK
CSeq: 315
```

---

Session: 47112345

OnDemandSessionId: be074250cc5a11d98cd50800200c9a66

---

Announce Request Notice Codes

The following announce codes shall be supported. These are the valid values for the "Notice:" header.

Code	Message	Expected Reaction
2101	End-of-Stream Reached	No NGOD reaction required
2104	Start-of-Stream Reached	No NGOD reaction required
4400	Error Reading Content Data	No NGOD reaction required
5200	Server Resources Unavailable	No NGOD reaction required
5401	Downstream Failure	No NGOD reaction required
5402	Client Session Terminated	No NGOD reaction required
5502	Internal Server Error	No NGOD reaction required
5601	Inband Stream Marker Mismatch	No NGOD reaction required
5601	Bandwidth Exceeded Limit	No NGOD reaction required
5700	Session In Progress	No NGOD reaction required
6000	Encryption Engine Failure	No NGOD reaction required
6001	Stream Bandwidth Exceeds That Available	No NGOD reaction required
6004	Downstream Destination Unreachable	No NGOD reaction required
6005	Unable to Encrypt one or more Components	No NGOD reaction required

Code	Message	Expected Reaction
6006	ECMG Session Failure	No NGOD reaction required

Announce Response Status Codes

The following is the complete set of NGOD RTSP response codes. Specific descriptions or clarifications as to how these codes relate to specific NGOD interfaces may be found in the respective interface specifications.

Code	Message
200	OK
205	Reset Context  This result status code is used in response to a PLAY request where the server is able to play the content, but either the returned Range header, or Scale header has been changed to comply with a restriction on a play list item.  This code is not necessary if the Range header indicates a different value due to the precision necessary for a specific splice location and the Scale header indicates a different value due to the specific track speeds that are supported by the content.
300	Redirect — Multiple Choices
400	Bad Request
401	Unauthorized
403	Forbidden  This status result code is used in response to a PLAY, or PAUSE where the server is unable to deliver the request change in play out of the content due to a restriction on a play list item.  This status result code is used in response to a PLAY, or PAUSE where the server is unable to deliver the request change in play out of the content due to a restriction on a play list item.
404	Not Found
405	Method Not Allowed

Code	Message
406	Not Acceptable
408	Request Time Out
410	Gone
413	Request Entity Too Large
414	Request URI TOO Long
415	Unsupported Media Type
451	Invalid Parameter
453	Not Enough Bandwidth
454	Session Not Found
455	Method Not Valid In This State
456	Header Field Not Valid
457	Invalid Range
458	Parameter Is Read-Only
459	Aggregate Operation Not Allowed
460	Only Aggregate Operation Allowed
461	Unsupported Transport
462	Destination Unreachable
499	No such content

Code	Message
500	Internal Server Error
501	Not implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
505	RTSP Version Not Supported
551	Option Not Supported
599	No available streaming servers
600	Insufficient Encryption Resources
601	IECMG Not Available or Responding
602	Stream/ Service identi#ed for encryption processing indicates it is pre-encrypted content
650	SM Setup Failed - No Response
651	SM Setup Failed — Unknown QAM Group
652	SM Setup Failed — Invalid Request
653	SM Setup Failed — Internal Error
660	PS Setup Failed - No Response
661	PS Setup Failed - Unknown Purchase Token
662	PS Setup Failed — Invalid Request

Code	Message
663	PS Setup Failed — Internal Error
670	ERM Setup Failed - No Response
671	ERM Setup Failed — Invalid Request
672	ERM Setup Failed — QAM Bandwidth Not Available
673	ERM Setup Failed — Network Bandwidth Not Available
674	ERM Setup Failed — Program Not Available
675	ERM Setup Failed — Service Group Not Found
676	ERM Setup Failed — QAM Groups Not Found
677	ERM Setup Failed — QAM Not Available
678	ERM Setup Failed — Edge Device Not Available
679	ERM Setup Failed — Internal Error
690	APM Locate Asset Failed — No Response
691	APM Locate Asset Failed - Asset Not Available
692	APM Locate Asset Failed — Invalid Request
693	APM Locate Asset Failed - Internal Error
750	ODRM Setup Failed — No Response
751	ODRM Setup Failed — Unknown SOP Group
752	ODRM Setup Failed — Bandwidth Not Available



Code	Message
753	ODRM Setup Failed — Stream Not Available
754	ODRM Setup Failed — Asset Not Available
755	ODRM Setup Failed — Invalid Request
756	ODRM Setup Failed — Internal Error
770	Server Setup Failed — No Response
771	Server Setup Failed — Asset Not Found
772	Server Setup Failed — SOP Not Available
773	Server Setup Failed — Unknown SOP Group
774	Server Setup Failed — Unknown SOP Names
775	Server Setup Failed — insufficient Volume Bandwidth
776	Server Setup Failed — insufficient Network Bandwidth
777	Server Setup Failed — invalid Request
778	Server Setup Failed — internal Error

### Announce Request Notice Codes

The following announce codes shall be supported. These are the valid values for the "Notice:" header.

Code	Message	Expected Reaction
2101	End-of-Stream Reached	No NGOD reaction required
2104	Start-of-Stream Reached	No NGOD reaction required

Code	Message	Expected Reaction
4400	Error Reading Content Data	No NGOD reaction required
5200	Server Resources Unavailable	No NGOD reaction required
5401	Downstream Failure	No NGOD reaction required
5402	Client Session Terminated	No NGOD reaction required
5502	Internal Server Error	No NGOD reaction required
5601	Inband Stream Marker Mismatch	No NGOD reaction required
5601	Bandwidth Exceeded Limit	No NGOD reaction required
5700	Session In Progress	No NGOD reaction required
6000	Encryption Engine Failure	No NGOD reaction required
6001	Stream Bandwidth Exceeds That Available	No NGOD reaction required
6004	Downstream Destination Unreachable	No NGOD reaction required
6005	Unable to Encrypt one or more Components	No NGOD reaction required
6006	ECMG Session Failure	No NGOD reaction required

## Announce Response Status Codes

The following is the complete set of NGOD RTSP response codes. Specific descriptions or clarifications as to how these codes relate to specific NGOD interfaces may be found in the respective interface specifications.

Code	Message
200	OK

Code	Message
205	<p>Reset Context</p> <p>This result status code is used in response to a PLAY request where the server is able to play the content, but either the returned Range header, or Scale header has been changed to comply with a restriction on a play list item.</p> <p>This code is not necessary if the Range header indicates a different value due to the precision necessary for a specific splice location and the Scale header indicates a different value due to the specific trick speeds that are supported by the content.</p>
300	Redirect — Multiple Choices
400	Bad Request
401	Unauthorized
403	<p>Forbidden This status result code is used in response to a PLAY, or PAUSE where the server is unable to deliver the request change in play out of the content due to a restriction on a play list item.</p> <p>This status result code is used in response to a PLAY, or PAUSE where the server is unable to deliver the request change in play out of the content due to a restriction on a play list item.</p>
404	Not Found
405	Method Not Allowed
406	Not Acceptable
408	Request Time Out
410	Gone
413	Request Entity Too Large
414	Request URI TOO Long

Code	Message
415	Unsupported Media Type
451	Invalid Parameter
453	Not Enough Bandwidth
454	Session Not Found
455	Method Not Valid In This State
456	Header Field Not Valid
457	Invalid Range
458	Parameter Is Read-Only
459	Aggregate Operation Not Allowed
460	Only Aggregate Operation Allowed
461	Unsupported Transport
462	Destination Unreachable
499	No such content
500	Internal Server Error
501	Not implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout

Code	Message
505	RTSP Version Not Supported
551	Option Not Supported
599	No available streaming servers
600	Insufficient Encryption Resources
601	IECMG Not Available or Responding
602	Stream/ Service identified for encryption processing indicates it is pre-encrypted content
650	SM Setup Failed - No Response
651	SM Setup Failed — Unknown QAM Group
652	SM Setup Failed — Invalid Request
653	SM Setup Failed — Internal Error
660	PS Setup Failed - No Response
661	PS Setup Failed - Unknown Purchase Token
662	PS Setup Failed — Invalid Request
663	PS Setup Failed — Internal Error
670	ERM Setup Failed - No Response
671	ERM Setup Failed — Invalid Request
672	ERM Setup Failed — QAM Bandwidth Not Available
673	ERM Setup Failed — Network Bandwidth Not Available

Code	Message
674	ERM Setup Failed — Program Not Available
675	ERM Setup Failed — Service Group Not Found
676	ERM Setup Failed — QAM Groups Not Found
677	ERM Setup Failed — QAM Not Available
678	ERM Setup Failed — Edge Device Not Available
679	ERM Setup Failed — Internal Error
690	APM Locate Asset Failed — No Response
691	APM Locate Asset Failed - Asset Not Available
692	APM Locate Asset Failed — Invalid Request
693	APM Locate Asset Failed - Internal Error
750	ODRM Setup Failed — No Response
751	ODRM Setup Failed — Unknown SOP Group
752	ODRM Setup Failed — Bandwidth Not Available
753	ODRM Setup Failed — Stream Not Available
754	ODRM Setup Failed — Asset Not Available
755	ODRM Setup Failed — Invalid Request
756	ODRM Setup Failed — Internal Error
770	Server Setup Failed — No Response

Code	Message
771	Server Setup Failed — Asset Not Found
772	Server Setup Failed — SOP Not Available
773	Server Setup Failed — Unknown SOP Group
774	Server Setup Failed — Unknown SOP Names
775	Server Setup Failed — insufficient Volume Bandwidth
776	Server Setup Failed — insufficient Network Bandwidth
777	Server Setup Failed — invalid Request
778	Server Setup Failed — internal Error

### 6.3.9.2 R2 GET\_PARAMETER

#### R2 GET\_PARAMETER Request

The SRM interacts with the Streaming Server via RTSP Get\_Parameter request and response to retrieve information about sessions.

#### Request Format

Get\_Parameter is a regular RTSP request from SRM to Video Server, with no body. In (quasi-) BNF the format of this request shall be

```
GET_PARAMETER SP <uri> SP RTSP/ 1.0 CRLF
1 *<header> CRLF
CRLF
<Get-Parameter-Extension>
```

#### Request URI

The request <uri> shall have the form:

---

`rtsp: //<StreamingServerAddress>:<StreamingServerPort>/`

---

Field	Description
<code>&lt;StreamingServerAddress&gt;</code>	The DNS name or IP address of the Streaming Server.
<code>&lt;StreamingServerPort&gt;</code>	The TCP port for the Streaming Server.

---

### RTSP Request Headers

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / Definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Require	<p>The Require header defines a mechanism by which the client indicates in a request that certain extensions are required to fulfill this request. The syntax of the Require header is as follows.</p> <p>Require: com.comcast.ngod.&lt;interface-id&gt; where &lt;interface-id&gt; is the NGOD identifier of the interface, e.g. "s1", "c1", "r2".</p> <p>The specific NGOD interfaces that use this feature will define the exact usage. An example of the Require header follows. Require: com.comcast.ngod.r2</p>	
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.	Yes



Header	Format / Definition	Mandatory?
	"Session" ":" session-id Session header example Session: 10000105	
Content-type	The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient Content-Type = "Content-Type" ":" media-type An example is Content-Type: text/ parameters	Yes
Content-length	The Content-Length entity-header field indicates the size of the entity-body, in decimal number of OCTETs, sent to the recipient. Content-Length = "Content-Length" ":" 1*DIGIT An example is Content-Length: 3495	Yes

**Request Example**

```
GET_PARAMETER rtsp://streamingserver34.nagra.com:554/ RTSP/1.0
CSeq: 315
Require: com.comcast.ngod.r2
Session: 47112345
Content-type: text-parametersContent-Length: 19
session_list
```

**GET\_PARAMETER Extensions**

This section describes extensions and clarifications to the valid parameter options and responses for the GET\_PARAMETER method.

**Parameter Options**

Parameter Type	Description
connection_timeout	Timeout setting for activity on a connection.

Parameter Type	Description
	<p>The RTSP server shall support a parameter of connection_timeout. The return value from a get_parameter of connection_timeout shall be as follows</p> <pre>connection_timeout: &lt;timeout&gt;</pre> <p>where</p> <p>&lt;timeout&gt; is an integer representing seconds.</p> <p>for example: connection_timeout: 300</p>
session_list	<p>List of current active sessions.</p> <p>The RTSP server shall support a parameter of session_list. As an optional message header to the GET_PARAMETER session_list request, the RTSP server shall support the SessionGroup header. The return value conveys a list of sessions' IDs for sessions that are active and were setup with the SessionGroup value. If the SessionGroup header is omitted, all sessions will be returned.</p> <p>The syntax for the return value for a GET_PARAMETER session_list is as follows. session_list: [&lt;rtsp-session-id&gt;:&lt;on-demand-session-id&gt;] [&lt;rtsp-session-id&gt;:&lt;on-demand-session-id&gt;]* where</p> <p>&lt;rtsp-session-id&gt; is the RTSP server's session ID &lt;on-demand-session-id&gt; is the OnDemandSessionId generated by the Session Manager.</p> <p>An example return value for a GET_PARAMETER session_list follows</p> <pre>session list: 12345:b50557b0fecc11d98cd60800200c9a66 12346:dec1b300fecc11d98cd60800200c9a66 12347:0257ce80fecc11d98cd60800200c9a66 12348:18315fa0fecc11d98cd60800200c9a66 12349:2ee54b30fecc11d98cd60800200c9a66 12360:3ddb3ff0fecc11d98cd608008200c9a66</pre>
position	<p>The current stream position. The NPT values are in seconds, consistent with the ntp-sec as defined in RFC 2326bis07</p>
presentation_state	<p>Current state of the stream</p> <p>The list of possible return values for presentation_state is as follows.</p> <pre>init ready play pause</pre> <p>Fast forward will be represented with a presentation_state of "play" and a scale greater than 1.0. Rewind will be represented with a presentation_state of "play" and a negative scale value</p>
scale	<p>The current play scale. E.g. 1.0 7.0, -7.0</p>

## R2 GET\_PARAMETER ResponseResponse Format

---

```
RTSP/1.0 <responseCode> SP <reasonPhrase> CRLF
1 *<header> CRLF
CRLF
<Get-Parameter-Extension>
```

---

**NOTE: Response Headers and Extensions are same as Get\_Parameter request.**

### Response Example

---

```
RTSP/1.0 200 OK
CSeq: 36393
Session: 1231796058
Content-Type: text/parameters
Content-Length: 23
presentation_state: play
```

---

## R2 GET\_PARAMETER Response

### Response Format

---

```
RTSP/1.0 <responseCode> SP <reasonPhrase> CRLF
1 *<header> CRLF
CRLF
<Get-Parameter-Extension>
```

---

**NOTE: Response Headers and Extensions are same as Get\_Parameter request.**

### Response Example

---

```
RTSP/1.0 200 OK
CSeq: 36393
Session: 1231796058
Content-Type: text/parameters
Content-Length: 23
presentation_state: play
```

---

## 6.3.9.3 R2 SET\_PARAMETER

## R2 SET\_PARAMETER Request

The SRM interacts with the Streaming Server via RTSP Set\_Parameter request and response messages to provide the Streaming Server with a list of sessions being managed by a single SRM connection.

### Request Format

Set\_Parameter is a regular RTSP request from SRM to Video Server, with no body. In (quasi-) BNF the format of this request shall be

```
GET_PARAMETER SP <uri> SP RTSP/1.0 CRLF
1 *<header> CRLF
CRLF
<rtsp-parameter>
```

### Request URI

The request <uri> shall have the form

```
rtsp: //<StreamingServerAddress>:<StreamingServerPort>/
```

Field	Description
<StreamingServerAddress>	The DNS name or IP address of the Streaming Server.
<StreamingServerPort>	The TCP port for the Streaming Server.

### RTSP Request Headers

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / Definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Require	The Require header defines a mechanism by which the client indicates in a request that certain extensions are required to fulfill this	

Header	Format / Definition	Mandatory?
	<p>request. The syntax of the Require header is as follows.</p> <p>Require: com.comcast.ngod.&lt;interface-id&gt;where&lt;interface-id&gt; is the NGOD identifier of the interface, e.g. "s1", "c1", "r2".</p> <p>The specific NGOD interfaces that use this feature will define the exact usage. An example of the Require header follows. Require: com.comcast.ngod.r2</p>	
Content-type	<p>The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient</p> <p>Content-Type = "Content-Type" ":" media-type</p> <p>An example is</p> <p>Content-Type: text/parameters</p>	Yes
Content-length	<p>The Content-Length entity-header field indicates the size of the entity-body, in decimal number of OCTETs, sent to the recipient.</p> <p>Content-Length = "Content-Length" ":" 1*DIGIT</p> <p>An example is</p> <p>Content-Length: 3495</p>	Yes

**RTSP Parameters**

When using the SET\_PARAMETER method in R2 the following parameters must be supported.

Parameter	Description
session_list	<p>Normally the connection between the RTSP client and server is persistent and ANNOUNCE methods can be sent across it.</p> <p>However, if the connection is broken due to an unexpected failure, there needs to be a way for an RTSP client to indicate which connection will be used for session communications. An RTSP client MAY send a list of sessions using the parameter session_list. This is a signal to the RTSP server that these sessions will be controlled via the connection over which the SET_PARAMETER method was received. Should an ANNOUNCE method be required any session in the</p>

Parameter	Description
	<p>session list, the RTSP server will use connection over which the session_list was received for sending the ANNOUNCE method.</p> <p>The syntax for session_list is as follows. session_list: [&lt;rtsp-session-id&gt;:&lt;on-demand-session-id&gt;] [&lt;rtsp-session-id&gt;:&lt;on-demand-session-id&gt;]*where &lt;rtsp-session-id&gt; is the RTSP server's session ID</p> <p>Example:</p> <pre>SET_PARAMETER rtsp://videoserver234.comcast.com:554 RTSP/1.0   CSeq: 2   Require: com.comcast.ngod.c1   Content-Type: text/parameters   Content-Length: 52   session_list: 12345:b50557b0fecc11d98cd60800200c9a66</pre>
session_groups	<p>Normally the connection between the RTSP client and server is persistent and ANNOUNCE methods can be sent across it.</p> <p>However, if the connection is broken due to an unexpected failure, there needs to be a way for the RTSP server to indicate which connection will be used for session communications. An RTSP client MAY send a list of session groups using the parameter session_groups. This is a signal to the RTSP server that any session within these session groups will be controlled via the connection over which the SET_PARAMETER method was received. Should an ANNOUNCE method be required for any session within the list of session groups, the RTSP server will use the connection over which the session_groups was received for sending the ANNOUNCE method.</p> <p>The format is as follows:</p> <pre>session_groups: &lt;session_group&gt;[ &lt;session_group&gt;]*</pre> <p>Example:</p> <pre>SET_PARAMETER rtsp://172.168.2.2/1234 RTSP/1.0   CSeq: 314   Require: com.comcast.ngod.r2   Content-Type: text/parameters   Content-Length: 40   session_groups: SM1.SG1 SM1.SG2 SM1.SG3</pre>

## R2 Set\_Parameter Request Example

### Session Groups Example

```
SET_PARAMETER rtsp://streamingserver32.comcast.com:554 RTSP/1.0 CSeq: 36394 Require: com.comcast.ngod.r2 Content-Type: text/parameters Content-Length: 40
```

```
session_groups: SM1.SG1 SM1.SG2 SMI.SG3
```

### Session List Example

---

SET\_PARAMETER rtsp://streamingserver32.comcast.com:554 RTSP/1.0 CSeq: 36395 Require: com.comcast.  
ngod.r2 Content-Type: text/parameters Content-Length: 52  
Session\_list: 12345:b50557b0fecc1d98cd60800200c9a66

---

R2 SET\_PARAMETER Response

### Response Format

---

RTSP/ 1.0 <responseCode> SP <reasonPhrase> CRLF  
1 \*<header> CRLF

---

### RTSP Request Headers

The R2 Set\_Parameter has only one header ( CSeq ) in response.

---

Header	Format / Definition
CSeq	64-bit decimal integer. An exact copy of the value in the request.

---

### Response Example

---

RTSP/1 .0 200 OK  
CSeq: 36393

---

## R2 SET\_PARAMETER Response

### Response Format

---

RTSP/ 1.0 <responseCode> SP <reasonPhrase> CRLF  
1 \*<header> CRLF

---

### RTSP Request Headers

The R2 Set\_Parameter has only one header ( CSeq ) in response.

Header	Format / Definition
CSeq	64-bit decimal integer. An exact copy of the value in the request.

**Response Example**

```
RTSP/1.0 200 OK
CSeq: 36393
```

**6.3.9.4 R2 SETUP**

**R2 SETUP Request**

The SRM interacts with Streaming Server with RTSP Setup request and response messages to establish new session

**Request Format**

To establish a session, the STB shall send an RTSP Setup request to the SRM. This is a regular RTSP request. In (quasi-) BNF the format of this request shall be

```
SETUP SP <uri> SP RTSP/1.0 CRLF
1 *<header> CRLF
<body>
```

**Request URI**

The request <uri> shall have the form:

```
rtsp://<StreamingServerPath>:<StreamingServerPort>;purchaseToken=<purchase-token>;serverId=<server-id> RTSP/1.0
```

Please note that all of the variables in URI are mandatory for a valid RTSP URL for an NGOD R2 SETUP request.

Field	Description
<StreamingServerPath>	The DNS name or IP address which the SRM uses to contact the Streaming Server.



Field	Description
<StreamingServerPort>	The TCP port which the SRM sends RTSP request to Streaming Server

**RTSP Request Headers**

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / Definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Require	<p>The Require header defines a mechanism by which the client indicates in a request that certain extensions are required to fulfill this request. The syntax of the Require header is as follows.</p> <p>Require: com.comcast.ngod.&lt;interface-id&gt; where &lt;interface-id&gt; is the NGOD identifier of the interface, e.g. "s1", "c1", "r2".</p> <p>The specific NGOD interfaces that use this feature will define the exact usage. An example of the Require header follows. Require: com.comcast.ngod.r2</p>	
OnDemandSessionId	<p>The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.</p> <p>The syntax of the OnDemandSessionId header is as follows. OnDemandSessionId: &lt;onDemandSessionId&gt; where &lt;onDemandSessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemandSessionId must follow the format specified in RFC 1422 except that</p>	Yes

Header	Format / Definition	Mandatory?						
	<p>there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemand SessionId header follows. OnDemandSessionId: be O74250cc5a1d98cd50800200c9a66</p>							
Volume	<p>The Volume header provides a mechanism in the SETUP request to specify the name of the volume on which the requested asset(s) reside. Only a single value can be included in this header. The syntax of the Volume header is as follows. Volume: &lt;name&gt;where &lt;name&gt; is a string that specifies the NGOD name of the volume. An example of the Volume header follows. Volume: Boston.v1</p>	Yes						
Transport	<pre>MP2T/DVBC/UDP; unicast; client=&lt;client-id&gt;; destination=&lt;destination&gt;; client_port=&lt;client-port&gt;; sop_name=&lt;sop-name&gt;; sop_group=&lt;sop-group&gt;; bandwidth=&lt;bandwidth&gt;</pre>	No						
<table border="1"> <thead> <tr> <th>Transport param</th> <th>Format / Description</th> <th>Mandatory?</th> </tr> </thead> <tbody> <tr> <td>&lt;client-id&gt;</td> <td> <p>&lt;client-id&gt; is a unique identifier of the client. Note that although this information is not needed to establish the</p> </td> <td></td> </tr> </tbody> </table>			Transport param	Format / Description	Mandatory?	<client-id>	<p>&lt;client-id&gt; is a unique identifier of the client. Note that although this information is not needed to establish the</p>	
Transport param	Format / Description	Mandatory?						
<client-id>	<p>&lt;client-id&gt; is a unique identifier of the client. Note that although this information is not needed to establish the</p>							

Header	Format / Definition	Mandatory?
<p><b>Transport param</b></p>	<p><b>Format / Description</b></p>	<p><b>Mandatory?</b></p>
	<p>session, the information is still required as per the NGOD specifications for the operational use of the solution.</p> <p>E.g. tracking behavior across the system.</p> <p>&lt;destination&gt; describes the IP/host address to which data is streamed.</p> <p>E.g. IP of Edge Device.</p>	
	<p>&lt;client-port&gt;</p> <p>&lt;client-port&gt; describes the port to</p>	

Header	Format / Definition	Mandatory?
	<p>which the data is streamed.</p> <p>E.g. a input port for a Edge Device</p>	
<sop-name>	<p>&lt;sop-name&gt; is the server output port (SOP) name from which the data is streamed.</p>	
<sop-group>	<p>&lt;sop-group&gt; is the SOP Group from which the data is streamed.</p>	
bandwidth	<p>&lt;bandwidth&gt; is the bandwidth requirement in bits</p>	

Header	Format / Definition	Mandatory?
SessionGroup	<p data-bbox="606 481 981 548"><b>Transport Format / Mandatory?</b> <b>param Description</b></p> <hr/> <p data-bbox="734 571 821 638">per second.</p> <hr/> <p data-bbox="590 750 997 907">The SessionGroup header defines a token passed on a SETUP request that is used to identify a group of sessions. The syntax of the Session Group header is as follows i.g. SessionGroup: SM1</p> <p data-bbox="590 918 997 1108">The SessionGroup header is used in the process of re-synchronizing session state between two NGOD components. The SessionGroup token is generated by an RTSP client component to group its established sessions together.</p> <p data-bbox="590 1120 997 1489">A SessionGroup token is passed to a RTSP server within SETUP request messages. RTSP servers remember the SessionGroup token for each established session. RTSP clients may send a GET_PARAMETER request for the "session_list" parameter, passing the SessionGroup token. The RTSP server returns the list of session IDs for currently active sessions that have that SessionGroup token. Note that a session may only belong to a single Session Group.</p> <p data-bbox="590 1500 997 1814">For example, a given Session Manager (#1) may choose to represent its sessions with the SessionGroup token "SM1". This value is configured to be unique. Within each SETUP request this Session Manager sends to an ODRM it will pass "SessionGroup: SM1". When this Session Manager wants to synchronize sessions with the ODRM it will send a GET_PARAMETER of session_list,</p>	No

Header	Format / Definition	Mandatory?
StartPoint	<p>passing "SessionGroup: SM1". The ODRM RTSP server will return the list of sessions with a SessionGroup of SM1.</p> <p>Note that there may be a second Session Manager establishing sessions on the same ODRM, but it could be configured to pass "SessionGroup: SM2". In this manner each Session Manager may synchronize with the ODRM for solely the sessions that it established.</p> <p>The StartPoint header specifies the point in the asset list at which the Streaming Server should begin streaming.</p> <p>The syntax of the StartPoint is given by the following ABNF: start-point = "StartPoint" HCOLON segment-npt-specifier CRLF segment-npt-specifier = slot SP npt-sec</p> <p>slot = 1*DIGIT ; Index (1-based) of asset list element An example of the StartPoint header follows. StartPoint: 2 1234.78</p>	Yes
InbandMarker	<p>The InbandMarker header defines a mechanism by which Resource Managers can request that specific data be injected into the In-Band stream by an Edge Device. For example, this would allow the Edge Device to be instructed to install CA Descriptors to support pre-encryption.</p> <p>In-Band data to be injected can be specified as an MPEG descriptor, an MPEG Table, or as an MPEG packet. This data can be injected into the In-Band stream in a number of ways. The following table defines the specific types of data which can be injected, along with the rules for how they will be injected.</p>	No

Header	Format / Definition	Mandatory?
Content-type	Content type of body	Yes
Content-length	Size of request body in bytes	Yes

**R2 SETUP Request Body**

The NGOD implementation will utilize the “a=X-playlist-item” type to describe playlist elements to be played out within a session. Note that the “a=X-playlist-item” lines are in the order of desired playlist.

The syntax of the playlist item definition is as follows:-

a=X-playlist-item: <provider-id> <asset-id> [ <range> ][ tricks/[F][R][P] 1

Variable	Description
a=	“a=” is the syntax for a media attribute. For more information on this syntax please see RFC 2327.
<provider-id>	<provider-id> is the CableLabs provider_id for the content asset
<asset-id>	<asset-id> is the CableLabs asset_id for the content asset
<range>	<range> is the playout range in the form: [<start-ntp>]-<stop-ntp>]
tricks/ [F][R] [P]	tricks/ [F][R] [P] is the trick mode #ags that disables that mode. F is Fast Forward, R is Rewind, P is Pause. The absence of #ags should mean that every trick mode is acceptable

The NPT values are in seconds, consistent with the ntp-sec as defined in [RFC2326bis07]

An example of the X-playlist-item type follows.

a=X-playlist-item: comcast.com abcd1234567890123456 1.0-200

The following example shows multiple playlist elements:

a=X-playlist-item: comcast.com advert12345678901234 0.0-30.

a=X-playlist-item: comcast.com advert98765432100976 0.0-30.

a=X-playlist-item: comcast.com main1234567890123456 0.0-600

Field Example	Description
v=0	This is the protocol version number for this NGOD release, this value is 0.

Field Example	Description
o=- 1234 2890842807 IN IP4 10.47.16.5	The #elds: (1) Email address. For this release this will be "-".(2) Session identi#er. In requests, it is the OnDemandSessionId ID.(3) Session description version. It should be a Network Time Protocol (NTP) format timestamp. This should be the time that the session setup message was created.(4) Indicates address format. "IN" indicates internet format.(5) Indicates address format. "IP4" indicates internet protocol version 4.(6) IP address creating the session. In requests, it's the address of the server initiating the request (SM or ODRM).
s=	This is a text string description of the session. For this NGOD release this value is "".
t=0 0	Describes the validity start/end times of the session. 0 indicates media is always available.
a=X-playlist-item: comcast.com abcd1234567890123456 1000-6000	See playlist-item description above.
c=IN IP4 0.0.0.0	Describes the media stream. This can be considered a #xed string for this NGOD release.
m=video 0 udp MP2T	This can be considered a #xed string for this NGOD release.

**R2 SETUP Request Example**

```

SETUP rtsp://srm.nagra.com:554 RTSP/1.0
CSeq: 896Require: com.comcast.ngod.r2OnDemandSessionId:
be074250cc5a11d98cd50800200c9a66Volume: boston.v1Transport: MP2T/DVBC/
UDP;unicast;client=00AF123456DE;bandwidth=2920263;destination=2.2.2.2;client_port=45;sop_name=Boston.
Pump1.2,
MP2T/DVBC/UDP;unicast:
client=00AF123456DE;bandwidth=2920263;destination=2.2.2.2;client_port=45;sop_
group=Boston.PGA2SessionGroup: SM1StartPoint: 1 3.0InbandMarker:type=4;pidType=A;pid
Value=01EE;dataType=T;insertDuration=10000:data=4002003030mContent-type: application/
sdpContent-length: 184
v=0o=- be074250-cc5a-11d9-8cd5-0800200c9a662890842807 IN IP4 10.47.16.5s:
t=0 0
a=X-playlist-item: comcast.com abcd1234567890123456 0.0-60.0
c=IN IP4 0.0.0.0

```



---

m=video 0 udp MP2T

---

## R2 SETUP Response

### Response Format

Similar to request response consist of response code & phrase with a set of headers and body. This is a regular RTSP response with a body. In (quasi-) BNF the format of this response shall be:-

---

```
RTSP/ 1.0 SP <code> SP <phrase> SP CRLF
1 *<header> CRLF
CRLF
<body>
```

---

### RTSP Response Headers

If the request succeeds, the required / permitted headers included in the response are as defined in the below table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / Definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	<p>This response header field identifies an RTSP session started by the media server. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session. A server does not have to set up a session identifier if it has other means of identifying a session, such as dynamically generated URLs.</p> <p>"Session" ":" session-id [ ";" "timeout" "=" delta-seconds ]</p> <p>Session header example Session: 10000105;timeout=60</p> <p>The timeout parameter is used by the server to indicate to the client how long the server is prepared to wait between RTSP commands before closing the session due to lack of activity. The timeout is</p>	

Header	Format / Definition	Mandatory?				
	measured in seconds, with a default of 60 seconds.					
OnDemandSessionId	<p>The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.</p> <p>The syntax of the OnDemandSessionId header is as follows. On DemandSessionId: &lt;onDemandSessionId&gt;where&lt;onDemandSessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemandSessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemandSessionId header follows. OnDemandSessionId: be 074250cc5a1d98cd50800200c9a66</p>					
Transport	<pre>MP2T/DVBC/UDP; unicast; client=&lt;client-id&gt;; bandwidth=&lt;bandwidth&gt;; destination=&lt;destination&gt;; client_port=&lt;client-port&gt;; source=&lt;source&gt;; server_port=&lt;server-port&gt;; sop_name=&lt;sop-name&gt;; sop_group=&lt;sop-group&gt;;</pre>	No				
	<table border="1"> <thead> <tr> <th>Transport param</th> <th>Format / Description</th> </tr> </thead> <tbody> <tr> <td>&lt;client-id&gt;</td> <td>&lt;client-id&gt; is a unique identifier of the client. Note that</td> </tr> </tbody> </table>	Transport param	Format / Description	<client-id>	<client-id> is a unique identifier of the client. Note that	
Transport param	Format / Description					
<client-id>	<client-id> is a unique identifier of the client. Note that					

Header	Format / Definition		Mandatory?
	Transport param	Format / Description	
		<p>although this information is not needed to establish the session, the information is still required as per the NGOD specifications for the operational use of the solution.</p> <p>E.g. tracking behaviour across the system.</p>	
	<bandwidth>	<p>&lt;bandwidth&gt; is the bandwidth requirement in bits per second.</p>	
	<destination>	<p>&lt;destination&gt; describes the IP/host address to which data is streamed.</p> <p>E.g. IP of Edge Device.</p>	
	<client-port>	<p>&lt;client-port&gt; describes the port to which</p>	

Header	Format / Definition		Mandatory?
	<b>Transport param</b>	<b>Format / Description</b>	
		<p>the data is streamed.</p> <p>E.g. a input port for a Edge Device</p>	
	<source>	<p>&lt;source&gt; is the IP/ host address from which the data is streamed. E.g. the IP address of the streaming server</p>	
	<server-port>	<p>&lt;server-port&gt; is the port from which the data is streamed. E.g. the output port of a streaming server</p>	
	<sop-name>	<p>&lt;sop-name&gt; is the server output port (SOP) name from which the data is streamed.</p>	
	<sop-group>	<p>&lt;sop-group&gt; is the SOP Group from which the</p>	

Header	Format / Definition	Mandatory?				
	<table border="1"> <thead> <tr> <th>Transport param</th> <th>Format / Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>data is streamed.</td> </tr> </tbody> </table>	Transport param	Format / Description		data is streamed.	
Transport param	Format / Description					
	data is streamed.					
Content-type	<p>The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient</p> <p>Content-Type = "Content-Type" ":" media-type</p> <p>An example is</p> <p>Content-Type: text/parameters</p>					
Content-length	<p>The Content-Length entity-header field indicates the size of the entity-body, in bytes, sent to the recipient.</p> <p>Content-Length = "Content-Length" ":" 1*DIGIT</p> <p>An example is</p> <p>Content-Length: 3495</p>					

**R2 SETUP response Body**

SDP Text Example	Description
v=0	This is the SDP protocol version number. For this NGOD release, this value is 0.
o=- 777 2890842817 IN IP4 1.2.3.4	The #elds:(1) Email address. For this release this will be "-".(2) Session identi#er. The RTSP session ID of this session on the Streaming Server. (Note on S1: The On Demand Client will use this session ID when sending RTSP stream control commands over C1)(3) Session description version. It should Network Time Protocol (NTP) format timestamp. This should be the time that the session setup message was created.(4) Indicates address format. "IN" indicates internet format.(5) Indicates address format. "IP4" indicates internet protocol version 4.(6) IP address of the RTSP server on the Streaming Server.
s=	This is a text string description of the session. For this NGOD release this value is "".

**SDP Text Example**

**Description**

t=0 0

Describes the validity start/end times of the session. 0 indicates media is always available.

a=control:lscp://  
videosever234.comcast.com:  
554/9 876

The syntax of the control location is as follows.

a=control:<protocol>://<host>:<port>/<streamhandle>

**Variable**

**Description**

a=

“a=” is the syntax for a media attribute. For more information on this syntax please see RFC 2327.

<protocol>

<protocol> indicates the stream control protocol selected e.g. rtsp, lscp, lscpu.

<host>

<host> is the IP address or fully qualified DNS name of the stream control port.

<port>

<port> is the TCP or UDP port number of the stream control port. Available TCP and UDP choices may use identical port numbers. <streamhandle> is the RTSP session ID or LSCP stream handle to control this stream, depending on the <protocol> value.

The stream handle value is represented as decimal. Available RTSP and LSCP choices may use identical values.

c=IN IP4 0.0.0.0

Describes the media stream. This can be considered a #xed string for this NGOD release.

m=video 0 udp  
MP2T

This can be considered a #xed string for this NGOD release.

**R2 SETUP Response Example**

RTSP/1.0 200 OKCSeq: 896Session: 777OnDemandSessionId: be074250cc5a1d98cd50800200c9a66Transport: MP2T/DVBC/UDP:unicast;client=00AF123456DE;bandwidth=2920263;destination=2.2.2.2:client\_port=45;source=l.2.3.4;server\_port=123;sop\_name=Boston.Pump1.2;sop group=Boston.PGA1Content-type: application/sdpContent-length: 138

---

```
v=0o=- 777 2890842817 IN IP4 1.2.3.4s=t=0 0a=control:iscp://videosever4.nagra.com 554/9876c=IN IP4
2.2.2.2m=video 45 udp MP2T
```

---

## R2 SETUP Response

### Response Format

Similar to request response consist of response code & phrase with a set of headers and body. This is a regular RTSP response with a body. In (quasi-) BNF the format of this response shall be:-

---

```
RTSP/ 1.0 SP <code> SP <phrase> SP CRLF
1 *<header> CRLF
CRLF
<body>
```

---

### RTSP Response Headers

If the request succeeds, the required / permitted headers included in the response are as defined in the below table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / Definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	<p>This response header field identifies an RTSP session started by the media server. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session. A server does not have to set up a session identifier if it has other means of identifying a session, such as dynamically generated URLs.</p> <pre>"Session" ":" session-id [ ";" "timeout" "=" delta-seconds ]</pre> <p>Session header example Session: 10000105;timeout=60</p> <p>The timeout parameter is used by the server to indicate to the client how long the server is prepared to wait between RTSP commands</p>	

Header	Format / Definition	Mandatory?				
	before closing the session due to lack of activity. The timeout is measured in seconds, with a default of 60 seconds.					
OnDemandSessionId	<p>The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.</p> <p>The syntax of the OnDemandSessionId header is as follows. OnDemandSessionId: &lt;onDemandSessionId&gt; where &lt;onDemandSessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemandSessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemandSessionId header follows. OnDemandSessionId: be074250cc5a11d98cd50800200c9a66</p>					
Transport	<pre>MP2T/DVBC/UDP; unicast; client=&lt;client-id&gt;; bandwidth=&lt;bandwidth&gt;; destination=&lt;destination&gt;; client_port=&lt;client-port&gt;; source=&lt;source&gt;; server_port=&lt;server-port&gt;; sop_name=&lt;sop-name&gt;; sop_group=&lt;sop-group&gt;;</pre>	No				
	<table border="1"> <thead> <tr> <th>Transport param</th> <th>Format / Description</th> </tr> </thead> <tbody> <tr> <td>&lt;client-id&gt;</td> <td>&lt;client-id&gt; is a unique identifier of</td> </tr> </tbody> </table>	Transport param	Format / Description	<client-id>	<client-id> is a unique identifier of	
Transport param	Format / Description					
<client-id>	<client-id> is a unique identifier of					



Header	Format / Definition		Mandatory?
	Transport param	Format / Description	
		<p>the client. Note that although this information is not needed to establish the session, the information is still required as per the NGOD specifications for the operational use of the solution.</p>	
	<bandwidth>	<p>E.g. tracking behaviour across the system.</p> <p>&lt;bandwidth&gt; is the bandwidth requirement in bits per second.</p>	
	<destination>	<p>&lt;destination&gt; describes the IP/host address to which data is streamed.</p> <p>E.g. IP of Edge Device.</p>	
	<client-port>	<p>&lt;client-port&gt; describes the port</p>	

Header	Format / Definition		Mandatory?
	<b>Transport param</b>	<b>Format / Description</b>	
		<p>to which the data is streamed.</p> <p>E.g. a input port for a Edge Device</p>	
	<source>	<p>&lt;source&gt; is the IP/ host address from which the data is streamed. E.g. the IP address of the streaming server</p>	
	<server-port>	<p>&lt;server-port&gt; is the port from which the data is streamed. E.g. the output port of a streaming server</p>	
	<sop-name>	<p>&lt;sop-name&gt; is the server output port (SOP) name from which the data is streamed.</p>	
	<sop-group>	<p>&lt;sop-group&gt; is the SOP Group from which the</p>	

Header	Format / Definition	Mandatory?				
	<table border="1"> <thead> <tr> <th>Transport param</th> <th>Format / Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>data is streamed.</td> </tr> </tbody> </table>	Transport param	Format / Description		data is streamed.	
Transport param	Format / Description					
	data is streamed.					
Content-type	<p>The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient</p> <p>Content-Type = "Content-Type" ":" media-type</p> <p>An example is</p> <p>Content-Type: text/parameters</p>					
Content-length	<p>The Content-Length entity-header field indicates the size of the entity-body, in bytes, sent to the recipient.</p> <p>Content-Length = "Content-Length" ":" 1*DIGIT</p> <p>An example is</p> <p>Content-Length: 3495</p>					

**R2 SETUP response Body**

SDP Text Example	Description
v=0	This is the SDP protocol version number. For this NGOD release, this value is 0.
o=- 777 2890842817 IN IP4 1.2.3.4	The #elds:(1) Email address. For this release this will be "-".(2) Session identi#er. The RTSP session ID of this session on the Streaming Server. (Note on S1: The On Demand Client will use this session ID when sending RTSP stream control commands over C1)(3) Session description version. It should Network Time Protocol (NTP) format timestamp. This should be the time that the session setup message was created.(4) Indicates address format. "IN" indicates internet format.(5) Indicates address format. "IP4" indicates internet protocol version 4.(6) IP address of the RTSP server on the Streaming Server.
s=	This is a text string description of the session. For this NGOD release this value is "".

**SDP Text Example**

**Description**

t=0 0

Describes the validity start/end times of the session. 0 indicates media is always available.

a=control:lscp://  
videoserver234.comcast.com:  
554/9 876

The syntax of the control location is as follows.

a=control:<protocol>://<host>:<port>/<streamhandle>

**Variable**

**Description**

a=

“a=” is the syntax for a media attribute. For more information on this syntax please see RFC 2327.

<protocol>

<protocol> indicates the stream control protocol selected e.g. rtsp, lscp, lscpu.

<host>

<host> is the IP address or fully qualified DNS name of the stream control port.

<port>

<port> is the TCP or UDP port number of the stream control port. Available TCP and UDP choices may use identical port numbers. <streamhandle> is the RTSP session ID or LSCP stream handle to control this stream, depending on the <protocol> value.

The stream handle value is represented as decimal. Available RTSP and LSCP choices may use identical values.

c=IN IP4 0.0.0.0

Describes the media stream. This can be considered a #xed string for this NGOD release.

m=video 0 udp  
MP2T

This can be considered a #xed string for this NGOD release.

**R2 SETUP Response Example**

RTSP/1.0 200 OKCSeq: 896Session: 777OnDemandSessionId: be074250cc5a1d98cd50800200c9a66Transport: MP2T/DVBC/UDP:unicast;client=00AF123456DE;bandwidth=2920263;destination=2.2.2.2:client\_port=45;source=l.2.3.4;server\_port=123;sop\_name=Boston.Pump1.2;sop group=Boston.PGA1Content-type: application/sdpContent-length: 138

---

```
v=0o=- 777 2890842817 IN IP4 1.2.3.4s=t=0 0a=control:iscp://videosever4.nagra.com 554/9876c=IN IP4
2.2.2.2m=video 45 udp MP2T
```

---

### 6.3.9.5 R2 TEARDOWN

#### R2 TEARDOWN Request

The SRM interacts with the Streaming Server with RTSP Teardown Request and Response messages to teardown existing sessions.

##### Request Format

To teardown a session, the SRM shall send an RTSP Teardown request to the Streaming Server. This is a regular RTSP request, with no body. In (quasi-) BNF the format of this request shall be

---

```
TEARDOWN SP <uri> SP RTSP/1.0 CRLF
1 *<header> CRLF
```

---

##### Request URI

The request <uri> shall have the form:

---

```
rtsp://<logicalStreamingServerAddress>:<logicalStreamingServerPort>/
```

---

Field	Description
<logicalStreamingServerAddress>	The DNS name or IP address of the Streaming Server.
<logicalStreamingServerPort>	The TCP port for Streaming Server.

---

##### RTSP Request Headers

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / Description	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Require	<p>The Require header defines a mechanism by which the client indicates in a request that certain extensions are required to fulfill this request. The syntax of the Require header is as follows.</p> <p>Require: com.comcast.ngod. &lt;interface-id&gt;where&lt;interface-id&gt; is the NGOD identifier of the interface, e.g. "s1", "c1", "r2".</p> <p>The specific NGOD interfaces that use this feature will define the exact usage. An example of the Require header follows. Require: com.comcast.ngod.s1</p>	Yes
Reason	<p>The Reason header defines the reason for a particular TEARDOWN message.</p> <p>The syntax for the Reason header is as follows. Reason: &lt;reason-code&gt; "&lt;reason-text&gt;" where 1. &lt;reason-code&gt; is an integer 2. &lt;reason-text&gt; is a textual description of the reason</p> <p>Please see Section RTSP TEARDOWN Reason Codes at the bottom of this page for a full list of the valid reason codes.</p> <p>An example of the Reason header follows. Reason: 200 "user pressed stop"</p>	Yes
Session	<p>This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.</p>	Yes

Header	Format / Description	Mandatory?
	"Session" ":" session-id Session header example Session: 10000105	
OnDemandSessionId	<p>The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.</p> <p>The syntax of the OnDemandSessionId header is as follows. OnDemandSessionId: &lt;onDemandSessionId&gt;where&lt;onDemandSessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemandSessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemandSessionId header follows. OnDemandSessionId: be074250cc5a11d98cd50800200c9a66</p>	Yes

**Request Example**

```
TEARDOWN rtsp://sessionmanager2.hot.com/ RTSP/1.0
CSeq: 335
Require: com.comcast.ngod.r2
Reason: 200 "User pressed stop"
Session: 6342345
OnDemandSessionId: be074250cc5a11d98cd50800200c9a66
```

R2 TEARDOWN Response

**Response Format**

In (quasi-)BNF the format of the response to this request shall be:

---

```
RTSP/ 1.0 <responseCode> SP <reasonPhrase> CRLF
1 *<header> CRLF
CRLF
<body>
```

---

### Response Headers

Header	Format / Description	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.  "Session" ":" session-id  Session header example Session: 10000105	Yes
FinalNPT	The FinalNPT header defines a means for an RTSP server to return the NPT of the stream as known at the time the session is torn down. This will return a value equivalent to that returned in response to A GET_PARAMETER of "position" without the need to use that method.  The syntax of the FinalNPT is given by the following ABNF: final-npt = "FinalNPT" HCOLON npt-sec CRLF Example: FinalNPT: 143.250	Yes
OnDemandSessionId	The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.	



Header	Format / Description	Mandatory?
StopPoint	<p>The syntax of the OnDemand SessionId header is as follows. On DemandSessionId: &lt;onDemand SessionId&gt;where&lt;onDemand SessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemand SessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemand SessionId header follows. OnDemandSessionId: be074250cc5a1d98cd50800200c9a66</p> <p>The StartPoint header specifies the point in the asset list at which the Streaming Server should begin streaming.</p> <p>The syntax of the Start Point is given by the following ABNF: start-point = "Start Point" HCOLON segment-npt-specifier CRLF segment-npt-specifier = slot SP npt-sec slot = 1 *DIGIT ; Index (1 based) of asset list element</p> <p>An example of the Start Point header follows. StartPoint: 2 1234.78</p> <p>Note that the NPT value is absolute. That is, the NPT is not relative to the NPT start time specified for that segment of the asset list (SDP payload data).</p> <p>For example, if the asset list indicates that the segment begins at an NPT of 5.0, and the Start Point indicates that the streaming server should begin streaming at NPT</p>	Yes

Header	Format / Description	Mandatory?
	<p>of 8.0, then the streaming server begins streaming at NPT 8.0 and will allow the user to rewind up to 3.0 seconds before hitting the beginning of the segment.</p> <p>The StartPoint NPT value MUST reside between the segment's starting NPT value and ending NPT value.</p>	
Content-Type	<p>The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient</p> <p>Content-Type = "Content-Type" ":" media-type</p> <p>An example is</p> <p>Content-Type: text/parameters</p>	
Content-Length	<p>The Content-Length entity-header field indicates the size of the entity-body, in decimal number of OCTETs, sent to the recipient.</p> <p>Content-Length = "Content-Length" ":" 1*DIGIT</p> <p>An example is</p> <p>Content-Length: 3495</p>	

**Response Body**

The body of the TEARDOWN response message SHALL contain XML data with the root level element <ResponseData>. The following example describes the sub elements that may appear.

```
<ResponseData>
  <ODRMSessionHistory>
  </ODRMSessionHistory>
</ResponseData>
```

**Response Example**

```
RTSP/1.0 200 OK
CSeq: 789
Session: 999
FinalNPT: 44.5
OnDemandSessionId: be074250cc5a11d98cd50800200c9a66
```

---

```
StopPoint: 2 14.5
Content-Type: text/xml
Content-Length: 1234
```

```
<ResponseData>
  <ODRMSessionHistory>
  </ODRMSessionHistory>
</ResponseData>
```

---

## R2 TEARDOWN Response

### Response Format

In (quasi)-BNF the format of the response to this request shall be:

---

```
RTSP/ 1.0 <responseCode> SP <reasonPhrase> CRLF
1 *<header> CRLF
CRLF
<body>
```

---

### Response Headers

Header	Format / Description	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.  "Session" ":" session-id Session header example Session: 10000105	Yes
FinalNPT	The FinalNPT header defines a means for an RTSP server to return the NPT of the stream as known at the time the session is torn down. This will return a value equivalent to	Yes

Header	Format / Description	Mandatory?
	<p>that returned in response to A GET_PARAMETER of "position" without the need to use that method.</p> <p>The syntax of the FinalNPT is given by the following ABNF:</p> <pre>final-npt = "FinalNPT" HCOLON npt-sec CRLF</pre> <p>Example: FinalNPT: 143.250</p>	
OnDemandSessionId	<p>The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.</p> <p>The syntax of the OnDemandSessionId header is as follows. OnDemandSessionId: &lt;onDemandSessionId&gt; where &lt;onDemandSessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemandSessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemandSessionId header follows.</p> <pre>OnDemandSessionId: be 074250cc5a1d98cd50800200c9a66</pre>	
StopPoint	<p>The StartPoint header specifies the point in the asset list at which the Streaming Server should begin streaming.</p> <p>The syntax of the StartPoint is given by the following ABNF:</p> <pre>start-point = "Start Point" HCOLON segment-npt- specifier CRLF segment-npt-specifier = slot SP npt-sec</pre>	Yes

Header	Format / Description	Mandatory?
Content-Type	<p>slot = 1*DIGIT ; Index (1 based) of asset list element An example of the Start Point header follows. StartPoint: 2 1234.78</p> <p>Note that the NPT value is absolute. That is, the NPT is not relative to the NPT start time specified for that segment of the asset list (SDP payload data).</p> <p>For example, if the asset list indicates that the segment begins at an NPT of 5.0, and the Start Point indicates that the streaming server should begin streaming at NPT of 8.0, then the streaming server begins streaming at NPT 8.0 and will allow the user to rewind up to 3.0 seconds before hitting the beginning of the segment.</p> <p>The StartPoint NPT value MUST reside between the segment's starting NPT value and ending NPT value.</p> <p>The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient Content-Type = "Content- Type" ":" media-type An example is Content-Type: text/ parameters</p>	
Content-Length	<p>The Content-Length entity-header field indicates the size of the entity-body, in decimal number of OCTETs, sent to the recipient. Content-Length = "Content- Length" ":" 1*DIGIT An example is Content-Length: 3495</p>	

**Response Body**

The body of the TEARDOWN response message SHALL contain XML data with the root level element <ResponseData>. The following example describes the sub elements that may appear.

```
<ResponseData>
  <ODRMSessionHistory>
  </ODRMSessionHistory>
</ResponseData>
```

**Response Example**

```
RTSP/1.0 200 OK
CSeq: 789
Session: 999
FinalNPT: 44.5
OnDemandSessionId: be074250cc5a11d98cd50800200c9a66
StopPoint: 2 14.5
Content-Type: text/xml
Content-Length: 1234
```

```
<ResponseData>
  <ODRMSessionHistory>
  </ODRMSessionHistory>
</ResponseData>
```

**6.3.10 NGOD Response And Notice Codes**

**RTSP Response Codes**

Code	Message
200	OK
400	Bad Request
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable

Code	Message
408	Request Time Out
410	Gone
413	Request Entity Too Large
415	Unsupported Media Type
451	Invalid Parameter
453	Not Enough Bandwidth
454	Session Not Found
457	Invalid Range
459	Aggregate Operation Not Allowed
461	Unsupported Transport
462	Destination Unreachable
504	Gateway Timeout
505	RTSP Version Not Supported
651	SM Setup Failed - Unknown QAM Group
652	SM Setup Failed - Invalid Request
653	SM Setup Failed - Internal Error
660	PS Setup Failed - No Response
661	PS Setup Failed - Unknown Purchase Token

Code	Message
662	PS Setup Failed - Invalid Request
663	PS Setup Failed - Internal Error

### Announce Request Notice Codes

The following announce codes shall be supported. These are the valid values for the "Notice:" header.

Code	Message	Expected Reaction
2101	End-of-Stream Reached	No NGOD reaction required
2104	Start-of-Stream Reached	No NGOD reaction required
4400	Error Reading Content Data	No NGOD reaction required
5200	Server Resources Unavailable	No NGOD reaction required
5401	Downstream Failure	No NGOD reaction required
5402	Client Session Terminated	No NGOD reaction required
5502	Internal Server Error	No NGOD reaction required
5601	Inband Stream Marker Mismatch	No NGOD reaction required
5601	Bandwidth Exceeded Limit	No NGOD reaction required
5700	Session In Progress	No NGOD reaction required
6000	Encryption Engine Failure	No NGOD reaction required
6001	Stream Bandwidth Exceeds That Available	No NGOD reaction required



Code	Message	Expected Reaction
6004	Downstream Destination Unreachable	No NGOD reaction required
6005	Unable to Encrypt one or more Components	No NGOD reaction required
6006	ECMG Session Failure	No NGOD reaction required

## 6.3.11 NGOD S1

### Overview

NGOD S1 is the main interface through which the On Demand Client (STB) communicates with the Session Resource Manager (SRM). The client uses S1 for session setup and teardown.

The STB uses S1 SETUP message to establish a new session in SRM. The S1 SETUP response message contains information that On Demand Client will use to establish interface C1.

### Limitations

1. Only TCP connections are supported.

### Implementation notes and guidance for STB developers

#### Caution!

These notes are provided in the hope that they will be useful. However, it is not guaranteed that the SRM will follow them. STB developers must test against a real SRM to validate.

1. By default, the SRM listens on port **5544**. The actual port you need to connect to may be affected by the actions of load balancers and other equipment between the STB and SRM.
2. When a TCP connection is used to send a PING (keepalive) message for a session, that connection becomes the **keepalive connection** for that session.
3. When a keepalive connection is closed, its session is torn down automatically.
4. Every session must have its own keepalive connection. A keepalive connection for one session cannot be re-used for a different session.
5. The connection used for the initial SETUP request may be left open and turned into a keepalive connection for the same session. Similarly, a keepalive connection may be used to send TEARDOWN or SETUP messages.
6. Before sending a SETUP, the STB must perform the NGOD E2 process with the SDP (not the SRM) in order to obtain a purchase token. See the SDP documentation.

7. The STB should send a single `qam_name` in its SETUP request. The response contains the `qam_name` of the frequency the STB needs to tune to; this may be different.
8. When a session is torn down due to an upstream failure, the Notice from the upstream device will be forwarded verbatim in the ANNOUNCE sent to the client, using the keepalive connection for the session.
9. When a session is torn down by the operator (forced disconnect) a 5402 Notice will be sent, also using the keepalive connection for the session.
10. When an ANNOUNCE is sent via a keepalive connection, the SRM will close the connection as soon as the ANNOUNCE has been sent. Don't bother to acknowledge ANNOUNCE requests.
11. TEARDOWNS always succeed.
  - Tearing down a non-existent session is a no-op (and hence successful).
  - Tearing down a session which is already being torn down is a no-op as well (and hence successful).
12. In deployments which have more than one SRM instance (the usual case) it is not necessary to implement any kind of affinity support in load balancers. It is not necessary for the keepalive connection to be sent to the same SRM instance which received the SETUP for that session.
13. Tearing down a session (and hence releasing its resources) does not guarantee that it will be possible to set up another session using the same resources.

### 6.3.11.1 S1 ANNOUNCE

#### S1 ANNOUNCE Request

SRM sends S1 ANNOUNCE request to the On Demand Client (STB) in the following scenarios:

1. When an NGOD session hasn't been kept alive and then the SRM checkKeepAlive job sends S1 ANNOUNCE out as a pre-check before adding it to the teardown queue
2. An incoming R2 ANNOUNCE from VS (e.g. following termination of a stream), which it then forwards onto to the On Demand Client (STB)

**Note** | The flow shown in diagram need to be confirmed with SRM code!

#### Request Format

Announce is a regular RTSP request from Video Server, with no body. In (quasi-) BNF the format of this request shall be

---

```
ANNOUNCE SP <uri> SP RTSP/1.0 CRLF
1 *<header> CRLF
```

---

#### Request URI

The request `<uri>` shall have the form:

---

```
rtsp: //<logicalSrmAddress>:<logicalSrmPort>/
```

---

Field	Description
<logicalSrmAddress>	The DNS name or IP address of the SRM.
<logicalSrmPort>	The TCP port for the SRM.

---

### RTSP Request Headers

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / Definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Require	<p>The Require header defines a mechanism by which the client indicates in a request that certain extensions are required to fulfill this request. The syntax of the Require header is as follows.</p> <p>Require: com.comcast.ngod.&lt;interface-id&gt; where &lt;interface-id&gt; is the NGOD identifier of the interface, e.g. "s1", "c1", "r2".</p> <p>The specific NGOD interfaces that use this feature will define the exact usage. An example of the Require header follows. Require: com.comcast.ngod.s1</p>	
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.	Yes

Header	Format / Definition	Mandatory?
	<p>"Session" ":" session-id</p> <p>Session header example Session: 10000105</p>	
Notice	<p>The Notice header defines information sent from a RTSP server to a RTSP client within an ANNOUNCE message.</p> <p>The syntax of the Notice header is as follows. Notice = "Notice" HCOLON event *(COMMA event)</p> <p>HCOLON = * ( SP / HT ) ":" SWS SP = %X20 HT = %X09 SWS = [ LWS ] LWS = [ CRLF ] 1 * ( SP / HT ) CRLF = CR LF CR = %X0D LF = %X0A</p> <p>event = announce-code SP text- description SP event-date SP npt</p> <p>announce-code = ; see Table for the list of possible values text- description = quoted-string quoted- string = ( DQ * qdtext DQ ) qdtext = %X20-21 / %X23—%X7E / %80— %XFFDQ = %X22</p> <p>event-date = "event-date" EQUAL utc-time</p> <p>EQUAL = SWS "=" SWS utc-time = utc-date "T" utc-clock "Z" utc- date = 8DIGIT ; Year (4) Month (2) Date (2) utc-clock = 6DIGIT [ ". " fraction ] ; Hour (2) Minute (2) Second (2) fraction = 1 * DIGIT DIGIT = %X30 - %X39</p> <p>npt = "npt" EQUAL [ npt - sec ] ; only if value is known npt-sec = 1 * DIGIT [ "." * DIGIT ] ; position as decimal sec.msec</p> <p>The NPT must be specified in decimal seconds.milliseconds format. This format is consistent with the [RFC 2326] definition. Where Notice is used but npt is not known, &lt;npt-sec&gt; (but not the attribute) will be omitted.</p>	Yes

Header	Format / Definition	Mandatory?
	<p>Example of the Notice header for end of stream (2314.223 sec is actual end of content): Notice: 2101 "End-of-Stream Reached" event-date=20150316T064707.735Z npt=2314.223</p> <p>Example of the Notice header for beginning of stream: Notice: 2101 "Start-of-Stream Reached"</p> <p>Please see Section ANNOUNCE Codes below for a full list of the valid options for Notice-code.</p>	
OnDemandSessionId	<p>The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.</p> <p>The syntax of the OnDemandSessionId header is as follows. OnDemandSessionId: &lt;onDemandSessionId&gt; where &lt;onDemandSessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemandSessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemandSessionId header follows. OnDemandSessionId: be074250cc5a1d98cd50800200c9a66</p>	Yes

**Request Example**

```
ANNOUNCE rtsp://sessionmanager.comcast.com:554/ RTSP/1.0
CSeq: 315
```

---

Require: com.comcast.ngod.s1

Session: 47112345

Notice: 5402 "Client Session Terminated" event-date=20160401T023735.13Z npt=342.554

OnDemandSessionId: be074250cc5a11d98cd50800200c9a66

---

## S1 ANNOUNCE Response

### Response Format

In (quasi-)BNF the format of the response to this request shall be:

---

```
RTSP/ 1.0 <responseCode> SP <reasonPhrase> CRLF
1 *<header> CRLF
```

---

### Response Headers

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.  "Session" ":" session-id  Session header example Session: 10000105	Yes
OnDemandSessionId	The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.  The syntax of the OnDemandSessionId header is as follows. OnDemandSessionId: <onDemand	Yes

---

Header	Format / definition	Mandatory?
	<p>SessionId&gt;where&lt;onDemand SessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemand SessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemand SessionId header follows. OnDemandSessionId: be074250cc5a11d98cd50800200c9a66</p>	

**Response Example**

```
RTSP/1.0 200 OK
CSeq: 315
Session: 47112345
OnDemandSessionId: be074250cc5a11d98cd50800200c9a66
```

**Announce Request Notice Codes**

The following announce codes shall be supported. These are the valid values for the "Notice:" header.

Code	Message	Expected Reaction
2101	End-of-Stream Reached	No NGOD reaction required
2104	Start-of-Stream Reached	No NGOD reaction required
4400	Error Reading Content Data	No NGOD reaction required
5200	Server Resources Unavailable	No NGOD reaction required
5401	Downstream Failure	No NGOD reaction required

Code	Message	Expected Reaction
5402	Client Session Terminated	No NGOD reaction required
5502	Internal Server Error	No NGOD reaction required
5601	Inband Stream Marker Mismatch	No NGOD reaction required
5601	Bandwidth Exceeded Limit	No NGOD reaction required
5700	Session In Progress	No NGOD reaction required
6000	Encryption Engine Failure	No NGOD reaction required
6001	Stream Bandwidth Exceeds That Available	No NGOD reaction required
6004	Downstream Destination Unreachable	No NGOD reaction required
6005	Unable to Encrypt one or more Components	No NGOD reaction required
6006	ECMG Session Failure	No NGOD reaction required

**Announce Response Status Codes**

The following is the complete set of NGOD RTSP response codes. Specific descriptions or clarifications as to how these codes relate to specific NGOD interfaces may be found in the respective interface specifications.

Code	Message
200	OK
205	Reset Context  This result status code is used in response to a PLAY request where the server is able to play the content, but either the returned Range header, or Scale header has been changed to comply with a restriction on a play list item.



Code	Message
	This code is not necessary if the Range header indicates a different value due to the precision necessary for a specific splice location and the Scale header indicates a different value due to the specific trick speeds that are supported by the content.
300	Redirect — Multiple Choices
400	Bad Request
401	Unauthorized
403	Forbidden This status result code is used in response to a PLAY, or PAUSE where the server is unable to deliver the request change in play out of the content due to a restriction on a play list item.
404	Not Found
405	Method Not Allowed
406	Not Acceptable
408	Request Time Out
410	Gone
413	Request Entity Too Large
414	Request URI TOO Long
415	Unsupported Media Type
451	Invalid Parameter
453	Not Enough Bandwidth

Code	Message
454	Session Not Found
455	Method Not Valid In This State
456	Header Field Not Valid
457	Invalid Range
458	Parameter Is Read-Only
459	Aggregate Operation Not Allowed
460	Only Aggregate Operation Allowed
461	Unsupported Transport
462	Destination Unreachable
499	No such content
500	Internal Server Error
501	Not implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
505	RTSP Version Not Supported
551	Option Not Supported
599	No available streaming servers

Code	Message
600	Insufficient Encryption Resources
601	IECMG Not Available or Responding
602	Stream/ Service identi#ed for encryption processing indicates it is pre-encrypted content
650	SM Setup Failed - No Response
651	SM Setup Failed — Unknown QAM Group
652	SM Setup Failed — Invalid Request
653	SM Setup Failed — Internal Error
660	PS Setup Failed - No Response
661	PS Setup Failed - Unknown Purchase Token
662	PS Setup Failed — Invalid Request
663	PS Setup Failed — Internal Error
670	ERM Setup Failed - No Response
671	ERM Setup Failed — Invalid Request
672	ERM Setup Failed — QAM Bandwidth Not Available
673	ERM Setup Failed — Network Bandwidth Not Available
674	ERM Setup Failed — Program Not Available
675	ERM Setup Failed — Service Group Not Found
676	ERM Setup Failed — QAM Groups Not Found

Code	Message
677	ERM Setup Failed — QAM Not Available
678	ERM Setup Failed — Edge Device Not Available
679	ERM Setup Failed — Internal Error
690	APM Locate Asset Failed — No Response
691	APM Locate Asset Failed - Asset Not Available
692	APM Locate Asset Failed — Invalid Request
693	APM Locate Asset Failed - Internal Error
750	ODRM Setup Failed — No Response
751	ODRM Setup Failed — Unknown SOP Group
752	ODRM Setup Failed — Bandwidth Not Available
753	ODRM Setup Failed — Stream Not Available
754	ODRM Setup Failed — Asset Not Available
755	ODRM Setup Failed — Invalid Request
756	ODRM Setup Failed — Internal Error
770	Server Setup Failed — No Response
771	Server Setup Failed — Asset Not Found
772	Server Setup Failed — SOP Not Available
773	Server Setup Failed — Unknown SOP Group

Code	Message
774	Server Setup Failed — Unknown SOP Names
775	Server Setup Failed — insufficient Volume Bandwidth
776	Server Setup Failed — insufficient Network Bandwidth
777	Server Setup Failed — invalid Request
778	Server Setup Failed — internal Error

### Announce Request Notice Codes

The following announce codes shall be supported. These are the valid values for the "Notice:" header.

Code	Message	Expected Reaction
2101	End-of-Stream Reached	No NGOD reaction required
2104	Start-of-Stream Reached	No NGOD reaction required
4400	Error Reading Content Data	No NGOD reaction required
5200	Server Resources Unavailable	No NGOD reaction required
5401	Downstream Failure	No NGOD reaction required
5402	Client Session Terminated	No NGOD reaction required
5502	Internal Server Error	No NGOD reaction required
5601	Inband Stream Marker Mismatch	No NGOD reaction required
5601	Bandwidth Exceeded Limit	No NGOD reaction required
5700	Session In Progress	No NGOD reaction required

Code	Message	Expected Reaction
6000	Encryption Engine Failure	No NGOD reaction required
6001	Stream Bandwidth Exceeds That Available	No NGOD reaction required
6004	Downstream Destination Unreachable	No NGOD reaction required
6005	Unable to Encrypt one or more Components	No NGOD reaction required
6006	ECMG Session Failure	No NGOD reaction required

### Announce Response Status Codes

The following is the complete set of NGOD RTSP response codes. Specific descriptions or clarifications as to how these codes relate to specific NGOD interfaces may be found in the respective interface specifications.

Code	Message
200	OK
205	Reset Context  This result status code is used in response to a PLAY request where the server is able to play the content, but either the returned Range header, or Scale header has been changed to comply with a restriction on a play list item.  This code is not necessary if the Range header indicates a different value due to the precision necessary for a specific splice location and the Scale header indicates a different value due to the specific trick speeds that are supported by the content.
300	Redirect — Multiple Choices
400	Bad Request
401	Unauthorized

Code	Message
403	Forbidden  This status result code is used in response to a PLAY, or PAUSE where the server is unable to deliver the request change in play out of the content due to a restriction on a play list item.
404	Not Found
405	Method Not Allowed
406	Not Acceptable
408	Request Time Out
410	Gone
413	Request Entity Too Large
414	Request URI TOO Long
415	Unsupported Media Type
451	Invalid Parameter
453	Not Enough Bandwidth
454	Session Not Found
455	Method Not Valid In This State
456	Header Field Not Valid
457	Invalid Range
458	Parameter Is Read-Only
459	Aggregate Operation Not Allowed

Code	Message
460	Only Aggregate Operation Allowed
461	Unsupported Transport
462	Destination Unreachable
499	No such content
500	Internal Server Error
501	Not implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
505	RTSP Version Not Supported
551	Option Not Supported
599	No available streaming servers
600	Insufficient Encryption Resources
601	IECMG Not Available or Responding
602	Stream/ Service identi#ed for encryption processing indicates it is pre-encrypted content
650	SM Setup Failed - No Response
651	SM Setup Failed — Unknown QAM Group
652	SM Setup Failed — Invalid Request



Code	Message
653	SM Setup Failed — Internal Error
660	PS Setup Failed - No Response
661	PS Setup Failed - Unknown Purchase Token
662	PS Setup Failed — Invalid Request
663	PS Setup Failed — Internal Error
670	ERM Setup Failed - No Response
671	ERM Setup Failed — Invalid Request
672	ERM Setup Failed — QAM Bandwidth Not Available
673	ERM Setup Failed — Network Bandwidth Not Available
674	ERM Setup Failed — Program Not Available
675	ERM Setup Failed — Service Group Not Found
676	ERM Setup Failed — QAM Groups Not Found
677	ERM Setup Failed — QAM Not Available
678	ERM Setup Failed — Edge Device Not Available
679	ERM Setup Failed — Internal Error
690	APM Locate Asset Failed — No Response
691	APM Locate Asset Failed - Asset Not Available
692	APM Locate Asset Failed — Invalid Request

Code	Message
693	APM Locate Asset Failed - Internal Error
750	ODRM Setup Failed — No Response
751	ODRM Setup Failed — Unknown SOP Group
752	ODRM Setup Failed — Bandwidth Not Available
753	ODRM Setup Failed — Stream Not Available
754	ODRM Setup Failed — Asset Not Available
755	ODRM Setup Failed — Invalid Request
756	ODRM Setup Failed — Internal Error
770	Server Setup Failed — No Response
771	Server Setup Failed — Asset Not Found
772	Server Setup Failed — SOP Not Available
773	Server Setup Failed — Unknown SOP Group
774	Server Setup Failed — Unknown SOP Names
775	Server Setup Failed — insufficient Volume Bandwidth
776	Server Setup Failed — insufficient Network Bandwidth
777	Server Setup Failed — invalid Request
778	Server Setup Failed — internal Error

### 6.3.11.2 S1 PING

## S1 PING Request

The On Demand Client (STB) interacts with SRM to send "heartbeat" messages to convey to the SRM that the client is still alive. This is accomplished by the client by periodically sending a RTSP PING request to SRM. The frequency of the PING message between On Demand Client (STB) and SRM is subject to further definition. SRM is very flexible in this regards and offers to configure interval value, number of misses and an option to set a grace period.

### Request Format

PING is a regular RTSP request, with no body. In (quasi-) BNF the format of this request shall be

```
PING SP <uri> SP RTSP/ 1.0 CRLF
1 *<header> CRLF
```

### Request URI

The request <uri> shall have the form:

```
rtsp: //<logicalSrmAddress>:<logicalSrmPort>/
```

Field	Description
<logicalSrmAddress>	The DNS name or IP address of the SRM.
<logicalSrmPort>	The TCP port for the SRM.

### RTSP Request Headers

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / Definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Require	The Require header defines a mechanism by which the client indicates in a request that certain extensions are required to fulfill this request. The syntax of the Require header is as follows.  Require: com.comcast.nqod.<interface-id>where<interface-id> is	Yes

Header	Format / Definition	Mandatory?
Session	<p>the NGOD identi#er of the interface, e.g. "s1", "c1", "r2".</p> <p>The speci#c NGOD interfaces that use this feature will de#ne the exact usage.An example of the Require header follows.Require: com.comcast.ngod.s1</p>	Yes
OnDemandSessionId	<p>The OnDemandSessionId header de#nes the unique session identi#er that the NGOD Session Manager assigns to a given session.</p> <p>The syntax of the OnDemand SessionId header is as follows.On DemandSessionId: &lt;onDemand SessionId&gt;where&lt;onDemand SessionId&gt; is a string representation of a 128-bit UUID as de#ned by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness.The OnDemand SessionId must follow the format speci#ed in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemand SessionId header follows. OnDemandSessionId: be 074250cc5a1d98cd50800200c9a66</p>	Yes

**Request Example**

---

```

PING rtsp://session-manager.nagra.com/ RTSP/1.0
CSeq: 315
Require: com.comcast.ngod.s1
Session: 47112345
OnDemandSessionId: be074250cc5a11d98cd50800200c9a66
    
```

---

S1 PING Response

**Response Format**

In (quasi-) BNF the format of the response to this request shall be:

---

```

RTSP/ 1.0 <responseCode> SP <reasonPhrase> CRLF
1 *<header> CRLF
    
```

---

**Response Headers**

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.  "Session" ":" session-id  Session header example Session: 10000105	Yes
OnDemandSessionId	The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.	Yes

---

Header	Format / definition	Mandatory?
	<p>The syntax of the OnDemand SessionId header is as follows. OnDemandSessionId: &lt;onDemandSessionId&gt;where&lt;onDemandSessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemandSessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemandSessionId header follows. OnDemandSessionId: be074250cc5a11d98cd50800200c9a66</p>	

**Response Example**

```
RTSP/1.0 200 OK
CSeq: 315
session: 47112345
OnDemandSessionId: be074250cc5a11d98cd50800200c9a66
```

**Error Codes**

Code	Message
200	OK
400	Bad Request
403	Forbidden
404	Not Found
405	Method Not Allowed

<b>Code</b>	<b>Message</b>
406	Not Acceptable
408	Request Time Out
410	Gone
413	Request Entity Too Large
415	Unsupported Media Type
451	Invalid Parameter
453	Not Enough Bandwidth
454	Session Not Found
457	Invalid Range
459	Aggregate Operation Not Allowed
461	Unsupported Transport
462	Destination Unreachable
504	Gateway Timeout
505	RTSP Version Not Supported
651	SM Setup Failed - Unknown QAM Group
652	SM Setup Failed - Invalid Request
653	SM Setup Failed - Internal Error
660	PS Setup Failed - No Response

Code	Message
661	PS Setup Failed - Unknown Purchase Token
662	PS Setup Failed - Invalid Request
663	PS Setup Failed - Internal Error

## S1 PING Response

### Response Format

In (quasi-) BNF the format of the response to this request shall be:

```
RTSP/ 1.0 <responseCode> SP <reasonPhrase> CRLF
1 *<header> CRLF
```

### Response Headers

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.  "Session" ":" session-id  Session header example Session: 10000105	Yes
OnDemandSessionId	The OnDemandSessionId header defines the unique session identifier	Yes



Header	Format / definition	Mandatory?
	<p>that the NGOD Session Manager assigns to a given session.</p> <p>The syntax of the OnDemand SessionId header is as follows. On DemandSessionId: &lt;onDemand SessionId&gt;where&lt;onDemand SessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemand SessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemand SessionId header follows. OnDemandSessionId: be074250cc5a11d98cd50800200c9a66</p>	

**Response Example**

```
RTSP/1.0 200 OK
CSeq: 315
Session: 47112345
OnDemandSessionId: be074250cc5a11d98cd50800200c9a66
```

**Error Codes**

Code	Message
200	OK
400	Bad Request
403	Forbidden
404	Not Found

Code	Message
405	Method Not Allowed
406	Not Acceptable
408	Request Time Out
410	Gone
413	Request Entity Too Large
415	Unsupported Media Type
451	Invalid Parameter
453	Not Enough Bandwidth
454	Session Not Found
457	Invalid Range
459	Aggregate Operation Not Allowed
461	Unsupported Transport
462	Destination Unreachable
504	Gateway Timeout
505	RTSP Version Not Supported
651	SM Setup Failed - Unknown QAM Group
652	SM Setup Failed - Invalid Request
653	SM Setup Failed - Internal Error

Code	Message
660	PS Setup Failed - No Response
661	PS Setup Failed - Unknown Purchase Token
662	PS Setup Failed - Invalid Request
663	PS Setup Failed - Internal Error

## Error Codes

Code	Message
200	OK
400	Bad Request
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
408	Request Time Out
410	Gone
413	Request Entity Too Large
415	Unsupported Media Type
451	Invalid Parameter
453	Not Enough Bandwidth

Code	Message
454	Session Not Found
457	Invalid Range
459	Aggregate Operation Not Allowed
461	Unsupported Transport
462	Destination Unreachable
504	Gateway Timeout
505	RTSP Version Not Supported
651	SM Setup Failed - Unknown QAM Group
652	SM Setup Failed - Invalid Request
653	SM Setup Failed - Internal Error
660	PS Setup Failed - No Response
661	PS Setup Failed - Unknown Purchase Token
662	PS Setup Failed - Invalid Request
663	PS Setup Failed - Internal Error

### 6.3.11.3 S1 SETUP

#### S1 SETUP Request

The STB interacts with SRM with RTSP Setup request and response messages to establish new session

#### Request Format

To establish a session, the STB shall send an RTSP Setup request to the SRM. This is a regular RTSP request, with no body. In (quasi-) BNF the format of this request shall be

```
SETUP SP <uri> SP RTSP/1.0 CRLF
1*<header> CRLF
```

**Request URI**

The request <uri> shall have the form:

Please note that all of the variables in URI are mandatory for a valid RTSP URL for an NGOD S1 SETUP request.

```
rtsp://<logicalSrmAddress>:<logicalSrmPort>;purchaseToken=<purchase-token>;server
Id=<server-id>
```

Field	Description
<logicalSrmAddress>	The DNS name or IP address of the SRM.
<logicalSrmPort>	The TCP port for the SRM.
<purchaseToken >	A 128-bit UUID string token which is negotiated between application server and STB.
<server-id>	This identifier represents the IP address of the application server.

**RTSP Request Headers**

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / Definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Require	The Require header defines a mechanism by which the client indicates in a request that certain extensions are required to fulfill this request. The syntax of the Require header is as follows.	Yes

Header	Format / Definition	Mandatory?				
	<p>Require: com.comcast.ngod. &lt;interface-id&gt;where&lt;interface-id&gt; is the NGOD identi#er of the interface, e.g. "s1", "c1", "r2".</p> <p>The speci#c NGOD interfaces that use this feature will de#ne the exact usage.An example of the Require header follows.Require: com.comcast.ngod.s1</p>					
Transport	<p>The Transport header de#nes the transport parameters acceptable to the client for data transmission.An RTSP SETUP request may specify multiple transport headers which expresses that the device receiving the setup request has a choice of which transport to use.</p> <p>The receiving device informs the sending device of which choice it made in the (single) transport header of the setup response it sends.</p> <p>MP2T/DVBC/QAM; unicast; client=&lt;client-id&gt;; qam_name=&lt;qam-name&gt;</p> <hr/> <p><b>Transport param</b></p> <hr/> <table border="0"> <tr> <td style="vertical-align: top;">&lt;client- IP id&gt;</td> <td style="vertical-align: top;">A unique identifier of the client, usually a MAC address or Unit address.</td> </tr> <tr> <td style="vertical-align: top;">&lt;qam- String name&gt;</td> <td style="vertical-align: top;">A string representation of a QAM name discovered by the client.</td> </tr> </table>	<client- IP id>	A unique identifier of the client, usually a MAC address or Unit address.	<qam- String name>	A string representation of a QAM name discovered by the client.	Yes
<client- IP id>	A unique identifier of the client, usually a MAC address or Unit address.					
<qam- String name>	A string representation of a QAM name discovered by the client.					

Header	Format / Definition	Mandatory?
	<p><b>Transport param</b></p> <p><b>Format Description</b></p> <p>Please Note: It is possible to represent multiple qam-name values in a single Transport header, as shown in examples</p>	
ClientSessionId	<p>The ClientSessionId header defines a unique session identifier created by an On Demand Client. This identifier is globally unique. ClientSessionId is used in the S1 session setup message as the client's identifier of the session.</p> <p>The syntax of a ClientSessionId header is identical to the "session-id" defined in section 3.4 of [RFC 2326]. The syntax of a ClientSessionId is a 20 character ASCII representation of a 10 byte hexadecimal value.</p> <p>The most significant 6 bytes are the client ID, the least significant 4 bytes are a session ID unique to the client. The combination of the two provides a globally unique identifier.</p> <p>An example of the ClientSessionId header follows. ClientSessionId: 00AF123456DE00000001</p> <p>This ClientSessionId represents a client with a MAC address of 00:AF:12:34:56:DE and a session ID of 00000001.</p> <p>It should be noted that the ClientSessionId is different from the RTSP session #eld between the On Demand Client and Streaming</p>	Yes

Header	Format / Definition	Mandatory?
	Server.ClientSessionId is generated by the On Demand Client. The RTSP session #eld between the On Demand Client and Streaming Server is assigned by the Streaming Server and returned to the client over S3 for further stream control.	

### S1 SETUP Request Example

```
SETUP rtsp://sessionmanager.hot.com:554/;purchase
Token=44272c84d21d4d52b08e464925940397;serverID=1.1.1.1 RTSP/1.0
CSeq: 345
Require: com.comcast.ngod.s1
Transport:MP2T/DVBC/QAM;unicast;client=00AF123456DE;qam_name=lab2.qam.5,MP2T/DVBC/
QAM;unicast;client=00AF123456DE;qam_name=lab2.qam.10
ClientSessionId: 167af045dcfc4efbalcd09b8e0d815fb
```

### S1 SETUP Response

The S1 SETUP response is consist of response code & phrase and some RTSP headers.

```
RTSP/1.0 SP <code> SP <phrase> SP CRLF
1*<header> CRLF
CRLF
<body>
```

The Transport header which was part of S1 SETUP request will come back with some additional values.

Header	Format / Definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response.The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.	Yes



Header	Format / Definition	Mandatory?				
	<p>"Session" ":" session-id [ ";" "timeout" "=" delta-seconds ]</p> <p>Session header example Session: 10000105;timeout=60</p> <p>The timeout parameter is only allowed in a response header. The server uses it to indicate to the client how long the server is prepared to wait between RTSP commands before closing the session due to lack of activity. The timeout is measured in seconds, with a default of 60 seconds (1 minute).</p>					
Transport	<p>The Transport header defines the transport parameters acceptable to the client for data transmission. An RTSP SETUP request may specify multiple transport headers which expresses that the device receiving the setup request has a choice of which transport to use.</p> <p>The receiving device informs the sending device of which choice it made in the (single) transport header of the setup response it sends.</p> <p>Transport: MP2T/DVBC/QAM; unicast; [destination=&lt;destination&gt;;] [qam_name=&lt;qam-name&gt;;]</p>	No				
	<table border="1"> <thead> <tr> <th>Field</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>&lt;destination&gt;</td> <td>destination describes the "<b>&lt;frequency&gt;.&lt;program-number&gt;</b>" where &lt;frequency&gt; describes the frequency in Hertz with which to tune to the request stream</td> </tr> </tbody> </table>	Field	Description	<destination>	destination describes the " <b>&lt;frequency&gt;.&lt;program-number&gt;</b> " where <frequency> describes the frequency in Hertz with which to tune to the request stream	
Field	Description					
<destination>	destination describes the " <b>&lt;frequency&gt;.&lt;program-number&gt;</b> " where <frequency> describes the frequency in Hertz with which to tune to the request stream					

Header	Format / Definition	Mandatory?						
	<table border="1"> <thead> <tr> <th data-bbox="606 481 710 515">Field</th> <th data-bbox="718 481 981 515">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="606 548 710 694"></td> <td data-bbox="718 548 981 694">and &lt;program-number&gt; is the program number of the requested stream.</td> </tr> <tr> <td data-bbox="606 728 710 1052">&lt;qam-name&gt;</td> <td data-bbox="718 728 981 1052">The qam-name is a string describing the assigned qam-name for the transport stream that the STB needs to tune to, this can be compared with qam-name list from client auto-discovery.</td> </tr> </tbody> </table>	Field	Description		and <program-number> is the program number of the requested stream.	<qam-name>	The qam-name is a string describing the assigned qam-name for the transport stream that the STB needs to tune to, this can be compared with qam-name list from client auto-discovery.	
Field	Description							
	and <program-number> is the program number of the requested stream.							
<qam-name>	The qam-name is a string describing the assigned qam-name for the transport stream that the STB needs to tune to, this can be compared with qam-name list from client auto-discovery.							
OnDemandSessionId	<p>The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.</p> <p>The syntax of the OnDemandSessionId header is as follows. OnDemandSessionId: &lt;onDemandSessionId&gt; where &lt;onDemandSessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemandSessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemandSessionId header follows. OnDemandSessionId: be O74250cc5a1d98cd50800200c9a66</p>	Yes						

Header	Format / Definition	Mandatory?
ClientSessionId	<p>The ClientSessionId header defines a unique session identifier created by an On Demand Client. This identifier is globally unique. ClientSessionId is used in the S1 session setup message as the client's identifier of the session.</p> <p>The syntax of a ClientSessionId header is identical to the "session-id" defined in section 3.4 of [RFC 2326]. The syntax of a ClientSessionId is a 20 character ASCII representation of a 10 byte hexadecimal value.</p> <p>The most significant 6 bytes are the client ID, the least significant 4 bytes are a session ID unique to the client. The combination of the two provides a globally unique identifier.</p> <p>An example of the ClientSessionId header follows. ClientSessionId: 00AF123456DE00000001</p> <p>This ClientSessionId represents a client with a MAC address of 00:AF:12:34:56:DE and a session ID of 00000001.</p> <p>It should be noted that the ClientSessionId is different from the RTSP session field between the On Demand Client and Streaming Server. ClientSessionId is generated by the On Demand Client. The RTSP session field between the On Demand Client and Streaming Server is assigned by the Streaming Server and returned to the client over S3 for further stream control.</p>	Yes
Content-type	<p>The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient</p> <p>Content-Type = "Content-Type" ":" media-type</p> <p>An example is</p> <p>Content-Type: text/parameters</p>	Yes

Header	Format / Definition	Mandatory?
Content-length	<p>The Content-Length entity-header field indicates the size of the entity-body, in decimal number of OCTETs, sent to the recipient.</p> <p>Content-Length = "Content-Length" ":" 1*DIGIT</p> <p>An example is</p> <p>Content-Length: 3495</p>	Yes

## S1 SETUP Response Example

```

RTSP/1.0 200 OK
CSeq: 345
Session: 821130592
Transport:MP2T/DVBC/QAM;unicast;destination=24000000.23
OnDemandSessionId: be074250cc5a11d98cd50800200c9a66
ClientSessionId: 167af045dcfc4efba1cd09b8e0d815fb
Content-type: application/sdp
Content-length: 149

v=0
o=- 777 2890842817 IN IP4 1.2.3.4
s=
t=0 0
a=control:rtsp://videoserver234.comcast.com:554/9876
c=IN IP4 0.0.0.0
m=video 0 udp MP2T

```

## Error Codes

Code	Message
200	OK
400	Bad Request
403	Forbidden
404	Not Found

<b>Code</b>	<b>Message</b>
405	Method Not Allowed
406	Not Acceptable
408	Request Time Out
410	Gone
413	Request Entity Too Large
415	Unsupported Media Type
451	Invalid Parameter
453	Not Enough Bandwidth
454	Session Not Found
457	Invalid Range
459	Aggregate Operation Not Allowed
461	Unsupported Transport
462	Destination Unreachable
504	Gateway Timeout
505	RTSP Version Not Supported
651	SM Setup Failed - Unknown QAM Group
652	SM Setup Failed - Invalid Request
653	SM Setup Failed - Internal Error

Code	Message
660	PS Setup Failed - No Response
661	PS Setup Failed - Unknown Purchase Token
662	PS Setup Failed - Invalid Request
663	PS Setup Failed - Internal Error

### 6.3.11.4 S1 TEARDOWN

#### S1 TEARDOWN Request

The STB interacts with SRM with RTSP Teardown request and response to teardown an existing session.

##### Request Format

To teardown a session, the STB shall send an RTSP Teardown request to the SRM. This is a regular RTSP request, with no body. In (quasi-) BNF the format of this request shall be

```
TEARDOWN SP <uri> SP RTSP/1.0 CRLF
1 *<header> CRLF
```

##### Request URI

The request <uri> shall have the form:

```
rtsp://<logicalSrmAddress>:<logicalSrmPort>/
```

Field	Description
<logicalSrmAddress>	The DNS name or IP address of the SRM.
<logicalSrmPort>	The TCP port for the SRM.

##### RTSP Request Headers

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / Definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Require	<p>The Require header defines a mechanism by which the client indicates in a request that certain extensions are required to fulfill this request. The syntax of the Require header is as follows.</p> <p>Require: com.comcast.ngod.&lt;interface-id&gt;where&lt;interface-id&gt; is the NGOD identifier of the interface, e.g. "s1", "c1", "r2".</p> <p>The specific NGOD interfaces that use this feature will define the exact usage. An example of the Require header follows. Require: com.comcast.ngod.s1</p>	Yes
Reason	<p>The Reason header defines the reason for a particular TEARDOWN message.</p> <p>The syntax for the Reason header is as follows. Reason: &lt;reason-code&gt; "&lt;reason-text&gt;" where 1. &lt;reason-code&gt; is an integer 2. &lt;reason-text&gt; is a textual description of the reason</p> <p>Please see Section RTSP TEARDOWN Reason Codes at the bottom of this page for a full list of the valid reason codes.</p> <p>An example of the Reason header follows. Reason: 200 "user pressed stop"</p>	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return	Yes

Header	Format / Definition	Mandatory?
	<p>it for any request related to that session.</p> <p>"Session" ":" session-id</p> <p>Session header example Session: 10000105</p>	
OnDemandSessionId	<p>The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.</p> <p>The syntax of the OnDemandSessionId header is as follows. OnDemandSessionId: &lt;onDemandSessionId&gt; where &lt;onDemandSessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemandSessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemandSessionId header follows. OnDemandSessionId: be074250cc5a1d98cd50800200c9a66</p>	Yes
ClientSessionId	<p>The ClientSessionId header defines a unique session identifier created by an On Demand Client. This identifier is globally unique. ClientSessionId is used in the S1 session setup message as the client's identifier of the session.</p> <p>The syntax of a ClientSessionId header is identical to the "session-id" defined in section 3.4 of [RFC 2326]. The syntax of a ClientSessionId is a 20 character ASCII representation of a 10 byte hexadecimal value.</p> <p>The most significant 6 bytes are the client ID, the least significant 4</p>	Yes



Header	Format / Definition	Mandatory?
	<p>bytes are a session ID unique to the client. The combination of the two provides a globally unique identifier.</p> <p>An example of the ClientSessionId header follows. ClientSessionId: 00AF123456DE00000001</p> <p>This ClientSessionId represents a client with a MAC address of 00:AF:12:34:56:DE and a session ID of 00000001.</p> <p>It should be noted that the ClientSessionId is different from the RTSP session #eld between the On Demand Client and Streaming Server. ClientSessionId is generated by the On Demand Client. The RTSP session #eld between the On Demand Client and Streaming Server is assigned by the Streaming Server and returned to the client over S3 for further stream control.</p>	

**Request Example**

```
TEARDOWN rtsp://sessionmanager2.hot.com/ RTSP/1.0
CSeq: 315
Require: com.comcast.ngod.s1
Reason: 200 "User pressed stop"
Session: 47112345
OnDemandSessionId: be074250cc5a11d98cd50800200c9a66
ClientSessionId: 167af045dcfc4efba1cd09b8e0d815fb
```

S1 TEARDOWN Response

**Response Format**

In (quasi-)BNF the format of the response to this request shall be:

```
RTSP/ 1.0 <responseCode> SP <reasonPhrase> CRLF
1 *<header> CRLF
```

**Response Headers**

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	<p>This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.</p> <p>"Session" ":" session-id</p> <p>Session header example Session: 10000105</p>	Yes
OnDemandSessionId	<p>The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.</p> <p>The syntax of the OnDemandSessionId header is as follows. OnDemandSessionId: &lt;onDemandSessionId&gt; where &lt;onDemandSessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemandSessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemandSessionId header follows. OnDemandSessionId: be074250cc5a1d98cd50800200c9a66</p>	Yes
ClientSessionId	<p>The ClientSessionId header defines a unique session identifier created by an On Demand Client. This</p>	Yes

Header	Format / definition	Mandatory?
	<p>identi#er is globally unique. Client SessionId is used in the S1 session setup message as the client's identi#er of the session.</p> <p>The syntax of a ClientSessionId header is identical to the "session-id" de#ned in section 3.4 of [RFC 2326]. The syntax of a ClientSessionId is a 20 character ASCII representation of a 10 byte hexadecimal value.</p> <p>The most signi#cant 6 bytes are the client ID, the least signi#cant 4 bytes are a session ID unique to the client. The combination of the two provides a globally unique identi#er.</p> <p>An example of the ClientSessionId header follows. ClientSessionId: 00AF123456DE00000001</p> <p>This ClientSessionId represents a client with a MAC address of 00:AF:12:34:56:DE and a session ID of 00000001.</p> <p>It should be noted that the ClientSessionId is different from the RTSP session #eld between the On Demand Client and Streaming Server. ClientSessionId is generated by the On Demand Client. The RTSP session #eld between the On Demand Client and Streaming Server is assigned by the Streaming Server and returned to the client over S3 for further stream control.</p>	

**Response Example**

```
RTSP/1.0 200 OK
CSeq: 315
Session: 47112345
OnDemandSessionId: be074250cc5a11d98cd50800200c9a66
ClientSessionId: 167af045dcfc4efba1cd09b8e0d815fb
```

**RTSP Teardown Reason Codes**

Teardown Reason Code	Description
200	User stop
201	End of stream
202	Beginning of stream
203	Pause timeout
400	Fail to tune
401	Loss of tune
402	Loss of tune
403	RTSP failure
404	Channel failure
405	No RTSP server
406	Trick-play failed
407	Internal ODA issue
408	Unknown
420	Settop Heartbeat Timeout
421	Settop Inactivity Timeout
422	Content Unavailable
423	Streaming Failure
424	QAM Failure

Teardown Reason Code	Description
425	Volume Failure
426	Stream Control Error
427	Stream Control Timeout
428	Session List Mismatch
550	Session timeout

## S1 TEARDOWN Response

### Response Format

In (quasi-)BNF the format of the response to this request shall be:

```
RTSP/ 1.0 <responseCode> SP <reasonPhrase> CRLF
1 *<header> CRLF
```

### Response Headers

If the request succeeds, the required / permitted headers are as defined in the table. If the request fails for any reason, only the CSeq: header shall be included.

Header	Format / definition	Mandatory?
CSeq	64-bit decimal integer. An exact copy of the value in the request.	Yes
Session	This request and response header field identifies an RTSP session started by the media server in a SETUP response. The session identifier is chosen by the media server. Once a client receives a Session identifier, it MUST return it for any request related to that session.  "Session" ":" session-id Session header example Session: 10000105	Yes

Header	Format / definition	Mandatory?
OnDemandSessionId	<p>The OnDemandSessionId header defines the unique session identifier that the NGOD Session Manager assigns to a given session.</p> <p>The syntax of the OnDemandSessionId header is as follows. OnDemandSessionId: &lt;onDemandSessionId&gt;where&lt;onDemandSessionId&gt; is a string representation of a 128-bit UUID as defined by IETF.</p> <p>The UUID will be constructed using the MAC address of the system on which the UUID is created to ensure global uniqueness. The OnDemandSessionId must follow the format specified in RFC 1422 except that there must be no dash (-) characters in the UUID.</p> <p>An example of the OnDemandSessionId header follows. OnDemandSessionId: be074250cc5a1d98cd50800200c9a66</p>	Yes
ClientSessionId	<p>The ClientSessionId header defines a unique session identifier created by an On Demand Client. This identifier is globally unique. ClientSessionId is used in the S1 session setup message as the client's identifier of the session.</p> <p>The syntax of a ClientSessionId header is identical to the "session-id" defined in section 3.4 of [RFC 2326]. The syntax of a ClientSessionId is a 20 character ASCII representation of a 10 byte hexadecimal value.</p> <p>The most significant 6 bytes are the client ID, the least significant 4 bytes are a session ID unique to the client. The combination of the two provides a globally unique identifier.</p> <p>An example of the ClientSessionId header follows. ClientSessionId: 00AF123456DE00000001</p>	Yes

Header	Format / definition	Mandatory?
	<p>This ClientSessionId represents a client with a MAC address of 00:AF:12:34:56:DE and a session ID of 00000001.</p> <p>It should be noted that the Client SessionId is different from the RTSP session #eld between the On Demand Client and Streaming Server. ClientSessionId is generated by the On Demand Client. The RTSP session #eld between the On Demand Client and Streaming Server is assigned by the Streaming Server and returned to the client over S3 for further stream control.</p>	

**Response Example**

```
RTSP/1.0 200 OK
CSeq: 315
Session: 47112345
OnDemandSessionId: be074250cc5a11d98cd50800200c9a66
ClientSessionId: 167af045dcfc4efba1cd09b8e0d815fb
```

**RTSP Teardown Reason Codes**

Teardown Reason Code	Description
200	User stop
201	End of stream
202	Beginning of stream
203	Pause timeout
400	Fail to tune

Teardown Reason Code	Description
401	Loss of tune
402	Loss of tune
403	RTSP failure
404	Channel failure
405	No RTSP server
406	Trick-play failed
407	Internal ODA issue
408	Unknown
420	Settop Heartbeat Timeout
421	Settop Inactivity Timeout
422	Content Unavailable
423	Streaming Failure
424	QAM Failure
425	Volume Failure
426	Stream Control Error
427	Stream Control Timeout
428	Session List Mismatch
550	Session timeout



## 6.4 Error Codes and status values

When an error is returned - either from a REST API or via RTSP - the SRM will include an error code and a status value. A client may be able to take error-specific action by checking this code.

SRM Error Code*	HTTP Response Status	RTSP Response Status	Usual status	Comment	Example
	200 OK	200	SUCCESS	Call succeeded	Successful SETUP request Successful Cable Keepalive
SRM-0001	404 Not Found	400	ERROR	Unsupported API version	
SRM-0002	400 Bad Request	400	ERROR	Malformed request or mandatory parameters missing from request	SETUP Request missing content
SRM-0002	404 Not Found	454	ERROR	Session Not Found	Cable Session Keep-Alive for non-existent session
SRM-0003	400 Bad Request	404	ERROR	Invalid / Unsupported parameter value	Content type not found
SRM-0003	500 Internal Server Error	404	ERROR	Unknown external resource	No VS endpoint

**Note**

Error is SRM-0003 for reasons of backwards

SRM Error Code*	HTTP Response Status	RTSP Response Status	Usual status	Comment	Example
					compatibility.
SRM-0004	<a href="#">#ott</a>	403	DENIED	Authorization declined	RTSP PAV errors like Invalid Smartcard/ Invalid Token for Ngod requests
SRM-0005	400 Bad Request	400	ERROR	Unsupported parameter value	Unsupported parameter value being passed in UBR reports/ Vod Failure Reports
SRM-0006	200 OK	N/A yet	DENIED	Session Limit Reached	
SRM-0007	204 OK	N/A	SUCCESS	Call succeeded. No Content	OTT Keepalive success
SRM-0008	404 Not Found	N/A	ERROR	Session Not Found	OTT Session not found for OTT Keepalive
SRM-0009	200 OK	<a href="#">#rtsp</a>	DENIED	Playout authorization refused	All OTT Pav error scenarios
SRM-0010	400 Bad Request	N/A	ERROR	Put Not Allowed	An HTTP PUT request was sent to the SRM for a session which was created without support for keep-alive.  PUT requests are only used to indicate a

SRM Error Code*	HTTP Response Status	RTSP Response Status	Usual status	Comment	Example
					session is still in use (i.e. kept-alive)
SRM-1000	N/A	502	UPSTREAM_ERROR	Insufficient resources (general)	
SRM-1001	200 OK	453	UPSTREAM_ERROR	Insufficient bandwidth	Insufficient bandwidth from an RTSP SETUP request, or the HTTP rtsp Setup via the Adapter.
SRM-1002	503	503	INTERNAL_ERROR	SRM too busy - please try later	
SRM-1003	404 Not Found	404 Not Found	UPSTREAM_ERROR	Content Not Found	Content not found on VS
SRM-2000	N/A	502	UPSTREAM_ERROR	General failure of upstream device	
SRM-2001	N/A	502	UPSTREAM_ERROR	No response from upstream device	
SRM-2002	N/A	502	UPSTREAM_ERROR	Upstream device unreachable	
SRM-2003	N/A	502	UPSTREAM_ERROR	Request NACKed by upstream device (device is reachable, but faulty)	

SRM Error Code*	HTTP Response Status	RTSP Response Status	Usual status	Comment	Example
SRM-9997	500 Internal Server Error	500	INTERNAL_ERROR	Internal data inconsistent	
SRM-9998	500 Internal Server Error	500	INTERNAL_ERROR	Unexpected internal error	
SRM-9999	500 Internal Server Error	500	INTERNAL_ERROR	General internal error	

**Note**

In some cases, a response which quotes a given error code may quote a more specific HTTP or RTSP response code, especially when the response is returned via RTSP.

The status value is one of the following.

Status	Description
ERROR	Problem with request. Repeating the request later is very likely to fail in the same way.
INTERNAL_ERROR	Problem with the SRM itself. Repeating the request later may succeed.
UPSTREAM_ERROR	Problem with a device or other system which the SRM co-ordinates with. Repeating the request later may succeed.
DENIED	Authorization declined. (This is a deliberate refusal, not a communications error or other failure in the authorization system.)

## 6.5 SRM-Adapter SETUP

### SETUP

Only used for VOD and UGC.

**Request Format**

HTTP SP <uri> SP HTTP/1.1 CRLF

**Request URI**

GET HTTP://<logicalSrmAddress>:<logicalSrmPort>/<pathString>?<uriParams> HTTP/1.1

Field	Description																														
<logicalSrmAddress>	The DNS name or IP address which the Client uses to contact the SRM-Adapter.																														
<logicalSrmPort>	The TCP port which the Client sends HTTP request to on the SRM-Adapter.																														
<pathString>	/qsp/gateway/http/js/NcPlaybackHueService/rtspSetup																														
<uriParams>	arg0=<location>&arg1=<assetName>&arg2=<assetType>&arg3=<assetDefinition>&token=<auth_token>																														
	<table border="1"> <thead> <tr> <th>Argument</th> <th>Description</th> <th>Mandatory?</th> </tr> </thead> <tbody> <tr> <td>location</td> <td>It contains the NIT and the TSID separate by a dot, i.e. NIT.TSID.</td> <td>Yes</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>NIT</th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>5 Digits</td> <td>The network ID.</td> <td>E.g. 64753</td> </tr> <tr> <td></td> <td>TSID</td> <td>3, 4 or 5 Digits</td> </tr> <tr> <td></td> <td></td> <td>Transport ID.</td> </tr> <tr> <td></td> <td></td> <td>E.g. 20101</td> </tr> </tbody> </table> </td> <td></td> </tr> <tr> <td>assetName</td> <td>The name of the content/asset, e.g. CAN00SD006532.ts</td> <td>Yes</td> </tr> <tr> <td>assetType</td> <td></td> <td></td> </tr> </tbody> </table>	Argument	Description	Mandatory?	location	It contains the NIT and the TSID separate by a dot, i.e. NIT.TSID.	Yes		<table border="1"> <thead> <tr> <th>NIT</th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>5 Digits</td> <td>The network ID.</td> <td>E.g. 64753</td> </tr> <tr> <td></td> <td>TSID</td> <td>3, 4 or 5 Digits</td> </tr> <tr> <td></td> <td></td> <td>Transport ID.</td> </tr> <tr> <td></td> <td></td> <td>E.g. 20101</td> </tr> </tbody> </table>	NIT			5 Digits	The network ID.	E.g. 64753		TSID	3, 4 or 5 Digits			Transport ID.			E.g. 20101		assetName	The name of the content/asset, e.g. CAN00SD006532.ts	Yes	assetType		
Argument	Description	Mandatory?																													
location	It contains the NIT and the TSID separate by a dot, i.e. NIT.TSID.	Yes																													
	<table border="1"> <thead> <tr> <th>NIT</th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>5 Digits</td> <td>The network ID.</td> <td>E.g. 64753</td> </tr> <tr> <td></td> <td>TSID</td> <td>3, 4 or 5 Digits</td> </tr> <tr> <td></td> <td></td> <td>Transport ID.</td> </tr> <tr> <td></td> <td></td> <td>E.g. 20101</td> </tr> </tbody> </table>	NIT			5 Digits	The network ID.	E.g. 64753		TSID	3, 4 or 5 Digits			Transport ID.			E.g. 20101															
NIT																															
5 Digits	The network ID.	E.g. 64753																													
	TSID	3, 4 or 5 Digits																													
		Transport ID.																													
		E.g. 20101																													
assetName	The name of the content/asset, e.g. CAN00SD006532.ts	Yes																													
assetType																															

Field	Description	
	The content/asset type, e.g. VOD or UGC.	Yes
asset Definition	The content/asset definition, e.g. SD	Yes
auth_token	The device's authorization token	Yes

### Response Format

```
HTTP/1.1 <responseCode> SP <reasonPhrase> CRLF
1*<header> CRLF
```

### Response Headers

Basic HTTP headers, i.e. Content-Length, Content-Type (application/json)

### Response Body

The response will carry a Json body with the following structure:

Field	Format	Description	Mandatory (+response)	Mandatory(-response)
resultCode	String	The result code from the upstream system. "0" in case of positive response.	Yes	Yes
assetUri	String	This is the name of the asset, which the session request has been sent for, e.g. CAN00SD006532.ts	Yes	N/A
result.sessionID	String	sessionId generated by SRM	Yes	
result.locationAddress	String	The DNS name or IP address which the STB should use to contact the VS. In systems where the physical VS and logical-VS / orchestrator are different, this will usually be the address of a physical VS.	Yes	

Field	Format	Description	Mandatory (+response)	Mandatory(-response)
<p><b>Note</b> As the SRM serves traffic for both VOD and Management servers for SDV sessions, the location details will be the address and port of an SRM server.</p>				
result.locationPort	String	The TCP port on which the STB should contact the VS.	Yes	
result.ServiceGroup.symbolRate	String	Is the symbol rate for the QAM channel in symbols per second; it is a 7 digit value in 4-bit BCD.	Yes	
result.ServiceGroup.modulation	String	Represents the QAM Modulation scheme used to convey the data. Possible values are 1, 2, 3, 4 or 5.	Yes	
result.ServiceGroup.innerFEC	String	Forward Error Correction (FWE) which provides error conversion rates for bad data in a transport stream. Its value ranges from 1 - 15	Yes	
resource.programNumber	String	The channel number	Yes	
resource.frequencyHertz	String	RF centre frequency. Required by the STB in order to tune: the STB will program its RF circuitry to tune to this frequency.	Yes	
resource.transportStreamID	String	The TSID value associated with this SCS session. This is used to identify the elementary stream being scrambled within the network and so the associated SCG.	Yes	
result	String	The <b>error</b> status from the upstream system, e.g. SRM	N/A	Yes

Field	Format	Description	Mandatory (+response)	Mandatory(-response)
localeMessage	String	<b>Error</b> message from the upstream system, e.g. SRM	Yes	

## 6.6 SRM-Adapter Setup Examples

### OTV\_VOD

```
GET /qsp/gateway/http/js/NcPlaybackHueService/rtspSetup?
arg0=170148375.6475320101&arg1=CAN00SD006532.ts&arg2=VOD&arg3=SD&token= HTTP/1.1
```

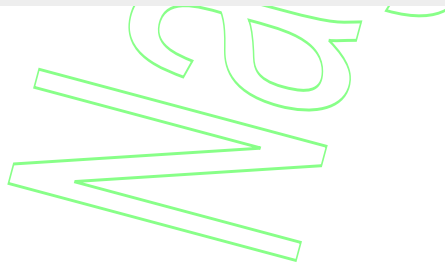
```
HTTP/1.1 200 OK
Content-Length: 392
Content-Type: application/json
"resultCode": "0",
"assetUri": "SDVideo.mpg",
"result": {
  "sessionID": "7336920069719725321",
  "locationAddress": "127.0.0.1",
  "locationPort": "554",
  "serviceGroup": {
    "symbolRate": "0070000",
    "modulation": "3",
    "innerFEC": "3"
  }
},
"resource": {
  "programNumber": "14",
  "frequencyHertz": "04100000",
  "transportStreamID": "10"
```



## UGC

```
GET /qsp/gateway/http/js/NcPlaybackHueService/rtspSetup?  
arg0=170148375.6475320101&arg1=CAN00SD006532.ts&arg2=UGC&arg3=SD&token= HTTP/1.1
```

```
vHTTP/1.1 200 OK  
Content-Length: 392  
Content-Type: application/json  
  "resultCode": "0",  
  "assetUri": "SDVideo.mpg",  
  "result": {  
    "sessionID": "1167066528740411697",  
    "locationAddress": "127.0.0.1",  
    "locationPort": "554",  
    "serviceGroup": {  
      "symbolRate": "0070000",  
      "modulation": "3",  
      "innerFEC": "3"  
    }  
  },  
  "resource": {  
    "programNumber": "11",  
    "frequencyHertz": "04100000",  
    "transportStreamID": "10"
```



## 7 User reference

### 7.1 Default users and passwords

Most of the HTTP UI requires authentication (username / password). The out-of-the-box values are as follows.

Username	Password	Purpose
operations	argan_\$tdq24	Human operator
nagra-agent	2856r4_\$tdq24	Another machine, eg. monitoring application

You can change these passwords by editing the file  
`/opt/srm/conf/tomcat-users.xml`

**Warning!** You are strongly advised to change these passwords on a production system!

**Note** The UI uses HTTP BASIC authentication. Your browser should negotiate this authentication mode automatically.

### 7.2 MongoDB write safety levels

The SRM uses MongoDB. MongoDB provides several levels of **write safety**. These levels provide different tradeoffs between performance, and the probability of **data loss** in the event that a machine fails while a write is in progress. (This is a rare event, but clearly hardware failures do happen occasionally.)

Level	Description
ERRORS_IGNORED	No safety at all - "fire and forget". Changes will be lost in the event of a network outage, or even severe network congestion. Not recommended.
UNACKNOWLEDGED	"Fire and forget", but will detect network failures. A good choice for keepalives.
ACKNOWLEDGED	Changes are sent to the DB synchronously, and will be written into it later. Changes will be lost if the DB server fails in the meantime. However, this is a rare event, and this mode of operation is fast, so it can be a fair tradeoff.

Level	Description
JOURNALED	Changes are sent to the DB synchronously, and the call does not return until the changes have been written into the on-disk journal. This is a good deal slower, but on server grade hardware data will not be lost except in the event of serious damage to the hardware.
FSYNCED	As for journal, except the master copies have also been flushed to disk. Slower still, but guards against failures of the on-disk journalling system. In most cases, this is a poor tradeoff - the extra assurance is not worth the cost.
REPLICA_ACKNOWLEDGED	The call does not return until the call has propagated to at least two replicas.
MAJORITY	The call does not return until the call has propagated to the majority of replicas.

**Note**

The levels are given in uppercase, because they must be given in uppercase when editing the [Configuration bean: mongoDB](#) <sup>p.43</sup>.

## 7.3 Ports used by the SRM

The SRM uses different ports than the SDP's internal (legacy) SRM, so that the two may be deployed on the same host. Ports used by the legacy SRM are reserved by the new SRM; this is not a conflict.

**Warning!**

The only ports which should be exposed to the outside world (ie. beyond the head-end internal LAN) are **5544/TCP** and **5544/UDP**. Firewall all the others.

You may need to route / remap additional reserved ports if you're supporting legacy STBs which use them.

**TCP**

Port	Description
5541	Management RTSP. LAN only - don't expose via firewall.
5544	Client RTSP. <b>Expose and load-balance.</b>
5580	Administration web UI. HTTP.
5581	Shutdown listener. Local-loopback only.

Port	Description
5583	<b>Reserved for future expansion.</b> (Web UI over HTTPS.)
5589	<b>Reserved.</b> JPDA access. For debugging only - leave this disabled on production systems.
8184	<b>Reserved.</b> Legacy client RTSP. Reserve on firewall / load balancer and route to 5544 on SRM.
10184	<b>Reserved.</b> Legacy client RTSP on one customer-specific branch. May be needed for compatibility during switchover.

**UDP**

Port	Description
5540	<b>Reserved.</b> Legacy VS ANNOUNCE. Route to port 5544/UDP.
5544	<b>Reserved for future expansion.</b> (Compatibility VS ANNOUNCE.)
8184	<b>Reserved for future expansion..</b> Legacy VS ANNOUNCE. Route to 5544/UDP.

## 7.4 QAM modulation codes

The modulation code describes the QAM encoding level.

The SRM stores modulation codes as integers. There are two codes for each encoding level, a standard code based on the numeric part of the encoding name, and an alternative code based on a logarithmic scale.

The two codes for each encoding level are equivalent when ingesting - either can be used. The standard code is stored internally. When the value is sent to other systems it is converted to the appropriate form.

Encoding	Modulation code	Alternative (log) modulation code
QAM-16	16	1
QAM-32	32	2

Encoding	Modulation code	Alternative (log) modulation code
QAM-64	64	3
QAM-128	128	4
QAM-256	256	5

## 7.5 Session contributors

**Caution!**

This is lower-level technical detail. It may be useful to know this information when reading SRM's log files, but not apart from that.

Logically, the SRM sets up a session by passing it to a set of **session contributors**, in order. These session contributors are abstractions - they are internal components within the SRM itself.

The session contributors are as follows. The order shown is the order used during session setup. The order is different during teardown, and not all contributors are called.

Contributor	Resource it contributes
NGOD PAV resolver	Resolves the purchase-token UUID quoted by the client. Retrieves the playlist and content-usage information from the SDP.
Session creator	Initializes the session internally.
ERM sequencer	Allocates QAM bandwidth, by calling the ERM which controls the QAM. The "sequencer" part arises because in some deployments, an STB may be able to access multiple QAMs; these in turn may be managed by different ERMs (and for full failover support, they should be.) In such a case, the SRM iterates through all the suitable QAMs until it finds one which has available bandwidth. Where a QAM uses <a href="#">Table-based bandwidth allocation</a> <sup>p.13</sup> , the "ERM" is a virtual ERM inside the SRM. Otherwise, the ERM is a physical device.
STS sequencer	Prepares a streamer to play back the content, by calling the streamer. The "sequencer" bit arises because in some deployments, there may be several streamers which can serve the content, providing failover. In such a case, the SRM iterates through all the candidate

Contributor	Resource it contributes
Content usage reporter	streamers until it finds one which can provide the stream.
Client announcer	During teardown, causes an ANNOUNCE message to be sent back to the client, if the client has a keepalive connection open to receive it. The keepalive connection is automatically closed.
Disconnecter tagger	Once a session has been set up, the disconnecter tagger applies additional tags to it, which allow it to be subject to force-disconnect; during teardown, it removes the same tags. This is done as a separate step in order to manage failed teardowns cleanly.
<b>Note</b>	When the session is first torn down, the disconnecter tags are removed first. If the teardown subsequently fails because a <a href="#">Contributor device</a> <sup>p.10</sup> is unavailable, then the teardown will automatically continue once the contributor device comes back; there is no point in attempting to force-disconnect it a second time, because the teardown will fail in exactly the same way. (Worse, if a second force-disconnect happened to coincide with the affected device re-appearing, then two different parts of the system would be trying to tear down the same session simultaneously; each would disrupt the other.)

## 8 Puppet Modules

### 8.1 Exposing SRM Operator API

**Note** | Since SRM 4.0.5

#### Overview

In the past, the operator APIs could only be accessed directly via SRM. Now, it is possible to allow/deny an operator from making operator API calls to the SRM via nginx. This can be easily done by configuring the srm-nginx configuration in foreman.

**Note** | Currently, this is only allowed for /srm/force-disconnect API.

#### Allow/deny Operator API via nginx

In foreman, modify your smart class variable to allow/ deny all or a range of IPs. This can be done by clicking on the parameter itself. Description and further information will be available under the smart class variable.

#### Overall Flow Diagram

Operator should follow this use case to:

- ▶ Configure srm-nginx-private in foreman to allow/deny all or a range of IPs.
- ▶ SETUP an OTT session on the SRM
- ▶ Send an an Operator API request via nginx

Reference	<TBD>
Main actor	srm-nginx
Secondary actors	PAV, Client application, NGINX, Operator
Pre-Conditions	- SRM and PAV are running and operational - Operator ingested SrmClient into SRM with correct Client data.

- srm and srm-nginx classes have been added and configured correctly in Foreman for your node.

Trigger

Operator sends Operator API request (force-disconnect) to srm via nginx.

## Operator API Steps

1. Operator configures srm-nginx in foreman to allow Operator API requests.
2. Client sets up an OTT session on SRM
3. Operator send an operator API request, i.e. force-disconnect, via nginx
4. Nginx checks if this resource and IP address are allowed
5. Nginx directs the request to the right SRM node and port
6. SRM tears down the session and returns 200 OK
7. Session does not exist in SRM DB any longer

## Operator API Extensions

Ref	From step	Description
A	4	If resource or IP are denied, then nginx returns an error to the Operator.



## 9 Release Notes

Nagra

## 10 Release Notes: 3.0

### 10.1 SRM 3.0STD1

#### Session Resource Manager Release 3.0STD1

This section details any information which may be relevant since the previous 3.0STD1 release.

##### Highlights

- ▶ Inclusion of PAV support for OTT
  - Inclusion of session limitation checking against newer style account profile
  - Does not include session limitation checking against older Service Provider based method
- ▶ Inclusion of Operator APIs for Session Discovery
- ▶ Inclusion of Operator APIs for Forced Teardown

##### Upgrade Notes

##### Configuration Changes

The following table illustrates any changes to the SRM configuration

Param	Change Description	Old Default	New Default

##### SRM Ingest Changes

These are the changes within the [Data ingest reference](#) <sup>p.63</sup>

- ▶ Update to the PAV interface
  - Version: 2.7
  - TcpPort: 80
  - Address: localhost
  - Status: ACTIVE
  - Name: loopback
  - UID: Pav\_loopback

##### Mongodb Config Changes

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**MLM Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
----------	--------	---------

**Open Issues**

Issue No	Status	Summary
----------	--------	---------

## 10.2 SRM 3.0STD2

### Session Resource Manager Release 3.0STD2

This section details any information which may be relevant since the previous 3.0STD1 release.

**Highlights**

- ▶ Update of PAV support for OTT
  - Inclusion of session limitation checking against older Service Provider based method
- ▶ Inclusion of session keep-alive support for OTT

**Upgrade Notes**

**Configuration Changes**

The following table illustrates any changes to the SRM configuration

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**SRM Ingest Changes**

These are the changes within the [Data ingest reference](#) <sup>p.63</sup>

- ▶ Update to the PAV interface
  - Version: 2.7
  - TcpPort: 80
  - Address: localhost
  - Status: ACTIVE
  - Name: loopback
  - UID: Pav\_loopback

**Mongodb Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**MLM Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

- ▶ New Service entry in srm\_moduke\_1.cfg
  - SRM Inbound OTT SETUP request range, and request range detail

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
98179	Implemented	Invalid status return on session setup when contentType is not provided in json payload
98182	Implemented	Invalid status return on session setup when contentType is invalid

**Open Issues**

Issue No	Status	Summary
98188	Analysed	SRM operator API are only available using a direct access to one SRM. Requirements need to be defined.

## 10.3 SRM 3.0STD3

### Session Resource Manager Release 3.0STD3

This section details any information which may be relevant since the previous 3.0STD2 release.

**Highlights**

- ▶ Extension in puppet to allow DB configuration

**Upgrade Notes**

**Configuration Changes**

The following table illustrates any changes to the SRM configuration

Param	Change Description	Old Default	New Default
sdmDBName	The name of the mongo db instance to use	N/A	srm
srmDbHostname	DB name of SRM database		localhost
srmDbAdminUserName	Admin username of SRM database		admin
srmDbAdminPassword	Password for admin user of SRM database		nagra
srmDbNormalUserName	Operational username of SRM database		operations

Param	Change Description	Old Default	New Default
srmDbNormalPassword	Password for operational user of SRM database		argan
srmDbMaxConnections	Max connections to SRM database		250
srmDbMaxWaitersPer Connection	Max waiters per connection to SRM database		1
srmDbInitialConnect TimeoutMillis	Initial timeout for connection to SRM database in milliseconds		1000
srmDbWaitTimeoutMillis	Wait timeout for SRM database in milliseconds		2000
srmDbIoTimeoutMillis	IO timeout for SRM database in milliseconds		5000
srmDbAllowSlaveReads	Allow slave reads policy of SRM database		FALSE
srmDbKeepaliveWrite Safety	Keep alive write safety policy of SRM database		ERRORS
srmDbDefaultWriteSafety	Default write safety policy of SRM database		ACKNOWLEDGED
srmDbExtraWriteSafety	Extra write safety policy of SRM database		JOURNALED
srmDbDropOnStartup	Drop on startup policy of SRM database		NO

The core-conf.xml file has been puppetised so can now be configured via foreman

**SRM Ingest Changes**

None

**Mongodb Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**MLM Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

None

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
----------	--------	---------

**Open Issues**

Issue No	Status	Summary
----------	--------	---------

**Configuration Summary**

[Configuration bean: mongoDB](#) <sup>p.43</sup>

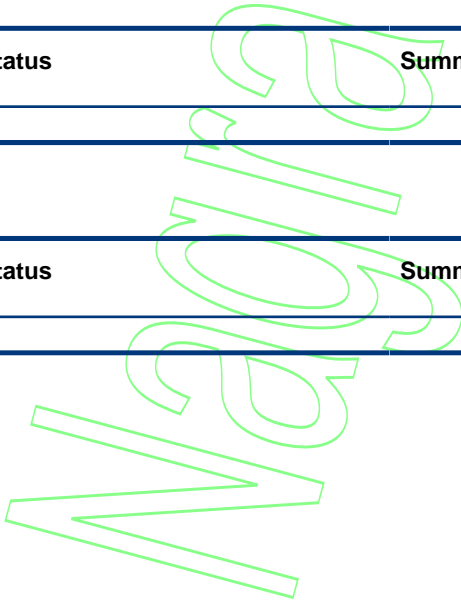
[Configuration bean: bodcmConfig](#) <sup>p.42</sup>

[Configuration bean: srmCoreConfig](#) <sup>p.48</sup>

[Configuration bean: ermConfig](#) <sup>p.46</sup>

[Configuration bean: vsConfig](#) <sup>p.58</sup>

[Configuration bean: sessionConfig](#) <sup>p.47</sup>



## 11 Release Notes: 3.1

### 11.1 SRM 3.1STD1

#### Session Resource Manager Release 3.1STD1

This section details any information which may be relevant since the previous 3.x release

##### Highlights

- ▶ Inclusion of RTSP support for Numericable variants of OTV
  - This version includes Switched Digital Video Support
  - Integration with Cisco ERM for resource management
  - Support for third-party bandwidth-only resource allocation via the cablelabs interfaces for both Bytel and Playcast
- ▶ This release has an issue with VOD support against the legacy Numericable OpenTV client VOD application which prevents validation via the STB

##### Upgrade Notes

##### Configuration Changes

The following table illustrates any changes to the SRM configuration (in srm-conf.xml)

Param	Change Description	Old Default	New Default
sdvVsSrmIpAddress	Address of the SRM which should listen for SDV keep-alive requests. Generally this should be the load balancer	N/A	127.0.0.1
sdvVsSrmPort	Port of the SRM which should listen for SDV keep-alive requests. Generally this should be the load balancer	N/A	80

##### SRM Ingest Changes

There are no specific changes. However because this release is geared to an OTV variant and third-party support it will be necessary to ensure the relevant clients are ingested.

##### Mongodb Config Changes



Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**MLM Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
----------	--------	---------

**Open Issues**

Issue No	Status	Summary
----------	--------	---------

## 11.2 SRM 3.1STD2

### Session Resource Manager Release 3.1STD2

This section details any information which may be relevant since the previous 3.1 release

**Highlights**

- ▶ Inclusion of RTSP support for Numericable variants of OTV
  - This version includes Switched Digital Video Support
  - Integration with Cisco ERM for resource management
  - Support for third-party bandwidth-only resource allocation via the cablelabs interfaces for both Bytel and Playcast
  - Inclusion of VOD support
- ▶ No PAV integration to non-ML SDP releases
- ▶ No SRM/USRM Resynchronization.

**Upgrade Notes**

**Configuration Changes**

The following table illustrates any changes to the SRM configuration (in srm-conf.xml)

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**SRM Ingest Changes**

There are no specific changes. However because this release is geared to an OTV variant and third-party support it will be necessary to ensure the relevant clients are ingested.

**Mongodb Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**MLM Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
----------	--------	---------

**Open Issues**

Issue No	Status	Summary
----------	--------	---------

**11.3 SRM 3.1STD3**

**Session Resource Manager Release 3.1STD3**

This section details any information which may be relevant since the previous 3.1 release

**Highlights**

- ▶ Inclusion of PAV integration with SDP (see below for notes on supporting legacy customers)
- ▶ SRM/USRM Resync Job
- ▶ Inclusion of UGC Support  

```
<property name="pavUriPrefix" value="/ws-gateway" />
```

**Upgrade Notes**

**Configuration Changes**

The following table illustrates any changes to the SRM configuration (in srm-conf.xml)

Param	Change Description	Old Default	New Default
pavUriPrefix	Ability to call legacy SDP PAV	N/A	/api

**SRM Ingest Changes**

There are no specific changes. However because this release is geared to an OTV variant and third-party support it will be necessary to ensure the relevant clients are ingested.

**Mongodb Config Changes**

Param	Change Description	Old Default	New Default

**MLM Config Changes**

Param	Change Description	Old Default	New Default

**Issues List**

**Fixed Issues**

Issue No	Status	Summary

**Open Issues**

Issue No	Status	Summary
----------	--------	---------

## 11.4 SRM 3.1STD4

### Session Resource Manager Release 3.1STD4

This section details any information which may be relevant since the previous 3.1 release

#### Highlights

- Inclusion of SCS

#### Upgrade Notes

#### Configuration Changes

The following table illustrates any changes to the SRM configuration (in srm-conf.xml)

Param	Change Description	Old Default	New Default
pavUriPrefix	Ability to call legacy SDP PAV	N/A	/api
srmScsChannelId	SCS Channel ID	N/A	0x0001
connectTimeoutSecs	SCS connection timeout.	N/A	15
requestTimeoutSecs	SCS request timeout	N/A	30
throttleTimeoutSecs	SCS throttle timeout	N/A	30
keepaliveIntervalSecs	SCS keepalive interval	N/A	60
keepaliveTimeoutSecs	SCS keepalive request timeout	N/A	15
concurrentRequestLimit	SCS concurrent request limit	N/A	500

Param	Change Description	Old Default	New Default
clusterSyncIntervalSecs	SCS resync job - interval period (seconds)	N/A	1800

**SRM Ingest Changes**

There are no specific changes. However because this release is geared to an OTV variant and third-party support it will be necessary to ensure the relevant clients are ingested.

**Mongodb Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**MLM Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
----------	--------	---------

**Open Issues**

Issue No	Status	Summary
----------	--------	---------

**11.5 SRM 3.1STD5**

**Session Resource Manager Release 3.1STD5**

This section details any information which may be relevant since the 3.1STD3 release

**Highlights**

- Inclusion of SCS

**Upgrade Notes**

**Configuration Changes**

The following table illustrates any changes to the SRM configuration (in srm-conf.xml)

Param	Change Description	Old Default	New Default
pavUriPrefix	Ability to call legacy SDP PAV	N/A	/api
srmScsChannelId	SCS Channel ID	N/A	0x0001
connectTimeoutSecs	SCS connection timeout.	N/A	15
requestTimeoutSecs	SCS request timeout	N/A	30
throttleTimeoutSecs	SCS throttle timeout	N/A	30
keepaliveIntervalSecs	SCS keepalive interval	N/A	60
keepaliveTimeoutSecs	SCS keepalive request timeout	N/A	15
concurrentRequestLimit	SCS concurrent request limit	N/A	500
clusterSyncIntervalSecs	SCS resync job - interval period (seconds)	N/A	1800

**SRM Ingest Changes**

There are no specific changes. However because this release is geared to an OTV variant and third-party support it will be necessary to ensure the relevant clients are ingested.

**Mongodb Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**MLM Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

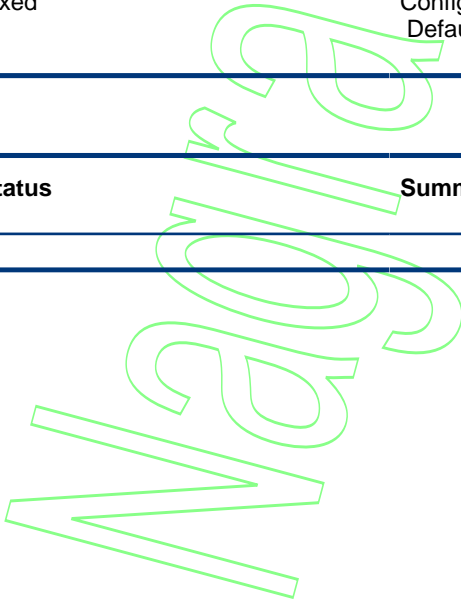
**Issues List**

**Fixed Issues**

Issue No	Status	Summary
100961	Fixed	Incorrect Cablelabs SETUP response.
100842	Fixed	Config for BODCM reconnection. Defaulted to 30

**Open Issues**

Issue No	Status	Summary
----------	--------	---------



## 12 Release Notes: 3.2

### 12.1 SRM 3.2STD0

#### Session Resource Manager Release 3.2STD0

This section details any information which may be relevant since the previous 3.1 release

##### Highlights

- ▶ Alpha version of Bandwidth Release API
- ▶ Bugfix release

##### Upgrade Notes

##### Configuration Changes

The following table illustrates any changes to the SRM configuration (in srm-conf.xml) for the scs-config bean.

Param	Change Description	Old Default	New Default
srmScsChannelId	Channel ID to use from this SRM instance	N/A	0x0001
connectTimeoutSecs	Connection timeout to the SCS (seconds)	N/A	15
requestTimeoutSecs	Request timeout to the SCS (seconds)	N/A	30
throttleTimeoutSecs	Throttle Timeout (seconds)	N/A	60
keepaliveTimeoutSecs	Keepalive period between SRM and SCS (seconds)	N/A	15
concurrencyRequestLimit	Max concurrent requests	N/A	500
clusterSyncIntervalSecs	SCS resync period (seconds)	N/A	1800

##### SRM Ingest Changes



There are no specific changes.

**Mongodb Config Changes**

Param	Change Description	Old Default	New Default

**MLM Config Changes**

Param	Change Description	Old Default	New Default

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
100832	Fixed	Teardown sent by SRM to VS following ANNOUNCE
100961	Fixed	Response to Bytel SETUP is malformed
101076	Fixed	ERM Resource Hung
101078	Fixed	No PAV for SDV
101136	Config	SDV Setup not working
101138	Fixed	VOD resources are no more released
101140	Config	Cannot purchase content on the legacy portal anymore
101181	Fixed	SRM Doesn't teardown session on VS after termination from ERM

Open Issues

Issue No	Status	Summary

## 12.2 SRM 3.2STD1

### Session Resource Manager Release 3.2STD1

This section details any information which may be relevant since the previous 3.2 release

**Highlights**

- ▶ Reserve Bandwidth API
- ▶ Release Bandwidth API
- ▶ RTSP SETUP extensions to support streaming over pre-reserved bandwidth

**Upgrade Notes**

**Configuration Changes**

The following table illustrates any changes to the SRM configuration (in rtsp-in-gateway.xml) for the otv2\_4config bean.

Param	Change Description	Old Default	New Default
otvFirstConfig	Support for OTV first config in the response handling.	N/A	true
useBCDFormat	Enable the symbol-rate and bitrate to use the BCD format	N/A	false
suppressVsTimeout	Do not display the vs timeout value in the SESSION header response.	N/A	false

The following table illustrates changes to the vs configuration (in srm-conf.xml) for the vsConfig bean.

Param	Change Description	Old Default	New Default
vsClientFormat	Format of the client string sent from the SRM to the VS.	N/A	TSID.SERVICE_GROUP. SMARTCARD_ID

**SRM Ingest Changes**

There are no specific changes.

**Mongodb Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**MLM Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
100833	Fixed	Request to VS from the SRM contains a modified client-id
101173	Fixed	Response to the STB should not contain timeout field
101349	Fixed	SRM sends an ANNOUNCE to Third-Party following a regular teardown
101283	Fixed	Added "453" to the default VS node failover list.

**Open Issues**

Issue No	Status	Summary
----------	--------	---------

## 12.3 SRM 3.2STD2

### Session Resource Manager Release 3.2STD2

This section details any information which is relevant since the previous 3.2 release

#### Highlights

- Bugfix release

#### Upgrade Notes

#### Configuration Changes

The following table illustrates changes to the vs configuration (in srm-conf.xml) for the srmCoreConfig bean.

Param	Change Description	Old Default	New Default
announceOutboundThirdpartyTimeoutSecs	Time to wait when the SRM sends an ANNOUNCE message to a Cablelabs client.	N/A	5

#### SRM Ingest Changes

There are no specific changes.

#### Mongodb Config Changes

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

#### MLM Config Changes

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

#### Issues List

**Fixed Issues**

Issue No	Status	Summary
101382	Fixed	SCS in INACTIVE state continues to provision SCS sessions
101265	Fixed	SRM: resource not released if teardown fails on SCS
101174	Fixed	SRM does not teardown ghost session after ANNOUNCE 6004 from USRM
101139	Fixed	During ERM resync job, PING are sent on inexistent sessions
101176	Fixed	SRM reconnects to USRM whereas USRM has not been disconnected
101260	Fixed	Bytel: resource not released if no response to the ANNOUNCE sent by the SRM

**Open Issues**

Issue No	Status	Summary
----------	--------	---------

## 12.4 SRM 3.2STD3

### Session Resource Manager Release 3.2STD3

This section details any information which is relevant since the previous 3.2 release

**Highlights**

- ▶ Bugfix release
- ▶ 4K Support

**Upgrade Notes**

**Configuration Changes**

The following table illustrates changes to the vs configuration (in srm-conf.xml) for the srmCoreConfig bean.

Param	Change Description	Old Default	New Default
bandwidthMap_4K	Attribute for specifying the bitrate of 4K content. Note that the values within the SDP configuration will no longer be used and are superseded by those in the SRM.  The next release will include support for configurable definitions which will allow deployments to configure any number of definitions (not just SD/HD/3D/4K)	N/A	5

**SRM Ingest Changes**

There are no specific changes.

**Mongodb Config Changes**

Param	Change Description	Old Default	New Default

**MLM Config Changes**

Param	Change Description	Old Default	New Default

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
101885	Fixed	SDV Sessions are cleared from the USRM but not the SRM when the USRM is offline

Issue No	Status	Summary
No PR	Fixed	"Notice" header from SRM to third-party client should have a lower case n, i.e. "notice"
No CR	Implemented	Supported Bandwidth should include 4K.
No PR	Fixed	Fixed support for SCS in a clustered environment. IGNORE_CHANNEL_ID
101345	Fixed	ANNOUNCE 6002 not required following an TEARDOWN from third-party

**Open Issues**

Issue No	Status	Summary

## 12.5 SRM 3.2STD4

### Session Resource Manager Release 3.2STD4

This section details any information which is relevant since the previous 3.2 release

**Highlights**

- ▶ Bugfix release

**Upgrade Notes**

**Configuration Changes**

The following table illustrates changes to the vs configuration (in srm-conf.xml) for the srmCoreConfig bean.

Param	Change Description	Old Default	New Default
ResponseCodeFailover List	The format of this within foreman has changed - the defaults are an example of the change.	<value>500</value><value>404</value>	500,404
BandwidthMap	This is a new property to replace the previous, fixed, definitions.  I.e. Previously the SRM was restricted to SD/HD/3D/4K unless manual modifications of the config were made.  It is now possible to configure custom definition/bitrate maps.	Not Previously configured via Foreman	HD:11250, SD:3750, THREE_D:15000, 3D:15000, 4K:16000

**SRM Ingest Changes**

There are no specific changes.

**Mongodb Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**MLM Config Changes**

Param	Change Description	Old Default	New Default
-------	--------------------	-------------	-------------

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
102113	Fixed	UGC Doesn't work any more - caused by a regression in the SRM Pav interface.



Issue No	Status	Summary
102158	Fixed	Bitrate mapping needs to be managed from within the SRM instead of the SDP - corrected within Foreman
102156	Fixed	"Notice" header from SRM to third-party client should have a lower case n, i.e. "notice"
102155	Fixed	Issues in clustered jobs.

**Open Issues**

Issue No	Status	Summary
----------	--------	---------

## 12.6 SRM 3.2STD5

### Session Resource Manager Release 3.2STD5

This section details any information which is relevant since the previous 3.2 release

**Highlights**

- ▶ Bugfix release
- ▶ Extended logging
- ▶ Updated documentation with details on range and SCS creation/management.

**Upgrade Notes**

**Configuration Changes**

The following table illustrates changes to the core configuration (in srm-conf.xml) for the srmCoreConfig bean.

Param	Change Description	Old Default	New Default
maxContentLengthInBytes	Enables the maximum support content length for requests to be changed	N/A	8192

**SRM Ingest Changes**

There are no specific changes.

**Mongodb Config Changes**

Param	Change Description	Old Default	New Default

**MLM Config Changes**

Param	Change Description	Old Default	New Default

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
102557	Fixed	STS resets connection with SRM. This fix requires a manual configuration change to the action-matrix.
102558	Fixed	SRM reset connection with USRM after a GET_PARAMETER
102560	Implemented	Logging extensions to provide additional logging, at DEBUG and INFO level <ul style="list-style-type: none"> <li>▶ Log at DEBUG all outbound RTSP requests and responses.</li> <li>▶ Log at INFO any exception being returned from the SRM, such as error responses back to the SRM-adapter.</li> <li>▶ Log at DEBUG when an inbound VS ANNOUNCE is received.</li> <li>▶ Log at INFO the session ID for the VS ANNOUNCE when received</li> </ul>

**Open Issues**

Issue No	Status	Summary

## 12.7 SRM 3.2STD6

### Session Resource Manager Release 3.2STD6

This section details any information which is relevant since the previous 3.2 release

**Highlights**

- ▶ Fix for CORS headers in nginx

**Upgrade Notes**

**Configuration Changes**

None

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
102730	Fixed	Issue whereby same HTTP header was being added via the handleApi and CORS config.

**Open Issues**

Issue No	Status	Summary

## 12.8 SRM 3.2STD7

## Session Resource Manager Release 3.2STD7

This section details any information which is relevant since the previous 3.2 release

### Highlights

- ▶ Bugfix Release

### Upgrade Notes

### Configuration Changes

None

### Issues List

#### Fixed Issues

Issue No	Status	Summary
102911	Fixed	No keepalive between SRM and streamers
102912	Fixed	Duplicate SDV sessions - Caused by lock based on Network ID

#### Open Issues

Issue No	Status	Summary
----------	--------	---------

## 12.9 SRM 3.2STD8

## Session Resource Manager Release 3.2STD8

This section details any information which is relevant since the previous 3.2 release

### Highlights

- ▶ Bugfix Release

### Upgrade Notes

### Configuration Changes

None

### Issues List

**Fixed Issues**

Issue No	Status	Summary
102995	Fixed	SRM UDP listener closes whenever there is an error processing a UDP request
103001	Fixed	SRM issue with some frequencies in the SRM response to the STB RTSP Setup

**Open Issues**

Issue No	Status	Summary

**12.10 SRM 3.2STD9**

**Session Resource Manager Release 3.2STD9**

This section details any information which is relevant since the previous 3.2 release

**Highlights**

- ▶ Bugfix Release

**Upgrade Notes**

**Configuration Changes**

None

**Issues List**

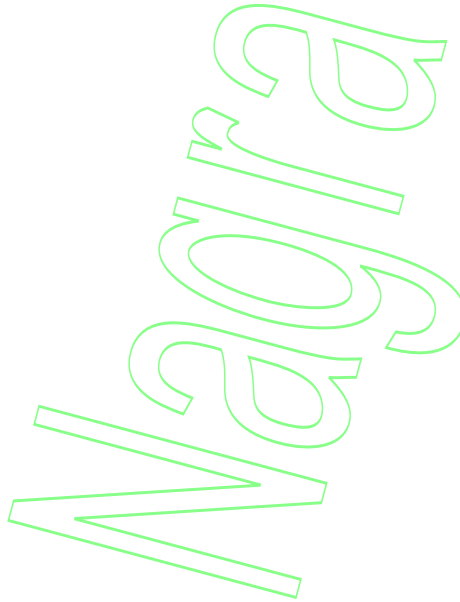
**Fixed Issues**

Issue No	Status	Summary
103173	Fixed	SCS device is intermittently set to FAULT by the SRM

Issue No	Status	Summary
103171	Fixed	SRM Sessions are not cleared if it was not possible to reserve bandwidth for a Clabs client

**Open Issues**

Issue No	Status	Summary
----------	--------	---------



## 13 Release Notes: 4.0

### 13.1 SRM 4.0STD1

#### Session Resource Manager Release 4.0STD1

This section details any information which is relevant since the previous 3.2STD9 release

##### Highlights

- ▶ Bugfix Release
- ▶ Extended logging for VAST requested as INFO level
- ▶ Support for CiscoQTVS Streamers

##### Upgrade Notes

##### Configuration Changes

The following table illustrates changes to the core configuration (in core-conf.xml) for the mongoDB bean.

Param	Change Description	Old Default	New Default
port	Inclusion to enable configuration of the port of the DB to connect to; this defaults to 4230 ready for use with Mongo 3.0 instances.	N/A	4230

##### Ingest changes

The inclusion of support for CiscoQTVS streamers means that from now on a specific CodeId of "CiscoQTVS" will need to be specified when ingesting a device of this type.

##### Issues List

##### Fixed Issues

Issue No	Status	Summary
103472	Fixed	NagraVS keepalive requests not working
103893	Fixed	Teardown queue not always cleared

**Change Requests**

Request No	Status	Summary
103912	Implemented	SRM - align logs for a VAST request with regular VOD request.

**Open Issues**

Issue No	Status	Summary
----------	--------	---------

**13.2 SRM 4.0STD3**

**Session Resource Manager Release 4.0STD3**

This section details any information which is relevant since the previous 4.0STD1 release. Please note 4.0STD2 was not released.

**Highlights**

- ▶ Bugfix Release
- ▶ Addressed issue in connecting to mongo cluster

**Upgrade Notes**

**Configuration Changes**

None

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
104917	Fixed	SRM is not able to connect to a SECONDARY MongoDB instance

**Change Requests**



Request No	Status	Summary
103242	Implemented	Extended logging to include SRM request processing time.

**Open Issues**

Issue No	Status	Summary
----------	--------	---------

### 13.3 SRM 4.0STD4

#### Session Resource Manager Release 4.0STD4

This section details any information which is relevant since the previous 4.0STD3 release.

**Highlights**

- ▶ Added KBRO User Session Limitation Feature

**Upgrade Notes**

**Configuration Changes**

The following table illustrates any changes to the SRM configuration (in srm-conf.xml).

Param	Change Description	Old Default	New Default
userUidTokenIndex	The position within the token of the user uid. This is configurable to avoid changes to the SRM if the token entries are amended/re-ordered.  This can be changed on the srm class in foreman.	N/A	6
<b>EnableUserLimitation</b>	This flag is ingested as part of the SrmClient. It enables/disabled the entire user limitation feature.	N/A	false

Param	Change Description	Old Default	New Default
UserSessionLimit	This flag is ingested as part of the SrmClient. It defines the limits of concurrent sessions a user is allowed by SRM in EnableUserLimitation flags is enabled	N/A	0

**Issues List**

**Fixed Issues**

Issue No	Status	Summary

**Change Requests**

Request No	Status	Summary
103968	Implemented	Session Limit Control Down To User Level

**Open Issues**

Issue No	Status	Summary

## 13.4 SRM 4.0STD5

### Session Resource Manager Release 4.0STD5

This section details any information which is relevant since the previous 4.0STD4 release.

**Highlights**

- ▶ Fixed the Teardown (force-disconnect) logic to use correct "sessionType/sessionVariant". This fix will ensure correct contributors are used from the action-matrix.

- ▶ Extracted the srm nginx mlds logic out of the srm-private.pp class and into a separate mlds class, i.e. srm-nginx-private.pp. This allows the deployments of srm and nginx on separate nodes.
- ▶ Added further configuration and logic to the srm-nginx-private.pp to allow an operator to hide/ expose force-disconnect operator API.

**Upgrade Notes**

**Configuration Changes**

The following table illustrates any changes to the srm-nginx configuration in Foreman. This will be part of the srm::srm-nginx-private mlds class.

Param	Change Description	Old Default	New Default
nginx_allow_operator	Double-quoted, comma-separated list of all permitted IPs to access the SRM operator APIs, e.g. ['10.10.10.10', '10.8.8.8/8']. To allow all, set the value to ['all']. This is configured in handle OperatorAPI.conf under nginx.  <b>WARNING</b> - Currently, only <b>force-disconnect</b> operator API is permitted for use through nginx in this release.	N/A	[]
nginx_deny_operator	Double-quoted, comma-separated list of all denied IPs from accessing the SRM operator APIs, e.g. ['10.10.10.10', '10.8.8.8/8']. To deny all, set the value to ['all']. This is configured in handle OperatorAPI.conf under nginx.	N/A	[]
ensureversion	Version of the srm-nginx config that should be installed. Can be 'present', 'latest', or a specific version (eg 2.6.7-1). Can also be 'absent' if you want to remove the package and associated resources.	N/A	"present"

Param	Change Description	Old Default	New Default
	Absent will remove your data directory.		
srmHost	SRM host IP address	N/A	"127.0.0.1"
srmOttPort	SRM host OTT listening port	N/A	"5588"
srmOperatorPort	SRM host Operator listening port	N/A	"5580"

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
IST [98188]	Fixed	operator API are only available using a direct access to one SRM
IST [105616]	Fixed	No workflow configured forVOD/QTV

**Change Requests**

Request No	Status	Summary

**Open Issues**

Issue No	Status	Summary

## 14 Release Notes: 5.0

### 14.1 SRM 5.0STD0

#### Session Resource Manager Release 5.0STD0

This section details any information which is relevant since the previous 4.0STD5 release.

##### Highlights

- ▶ Fixed the SDV locking mechanism. This fix will ensure that multiple SDV teardown requests from a mis-behaving STB does not remove a ChannelResource that is being used by other STBs.
- ▶ Rework of teardown queue to avoid duplicate entries

##### Upgrade Notes - 4.0STD5 to 5.0STD0

##### Terminate Legacy SRM Module

###### 1. Existing Configuration Backup

- ▶ Ensure all configuration under '/opt/srm/webapps/srm/WEB-INF/' is backed up.
- ▶ Ensure you have a screenshot of your srm configuration class in foreman.
- ▶ Ensure all srm-nginx conf is backed up.

###### 2. Disable SDV feature and wait for all SDV sessions to clear out.

3. Any remaining ChannelResource will have to be cleared out appropriately to ensure no blocking resources on the USRM.

###### 4. Uninstallation of Legacy SRM

- ▶ Stop srm service on all nodes, i.e. **service srm stop.**
- ▶ In foreman, change the ensureversion parameter for the srm class to 'absent'.
- ▶ kick puppet off on the srm nodes. This will remove the srm-tomcat and srm-deployer packages.

##### Data Migration

##### Note

Instructions on how to run each of these scripts are found within the scripts.

1. Add TenantName field to srmClient and MasterSession's srmClient. This is done using the following script:

multi-tenant-migration-script.js

2. Upgrade PAV collection structure data. This is done using the following script:

pav\_migration-script.js

##### Install Latest SRM

1. Upload SRM rpms to repo

2. Install SRM MLDS provided with the release
3. Ensure SRM MLDS classes are still configured correctly
  - ▶ Set 'defaultTenantName' to blank for the srm class in foreman.
  - ▶ Set 'ensureversion' back to latest
4. Ensure action-matrix is configured correctly
5. Deploy SRM
  - ▶ Kick puppet off, i.e. puppet agent -t

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
IST [106348]	Fixed	SDY resource released even if a second STB is using channel

**Change Requests**

Request No	Status	Summary
------------	--------	---------

**Open Issues**

Issue No	Status	Summary
----------	--------	---------

## 14.2 SRM 5.0STD1

### Session Resource Manager Release 5.0STD1

This section details any information which is relevant since the previous 5.0STD0 release.

**Highlights**

- ▶ Added a flexibility in SRM to allow an operator to **Increase Number of available service Groups on Network**.
- ▶ Fixed the srm-client re-ingestion issue with blank tenant.
- ▶ Removed a non-usable parameter from the srm-mlds parameters.csv file.

- ▶ Fixed the SDV ChannelResource teardown issue, which rises as a result of a failed setup with USRM
- ▶ Inbound RTSP setup/teardown endpoint for allocating resources, with optional short-term catch-up parameters, and ability to set any number of ordered playlist items.
- ▶ Ericsson VSPP outbound video RTSP setup/teardown calls for bandwidth allocation/release.

**Upgrade Notes**

**Configuration Changes**

The following table illustrates any changes to the configuration in Foreman.

Param	Change Description	Old Default	New Default
isDockerised	This parameter has been removed from the parameters.csv file. This should not appear in foreman anymore. This parameter was never used anywhere in the srm puppet class or manifests.	N/A	N/A

**Configuration Changes for Deployment with SAGEM and VSPP**

The following table illustrates all configuration changes that must be made in production in Foreman when SAGEM and VSPP are used in the SETUP/TEARDOWN workflow.

Param	DefaultValue	NewValue
otvFirstConfig	true	false
suppressVsTimeout	false	true

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
IST [107543]	Fixed	Re-ingesting srm-client with blank tenantName fails on the mongoDB dues to conraint violation.
IST [107531]	Fixed	Missing parameter "isDockerised" causes installation to fail with puppets

Issue No	Status	Summary
IST [107712]	Fixed	SDV ChannelResrouce is not tearing down when first setup fails with USRM

**Change Requests**

Request No	Status	Summary
IST[106390]	Fixed	Increase Number of available service Groups on Network.

**Open Issues**

Issue No	Status	Summary
----------	--------	---------

**Feature List**

Feature No	Status	Summary
E-10828	Implemented	Inbound and outbound calls completed in relation to the VSPP workflow

## 14.3 SRM 5.0STD2

### Session Resource Manager Release 5.0STD2

This section details any information which is relevant since the previous 5.0STD1 release.

**Highlights**

- ▶ Removed validation from the 'ordinal' and 'trickplay' playlist fields. Any value will be accepted by SRM for either of these, which will be passed through to the Ericsson VSPP server.
- ▶ SRM now interprets a 'valid' response from the Ericsson VSPP server as 200 (was 302).
- ▶ New header 'x-mayNotify' (empty value) added to setup/teardown requests to Ericsson VSPP server.



- ▶ 'client' key/value removed from the 'transport' header.

**Upgrade Notes**

**Configuration Changes**

The following table illustrates any changes to the configuration in Foreman.

Param	Change Description	Old Default	New Default
TODO			

**Configuration Changes for Deployment with SAGEM and VSPP**

The following table illustrates all configuration changes that must be made in production in Foreman when SAGEM and VSPP are used in the SETUP/TEARDOWN workflow.

Param	DefaultValue	NewValue
TODO		

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
TODO		

**Change Requests**

Request No	Status	Summary
107922	Implemented	VS Lookup request in VAST mode doesn't work with node id on 9 digits

**Open Issues**

Issue No	Status	Summary
TODO		

**Feature List**

Feature No	Status	Summary
TODO		

## 14.4 SRM 5.0STD4

### Session Resource Manager Release 5.0STD4

This section details any information which is relevant since the previous 5.0STD2 release.

**Highlights**

- ▶ Fixed the teardown queue to prevent sending two teardown requests to the SCS when a mis-behaving client sends two concurrent DELETE/ TEARDOWN requests
- ▶ Channel and ServiceGroupResource implementations have been changed to ensure data integrity in mongodb with discrepancies on internal resource management. In addition, Status and SrmSessionUUID having been added into ServiceGroupResource model.

**Upgrade Notes**

N/A

**Data Migration**

Instructions on how to run following script which can be found within the scripts.

1. Add SrmSessionUuid and Status fields to ServiceGroupResource collection. This is done using the following script:

servicegroupresource-fields-SRM50STD4.js

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
IST[108545]	Fixed	Increasing number of disconnections between SRM and SCS

**Change Requests**

N/A

**Feature List**

N/A

## 14.5 SRM 5.0STD5

### Session Resource Manager Release 5.0STD5

This section details any information which is relevant since the previous 5.0STD4 release.

#### Highlights

- ▶ Fixed the ingestion process in SRM to allow the update of ServiceGroupResources using the CSV ingestion API.
- ▶ Added non-persistent connection capability for keepalive requests in SRM. The configuration of setting the keepalive to be persistent/non-persistent has been added to the SRM client, i.e. SrmClient.PersistentConnection. It is set to true by default, which indicates persistent keepalive connection.

#### Upgrade Notes

##### Configuration Changes

The following table illustrates a new configuration change in the SrmClient. A new flag has been added to allow for setting a keepalive connection to non-persistent. By default, it is set to true, i.e. persistent keepalive connection. Also, SrmClient is backward compatible, i.e. old ingested SrmClient data will work as SRM will assume PersistentConnection = true.

Persistent Connection	A flag in SrmClient to set the keepalive connections to persistent (true) or non-persistent (false). It is set to true by default.	N/A	N/A
-----------------------	--	-----	-----

#### Issues List

##### Fixed Issues

Issue No	Status	Summary
IST[108694]	Fixed	SRM cannot update ServiceGroup Resources by CSV ingestion
IST[10841]	Fixed	SRM is not working properly with non-persistent connections with STB

#### Change Requests

N/A

**Feature List**

N/A

## 14.6 SRM 5.0STD6

### Session Resource Manager Release 5.0STD5

This section details any information which is relevant since the previous 5.0STD5 release.

**Highlights**

- ▶ Fixed an ingestion issue on SRM, where SRM takes long time to ingest CSV files with large data until fails due timeout.

**Upgrade Notes**

**Configuration Changes**

N/A

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
IST[109015]	Fixed	SRM takes long time to ingest CSV files with large data until fails due timeout

**Change Requests**

N/A

**Feature List**

N/A

## 14.7 SRM 5.0STD7

### Session Resource Manager Release 5.0STD7

This section details any information which is relevant since the previous 5.0STD6 release.

**Highlights**

- ▶ Fixed SRM Ingestion API to allow operator to inactivate a Channel.
- ▶ Added SRM data-discovery API to retrieve a Channel with its associated Service Groups Resources

**Upgrade Notes**

**Configuration Changes**

N/A

**Data Migration**

Instructions on how to run following script which can be found within the scripts.

1. Add SrmSessionUuid and Status fields to ServiceGroupResource collection. This is done using the following script:  
servicegroupresource-fields-SRM50STD4.js

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
<a href="http://ist.hq.k.grp/cgi-bin/WebObjects/ist.woa/wa/inspectRequest?requestNo=109023">http://ist.hq.k.grp/cgi-bin/WebObjects/ist.woa/wa/inspectRequest?requestNo=109023</a>	Fixed	SRM requires an API to retrieve the Channel and ServiceGroup Resource information
<a href="http://ist.hq.k.grp/cgi-bin/WebObjects/ist.woa/wa/inspectRequest?requestNo=109114">http://ist.hq.k.grp/cgi-bin/WebObjects/ist.woa/wa/inspectRequest?requestNo=109114</a>	Fixed	ServiceGroupResource allocBps will get reset during ingest of the SGR

**Change Requests**

N/A

**Feature List**

N/A

## 14.8 SRM 5.0STD8

### Session Resource Manager Release 5.0STD8

This section details any information which is relevant since the previous 5.0STD7 release.

**Highlights**

- ▶ Fixed PR 109342 ServiceGroup Resources are getting updated with DEALLOC\_ERROR status during teardowns due data discrepancy

**Upgrade Notes**

**Configuration Changes**

N/A

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
<a href="http://ist.hq.k.grp/cgi-bin/WebObjects/ist.woa/wa/inspectRequest?requestNo=109342">http://ist.hq.k.grp/cgi-bin/WebObjects/ist.woa/wa/inspectRequest?requestNo=109342</a>	Fixed	ServiceGroup Resources are getting updated with DEALLOC_ERROR status during teardowns due data discrepancy

**Change Requests**

N/A

**Feature List**

N/A

**14.9 SRM 5.0STD9**

**Session Resource Manager Release 5.0STD9**

This section details any information which is relevant since the previous 5.0STD8 release.

**Highlights**

- ▶ Fixed PR 108521 Teardown Backlog Size (teardown queue size) is growing all the time by collecting OLD and DEAD sessions.

**Upgrade Notes**

**Configuration Changes**

N/A

**Issues List**

**Fixed Issues**

Issue No	Status	Summary
<a href="http://ist.hq.k.grp/cgi-bin/WebObjects/ist.woa/wa/inspectRequest?requestNo=108521">http://ist.hq.k.grp/cgi-bin/WebObjects/ist.woa/wa/inspectRequest?requestNo=108521</a>	Fixed	Teardown Backlog Size (teardown queue size) is growing all the

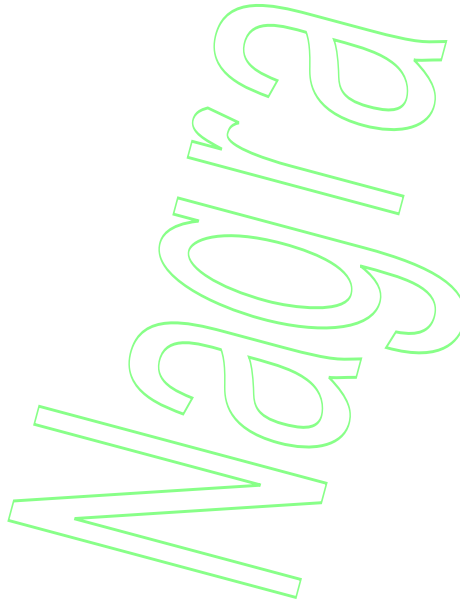
Issue No	Status	Summary
		time by collecting OLD and DEAD sessions.

**Change Requests**

N/A

**Feature List**

N/A



## 15 Release Notes: 5.1

### 15.1 SRM 5.1STD0

#### Session Resource Manager Release 5.0STD9

This section details any information which is relevant since the previous 5.0STD9 release.

##### Highlights

- ▶ Fixed PR 109687 where the default teardown pool size has been modified 32
- ▶ Implemented CR 103912 where we now align logs for a VAST request with regular VOD request
- ▶ Implemented CR 109315 where channel can now be reactivated when the are in use

##### Upgrade Notes

##### Configuration Changes

N/A

##### Issues List

##### Fixed Issues

Issue No	Status	Summary
IST[109687]	Fixed	TearDown Pool Size should have a smaller value by default

##### Change Requests

Request No	Status	Summary
IST [103912]	Fixed	SRM - align logs for a VAST request with regular VOD request
IST[109315]	Fixed	Require ability to reactivate Channels that are in use

##### Open Issues



Issue No	Status	Summary
----------	--------	---------

## 15.2 SRM 5.1STD1

### Session Resource Manager Release 5.1STD1

This section details any information which is relevant since the previous 5.1STD0 release.

#### Highlights

- ▶ Changes in SRM to make it OPF3 compliant - Fixed
- ▶ TSID has been restricted to have a length between 3 and 5 digits - Fixed

#### Upgrade Notes

#### Data Migration

[1] The datatype of three fields in MasterSession have been changed from Long to String. The script "**opf3-migration-script.js**" is provided with this release to make database migration changes to these 3 fields.

[2] Add new object TransportClient into SrmClient documents. The script "**tsid\_migration-script.js**" is provided with this release to make database migration.

#### Note

Instructions on how to run the script are found within the script

#### Issues List

#### Fixed Issues

Issue No	Status	Summary
<a href="http://ist.hq.k.grp/cgi-bin/WebObjects/ist.woa/92/wo/yFgLsuR3JB9s56VhR1ht80/4.6.1.0.15.1.3.3.9.3.3.7.1.3.0.1.1.3.1.1#anchor">http://ist.hq.k.grp/cgi-bin/WebObjects/ist.woa/92/wo/yFgLsuR3JB9s56VhR1ht80/4.6.1.0.15.1.3.3.9.3.3.7.1.3.0.1.1.3.1.1#anchor</a>	Fixed	TSID has been restricted to have a length between 3 and 5 digits

#### Change Requests

Request No	Status	Summary
------------	--------	---------

**Open Issues**

Issue No	Status	Summary

Nagra