



VDS-OS APIs

This chapter describes the HTTPS RESTful APIs for VDS-OS, which consist of the following:

- [API for Origin Service Management, page 2-1](#)
- [API for Live Channel Management, page 2-10](#)
- [API for File Management, page 2-20](#)
- [API for Channel and Resource Status, page 2-24](#)
- [APIs for External Storage, page 2-26](#)
 - [API for Network Storage Shares, page 2-26](#)
 - [API for Mount Option Profiles, page 2-29](#)
- [API for Service Router, page 2-31](#)

Each API provides parity with the Virtual Origin Services Manager (VOSM).

API for Origin Service Management

The Origin Service API supports listing and provisioning of the origin services and has the following API calls:

- [Origin Service List and Definition](#)
- [General Settings](#)
- [Assign Service Engines](#)
- [Cross-Domain Settings](#)
- [On-Demand Encapsulation Settings](#)
- [Asset Resolver Settings](#)
- [Content Protection Settings](#)
- [List of Assigned Live Channels](#)

Origin Service List and Definition

The Origin Service List and Definition part of the Origin Service API provides parity for the following VOSM GUI pages:

- **Services > Virtual Origin Services > Service Definition**
- **Services > Virtual Origin Services > Service Definition > Definition**

Table 2-1 Origin Services List and Definition API Calls

Resource URL	Method	Function Description
/api/OriginServices	GET	Gets the list of all origin services the API user has access to, which is determined by the Domain Management configured for the user.
	POST	Create a new origin service. The VOSM allocates a long type of positive integer ID for the newly created origin service. The “Location” header in the response returns the URI of the created origin service, which includes the ID (for example, Location: /api/OriginServices/237).
/api/OriginServices/{id}	GET	Get the definition settings of an origin service identified by {id}.
	PUT	Modify the definition settings of an origin service identified by {id}.
	DELETE	Delete an origin service identified by {id}.

GET Definition Response Example

Following is an example of a response getting an origin service with the ID 101.

```
<!-- Data Model (response): GET a single Origin Service -->
<OriginService id="101" uri="/api/OriginServices/101">
  <Name>vdsos1</Name>
  <Description>origin service vdsos1</Description>
  <Fqdn>vdsos1.sample.com</Fqdn>
</OriginService>
```

POST or PUT Definition Request Example

Following is an example of a request to create or modify the definition of an origin service.

```
<!-- Data Model (request): POST(create) or PUT(modify) a single Origin Service -->
<OriginService>
  <Name>vdsos1</Name>
  <Description>origin service vdsos1</Description>
  <Fqdn>vdsos1.sample.com</Fqdn>
</OriginService>
```

Get List of Origin Services and the Definition Settings Response Example

Following is an example of a response for getting the list of origin services.

```
<!-- Data Model (response): GET list of Origin Services -->
<OriginServices uri="/api/OriginServices">
  <OriginService id="101" uri="/api/OriginServices/101">
    <Name>vdsos1</Name>
    <Description>origin service vdsos1</Description>
    <Fqdn>vdsos1.sample.com</Fqdn>
  </OriginService>
  <OriginService id="102" uri="/api/OriginServices/102">
```

```

    <Name>vdsos2</Name>
    <Description>origin service vdsos2</Description>
    <Fqdn>vdsos2.sample.com</Fqdn>
  </OriginService>
</OriginServices>

```

General Settings

The General Settings part of the Origin Service API provides parity for the following VOSM GUI page:

- **Services > Virtual Origin Services > Service Definition > General Settings**

Table 2-2 *Origin Service General Settings API Call*

Resource URL	Method	Function Description
/api/OriginServices/{id}/VOSGenSettings	GET	Gets the general settings of the origin service identified by {id}.
	PUT	Configures or modifies the general settings for an origin service identified by {id}.
	DELETE	Deletes the general settings for an origin service identified by {id}. After successfully deleting the general settings, the factory default general settings take effect for this origin service.

GET General Settings Response Example

Following is an example of a response for getting the general settings of an origin service with the ID 101.

```

<!-- Data Model (response): GET the General Settings of an Origin Service -->
<VOSGenSettings id="101" uri="/api/OriginServices/101/VOSGenSettings">
  <Bitrate>30000</Bitrate>
  <DeliveryQos>0</DeliveryQos>
  <ContentExpiry>1464</ContentExpiry>
</VOSGenSettings>

```

PUT General Settings Request Example

Following is an example of a request to create or modify the general settings

```

<!-- Data Model (request): PUT(Create/Modify) the General Settings of an Origin Service -->
<VOSGenSettings>
  <Bitrate>36000</Bitrate>
  <DeliveryQos>10</DeliveryQos>
  <ContentExpiry>1464</ContentExpiry>
</VOSGenSettings>

```

Assign Service Engines

The Assign Service Engines part of the Origin Service API provides parity for the following VOSM GUI page:

- **Services > Virtual Origin Services > Service Definition > Assign Service Engines**

Table 2-3 *Origin Service Assign Service Engines API Call*

Resource URL	Method	Function Description
/api/OriginServices/{id}/SEs	GET	Gets the Service Engines assigned to the origin service identified by {id}.
	POST	Assigns Service Engines to or unassigns Service Engines from the origin service identified by {id}.

GET Assigned Service Engines Response Example

Following is an example of a response to get all Service Engines assigned to an origin service with ID 210.

```
<!-- Data Model (response): GET all Origin Servers assigned to an Origin Service -->
<SEs uri="/api/OriginServices/210/SEs">
  <SE id="215" uri="/api/OriginServices/SEs/215">
    <name>SPCDN-DE-1830000</name>
    <status>Online</status>
    <isServerOffload>>false</isServerOffload>
    <ipAddress>18.0.101.161</ipAddress>
    <location>SPCDN-DE-18-location</location>
    <version>3.0.0.b.85</version>
  </SE>
</SEs>
```

POST Assign Service Engines Request Example

Following is an example of a request to assign Service Engines with IDs 215, 216, 217, and 218 to an origin service.

```
<!-- Data Model (request): POST (assign) Service Engines to an Origin Service -->
<assign>
  <assignIds>215,216,217,218</assignIds>
</assign>

<!-- Data Model (request): POST (unassign) Service Engines from an Origin Service -->
<assign>
  <removeIds>215,216</removeIds>
</assign>
```

Cross-Domain Settings

The Cross-Domain Settings part of the Origin Service API provides parity for the following VOSM GUI page:

- **Services > Virtual Origin Services > Service Definition > Cross-Domain Settings**

Table 2-4 *Origin Service Cross-Domain Settings API Call*

Resource URL	Method	Function Description
/api/OriginServices/{id}/CrossDomainFiles	GET	Get Cross-Domain file information assigned to the origin service identified by {id}.
	DELETE	Removes the Cross-Domain file assignment from the origin service identified by {id}.
/api/OriginServices/{id}/CrossDomainFiles/{fileId}	POST	Assigns a Cross-Domain File to the origin service identified by {id}.

On-Demand Encapsulation Settings

The On-Demand Encapsulation Settings part of the Origin Service API provides parity for the following VOSM GUI page:

- **Services > Virtual Origin Services > Service Definition > On-Demand Encapsulation Settings**

Table 2-5 *Origin Service On-Demand Encapsulation Settings API Call*

Resource URL	Method	Function Description
/api/OriginServices/{id}/ODEConfiguration	GET	Get the current on-demand encapsulation settings enable status. <ul style="list-style-type: none"> • If enabled, list all assigned external Common Intermediate Format (CIF) origin servers or internal CIF origin servers (service engines). • If disabled, do not return a response.
	DELETE	Delete the on-demand encapsulation settings for the origin service identified by {id}.
	PUT	Create or modify on-demand encapsulation by enabling or disabling it, or by assigning or unassigning external or internal CIF origin servers.

Getting an On-Demand Encapsulation Request

```
curl -w@format -v -k -i -X GET -H "Accept: application/xml" -H
"Content-Type: application/xml" -u admin:default
'https://47.112.0.31:8443/api/OriginServices/1860/ODEConfiguration'
```

Sample Data for GET

In the following example:

- 47.112.0.31:8443 is the VOSM IP address and port
- 1860 is the origin service ID
- 271 is the Service Engine ID

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ODEConfiguration uri="/api/OriginServices/1860/ODEConfiguration">
  <ODEConfiguration-row id="1906" uri="/api/ODEConfiguration/1906">
    <enableODE>1</enableODE>
    <isExternalCIF>0</isExternalCIF>
    <externalCIF></externalCIF>
    <locationId>242</locationId>
    <httpResponseReadTimeout></httpResponseReadTimeout>
    <internalCIF>271</internalCIF>
    <channelId>1860</channelId>
  </ODEConfiguration-row>
</ODEConfiguration>
```

Creating or Modifying an On-Demand Encapsulation Request for an Internal CIF

```
<ODEConfiguration>
  <enableODE>1</enableODE>
  <isExternalCIF>0</isExternalCIF>
  <internalCIF>271</internalCIF>
</ODEConfiguration>
```

Creating or Modifying an On-Demand Encapsulation Request for an External CIF

```
<ODEConfiguration>
  <enableODE>1</enableODE>
  <isExternalCIF>1</isExternalCIF>
  <externalCIF>auto.ext-ipndvr.com</externalCIF>
  <internalCIF/>
</ODEConfiguration>
```

Deleting an On-Demand Encapsulation Request

```
curl -w@format -v -k -i -X DELETE -H "Accept: application/xml" -H
"Content-Type: application/xml" -u admin:default
'https://47.112.0.31:8443/api/ODEConfiguration/1906'
```

Asset Resolver Settings

The Asset Resolver Settings part of the Origin Service API provides parity for the following VOSM GUI page:

- **Services > Virtual Origin Services > Service Definition > Asset Resolver Settings**

Table 2-6 *Origin Service Asset Resolver Settings API Call*

Resource URL	Method	Function Description
/api/OriginServices/{id}/AssetResolverFiles	GET	Get Asset Resolver file information assigned to the origin service identified by {id}.
	DELETE	Removes the Asset Resolver file assignment from the origin service identified by {id}.
/api/OriginServices/{id}/AssetResolverFiles/{fileId}	POST	Assigns an Asset Resolver File to the origin service identified by {id}.

Content Protection Settings

The Content Protection Settings part of the Origin Service API provides parity for the following VOSM GUI page:

- **Services > Virtual Origin Services > Service Definition > Content Protection Settings**

Table 2-7 *Origin Service Content Protection Settings API Call*

Resource URL	Method	Function Description
/api/OriginServices/{OsId}/ContentProtectionKeyProfiles/{ProfileId}	PUT	Update the specified content protection key profile.
	DELETE	Delete the specified content protection key profile.
	GET	Get the specified content protection key profile.
/api/OriginServices/{OsId}/ContentProtectionKeyProfiles	POST	Create a content protection key profile.

Creating a New Content Protection Key Profile

Creating or Updating a Cisco Key Store Profile Request

All fields are required unless otherwise indicated.

```
<ContentProtectKeyProfiles-row>
  <Name>cisco_keyStore</Name>
  <Domain>171.71.50.9</Domain>
  <ProfileType>4</ProfileType>                                <!-- ProfileType 4 = Cisco Key Store -->
  <Protocol>HTTP</Protocol>
  <KeyProviderType>PR-AES-128</KeyProviderType><!-- optional -->
  <Port>12</Port>                                           <!-- optional -->
  <QueryStr>NONE</QueryStr>                                <!-- optional -->
  <serviceURI>/test.asmx</serviceURI>                       <!-- optional -->
</ContentProtectKeyProfiles-row>
```

Creating or Updating a Verimatrix Profile Request

All fields are required unless otherwise indicated.

```
<ContentProtectKeyProfiles-row>
  <Name>VerimatrixTest</Name>
  <Domain>171.71.50.9</Domain>
  <ProfileType>2</ProfileType>          <!-- ProfileType 2= Verimatrix -->
  <Protocol>HTTP</Protocol>
  <keyProviderType>PR-AES-128</keyProviderType><!-- optional -->
  <Port>12</Port>                       <!-- optional -->
  <serviceURI>/test.asmx</serviceURI>   <!-- optional -->
  <Certificate>cert.cer</Certificate>   <!-- optional -->
  <Key>key.cer</Key>                     <!-- optional -->
  <QueryStr>NONE</QueryStr>             <!-- optional -->
</ContentProtectKeyProfiles-row>
```

Creating or Updating a Cisco Videoscape Media Suite Key Management Server (VMS-KMS) Profile Request

All fields are required unless otherwise indicated.

```
<ContentProtectKeyProfiles-row>
  <Name>VMS-KMSTest</Name>
  <Domain>171.71.50.9</Domain>
  <ProfileType>3</ProfileType>          <!-- ProfileType 3 = Cisco VMS-KMS -->
  <Protocol>HTTP</Protocol>
  <Port>12</Port>                       <!-- optional -->
  <QueryStr>NONE</QueryStr>             <!-- optional -->
  <serviceURI>/test.asmx</serviceURI>   <!-- optional -->
  <Certificate>cert.cer</Certificate>   <!-- optional -->
  <Key>key.cer</Key>                     <!-- optional -->
  <KeyProviderType>PR-AES-128</KeyProviderType><!-- optional -->
  <authKey>kdwieowi212323</authKey>
  <KeySetTID>11111111-1111-1111-1111-111111111111</KeySetTID>
  <RSAFileDest>rsa.cer</RSAFileDest>
</ContentProtectKeyProfiles-row>
```

Getting a Content Protection Key Profile

Getting a Cisco Key Store Profile Request-Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ContentProtectKeyProfiles-row id="300"
uri="/api/OriginServices/224/ContentProtectKeyProfiles/300">
  <Name>cisco_keyStore</Name>
  <Domain>171.71.50.9</Domain>
  <ProfileType>4</ProfileType>
  <Protocol>HTTP</Protocol>
  <Port>12</Port>
  <QueryStr>NONE</QueryStr>
  <KeyProviderType>PR-AES-128</KeyProviderType>
  <ServiceURI>/test.asmx</ServiceURI>
</ContentProtectKeyProfiles-row>
```

Getting a Verimatrix Profile Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ContentProtectKeyProfiles-row id="375"
uri="/api/OriginServices/350/ContentProtectKeyProfiles/375">
  <Name>RETS_TEST</Name>
  <Domain>171.71.50.9</Domain>
```

```

    <ProfileType>2</ProfileType>
    <Protocol>HTTP</Protocol>
    <Port>12</Port>
    <QueryStr>NONE</QueryStr>
    <ServiceURI>/test.asmx</ServiceURI>
    <keyProviderType>PR-AES-128</keyProviderType>
    <Certificate>cert.cer</Certificate>
    <Key>key.cer</Key>
  </ContentProtectKeyProfiles-row>

```

Getting a Cisco VMS-KMS Profile Request

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ContentProtectKeyProfiles-row id="372"
  uri="/api/OriginServices/350/ContentProtectKeyProfiles/372">
  <Name>VMS-KMS1</Name>
  <Domain>171.71.50.9</Domain>
  <ProfileType>3</ProfileType>
  <Protocol>HTTP</Protocol>
  <Port>12</Port>
  <QueryStr>NONE</QueryStr>
  <ServiceURI>/test.asmx</ServiceURI>
  <authKey>SfFaDwVr3476</authKey>
  <KeyProviderType>PR-AES-128</KeyProviderType>
  <KeySetTId>11111111-1111-1111-1111-111111111111</KeySetTId>
  <RSAKeyFile>-----BEGIN X509 CERTIFICATE-----
  kz/MdXSFmLr/YnpNH4n+rr2UAJm/EaXc4HnFFgt9AmEd6oX5AhVP51qJThRv4zdL&#xD;
  hfXBPGHg/QVbspJ/wx2g0K5SZGBrGMymnNj1ZOQ2GmKfig8+/21OGVZOIJFsnzQz&#xD;
  OjRXUDpvgV4GxvU+FE6OK851Bi5d0ipTdF7Tbleej
  -----END CERTIFICATE-----
  </RSAKeyFile>
  <RSAFileDest>7654321</RSAFileDest>
</ContentProtectKeyProfiles-row>

```

List of Assigned Live Channels

The List of Assigned Live Channels part of the Origin Service API provides parity for the following VOSM GUI page:

- [Services > Virtual Origin Services > Service Definition > Assigned Live Channels](#)

Table 2-8 *Origin Service Assigned Live Channels List API Call*

Resource URL	Method	Function Description
/api/OriginServices/{id}/LiveChannels	GET	Gets all live channels assigned to the origin service identified by {id}.

GET Assigned Live Channels Response Example

Following is an example of a response to get all the live channels assigned to the origin service with ID 101.

```

<!-- Data Model (response): GET all Live Channels associated with an Origin Service -->
<LiveChannels uri="/api/OriginServices/101/LiveChannels">
  <LiveChannels-row id="235" uri="/api/LiveChannels/235">
    <name>live-channell</name>
    <description>live channell</description>
    <tsWindow>120</tsWindow> <!-- default Timeshift Window: minutes in unit -->
    <inMemoryLive>0</inMemoryLive>
  </LiveChannels-row>
</LiveChannels>

```

```

        <scheduleFileId></scheduleFileId>
        <nSS></nSS>
    </LiveChannels-row>
<LiveChannels-row id="236" uri="/api/LiveChannels/236">
    <name>live-channel2</name>
    <description>live channel2</description>
    <tsWindow>100</tsWindow> <!-- default Timeshift Window: minutes in unit -->
    <inMemoryLive>0</inMemoryLive>
    <scheduleFileId></scheduleFileId>
    <nSS></nSS>
</LiveChannels-row>
</LiveChannels>

```

API for Live Channel Management

The Live Channel API supports listing, provisioning, and control of the live channels and has the following API calls:

- [Live Channel List and Definition](#)
- [Assign Origin Service for Live Channel](#)
- [Capture Resources](#)
- [Capture Schedule File](#)
- [Publish Resources](#)
- [Assign Acquisition Nodes](#)

Live Channel List and Definition

The Live Channel List and Definition part of the Live Channel API provides parity for the following VOSM GUI pages:

- **Services > Virtual Origin Services > Live Channels**
- **Services > Virtual Origin Services > Live Channel > Channel Definition**

Table 2-9 Live Channel List and Definition API Calls

Resource URL	Method	Function Description
/api/LiveChannels	GET	Gets the list of all the live channels the API user has access to, which is determined by the Domain Management configured for the user.
	POST	Creates a new live channel.
/api/LiveChannels/{id}	GET	Gets the live channel identified by {id}, a long type of positive integer number allocated by the VOSM when the live channel is created.
	PUT	Modifies the live channel identified by {id}.
	DELETE	Deletes the live channel identified by {id}.
/api/LiveChannels/{id}/control/start	POST	Starts or stops a live channel (no request body needed).
/api/LiveChannels/{id}/control/stop		

GET List of Channels Response Example

Following is an example of a response to get the list of live channels.

```
<!-- Data Model (response): GET list of Live Channels -->
<LiveChannels uri="/api/LiveChannels">
  <LiveChannels-row id="235" uri="/api/LiveChannels/235">
    <name>live-channel1</name>
    <description>live channel1</description>
    <tsWindow>120</tsWindow> <!-- default Timeshift Window: minutes in unit -->
    <inMemoryLive>0</inMemoryLive>
    <scheduleFileId></scheduleFileId>
    <nSS></nSS>
  </LiveChannels-row>
  <LiveChannels-row id="236" uri="/api/LiveChannels/236">
    <name>live-channel2</name>
    <description>live channel2</description>
    <tsWindow>100</tsWindow> <!-- default Timeshift Window: minutes in unit -->
    <inMemoryLive>0</inMemoryLive>
    <scheduleFileId></scheduleFileId>
    <nSS></nSS>
  </LiveChannels-row>
</LiveChannels>
```

GET Live Channel Definition Response Example

Following is an example of a response to get the definition settings of a live channel with ID 235.

```
<!-- Data Model (response): GET a Live Channel -->
<LiveChannels-row id="235" uri="/api/LiveChannels/235">
  <name>live-channel1</name>
  <description>live channel1</description>
  <tsWindow>120</tsWindow> <!-- default Timeshift Window: minutes in unit -->
  <inMemoryLive>0</inMemoryLive>
  <scheduleFileId></scheduleFileId>
  <nSS></nSS>
</LiveChannels-row>
```

POST or PUT Live Channel Definition Request Example

Following is an example of a request to create or modify the definition settings for a live channel.

```
<!-- Data Model (request): POST(create) or PUT(modify) a Live Channel -->
<LiveChannels-row>
  <name>live-channel1</name>
  <description>live channel1</description>
  <tsWindow>120</tsWindow> <!-- default Timeshift Window: minutes in unit -->
  <inMemoryLive>0</inMemoryLive>
  <scheduleFileId></scheduleFileId>
  <nSS></nSS>
</LiveChannels-row>
```

Assign Origin Service for Live Channel

The Assign Origin Service for Live Channel part of the Live Channel API provides parity for the following VOSM GUI page:

- **Services > Virtual Origin Services > Live Channel > Assign Origin Service**

Table 2-10 Live Channel Assign Origin Service API Call

Resource URL	Method	Function Description
/api/LiveChannels/{id}/OriginServices	GET	Gets the origin service assigned to the live channel identified by the {id}.
	DELETE	Unassign origin service from the live channel identified by {id}.
/api/LiveChannels/{id}/OriginServices/{id2}	POST	Assigns the origin service identified by {id2} to the live channel identified by {id}. The request body is empty.

GET Assigned Origin Service Response Example

Following is an example of a response to get the assigned origin service for a live channel.

```
<!-- Data Model (response): GET the associated Origin Service of a Live Channel -->
<OriginServices uri="/api/OriginServices">
  <OriginService id="101" uri="/api/OriginServices/101">
    <Name>vos1</Name>
    <Description>origin service vos1</Description>
    <Fqdn>vos1.sample.com</Fqdn>
  </OriginService>
</OriginServices>
```

Capture Resources

The Capture Resources part of the Live Channel API provides parity for the following VOSM GUI page:

- **Services > Virtual Origin Services > Live Channel > Capture Resources**

Table 2-11 Live Channel Capture Resources API Calls

Resource URL	Method	Function Description
/api/LiveChannels/{id}/CaptureResources	GET	Gets all capture resources for the live channel identified by {id}.
	POST	Creates a capture resource for the live channel identified by {id}.
/api/LiveChannels/{id}/CaptureResources/{id2}	GET	Gets a capture resource.
	PUT	Modifies a capture resource.
	DELETE	Deletes a capture resource.

Table 2-11 Live Channel Capture Resources API Calls (continued)

Resource URL	Method	Function Description
/api/LiveChannels/{id}/CaptureResources/{id2}/control/start	POST	Starts or stops a capture resource.
/api/LiveChannels/{id}/CaptureResources/{id2}/control/stop		
/api/LiveChannels/{id}/CaptureResources/{id2}/CaptureStreams	GET	Gets all stream profiles of a capture resource identified by {id2}
	POST	Creates (adds) a capture stream for a capture resource.
/api/LiveChannels/{id}/CaptureResources/{id2}/CaptureStreams/{id3}	PUT	Modifies a capture stream identified by {id3}.
	DELETE	Deletes a capture stream identified by {id3}.
	GET	Gets a capture stream identified by {id3}.

HLS Capture Resource

The HLS Capture Resource and Capture Stream can be created, modified, or deleted using the APIs specified in [Table 2-11](#).

GET HLS Capture Resource Response Example

Following is an example of a response to get an HLS capture resource for a live channel.

```
<!-- Data Model (response): GET a Capture Resource -->
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CaptureResources-row id="608" uri="/api/LiveChannels/521/CaptureResources/608">
  <baseURI>apple/hls/index.m3u8</baseURI>
  <abrType>1</abrType>
  <arrivalRate>44</arrivalRate>
  <description></description>
  <CaptureStreamList uri="/api/LiveChannels/521/CaptureResources/608">
    <CaptureStream id="610">
      <name>112</name>
      <bitrate>22</bitrate>
    </CaptureStream>
    <CaptureStream id="611">
      <name>113</name>
      <bitrate>22</bitrate>
    </CaptureStream>
    <CaptureStream id="614">
      <name>114</name>
      <bitrate>22</bitrate>
    </CaptureStream>
  </CaptureStreamList>
</CaptureResources-row>
```

GET All HLS Capture Resources Response Example

Following is an example of a response to get all the HLS capture resources for a live channel.

```
<!-- Data Model (response): GET all Capture Resources of a Live Channel -->
<CaptureResources owner-id="521" uri="/api/LiveChannels/521/CaptureResources/">
  <CaptureResources-row owner-id="521" uri="/api/LiveChannels/521/CaptureResources/616">
    <baseURI>apple1/hls/index.m3u8</baseURI>
    <abrType>1</abrType>
    <arrivalRate>2</arrivalRate>
```

```

</description></description>
<CaptureStreamList uri="/api/LiveChannels/521/CaptureResources/">
  <CaptureStream id="617">
    <name>112</name>
    <bitrate>22</bitrate>
  </CaptureStream>
</CaptureStreamList>
</CaptureResources-row>
  <CaptureResources-row owner-id="521"
  uri="/api/LiveChannels/521/CaptureResources/608">
    <baseURI>apple1/hls/index.m3u8</baseURI>
    <abrType>1</abrType>
    <arrivalRate>44</arrivalRate>
    <description></description>
    <CaptureStreamList uri="/api/LiveChannels/521/CaptureResources/">
      <CaptureStream id="610">
        <name>112</name>
        <bitrate>22</bitrate>
      </CaptureStream>
      <CaptureStream id="611">
        <name>113</name>
        <bitrate>22</bitrate>
      </CaptureStream>
      <CaptureStream id="614">
        <name>114</name>
        <bitrate>22</bitrate>
      </CaptureStream>
    </CaptureStreamList>
  </CaptureResources-row>
</CaptureResources>

```

POST New HLS Capture Resource Request Example

Following is an example of a request to create an HLS capture resource for a live channel.

```

<!-- Data Model (request): POST(create) a Capture Resource -->
<CaptureResource>
  <baseURI>apple/hls/index.m3u8</baseURI>
  <abrType>1</abrType> <!-- abrType: 1:HLS, 2:HSS, 3:DASH -->
  <arrivalRate>33</arrivalRate>
  <description></description>
  <CaptureStreamList>
    <CaptureStream>
      <name>high1800</name>
      <bitrate>1800</bitrate> <!-- Kbps -->
    </CaptureStream>
  </CaptureStreamList>
</CaptureResource>

```

PUT Modify HLS Capture Resource Request Example

Following is an example of a request to modify an HLS capture resource for a live channel.

```

<!-- Data Model (request): PUT(modify) a Capture Resource -->
<CaptureResource>
  <baseURI>apple/hls/index.m3u8</baseURI>
  <abrType>1</abrType> <!-- abrType: 1:HLS, 2:HSS, 3:DASH -->
  <arrivalRate>33</arrivalRate>
  <description></description>
  <CaptureStreamList>
    <CaptureStream id="123"> <!-- for update -->
      <name>high1800</name>
      <bitrate>1800</bitrate> <!-- Kbps -->
    </CaptureStream>
    <CaptureStream> <!-- for add -->

```

```

        <name>middle1200</name>
        <bitrate>1200</bitrate> <!-- Kbps -->
    </CaptureStream>
</CaptureStreamList>
</CaptureResource>

```

POST or PUT HLS Stream Profile Request Example

Following is an example of a request to create or modify an HLS capture stream for a capture resource.

```

<!-- Data Model (request): POST(create) or PUT(modify) a stream for a Capture Resource -->
<CaptureStream>
    <name>high3000</name>
    <bitrate>3000</bitrate> <!-- Kbps -->
</CaptureStream>

```

Capture Schedule File

The Capture Schedule File part of the Live Channel API provides parity for the following VOSM GUI page:

- **Services > Virtual Origin Services > Live Channel > Capture Schedule**

The capture schedule files are registered with VOSM first by using the File Management API. For more information, see the [“API for File Management” section on page 2-20](#).

Each live channel can have one Capture Schedule file associated with it.

Table 2-12 Live Channel Capture Schedule File API Calls

Resource URL	Method	Function Description
/api/LiveChannels/{id}/ScheduleFiles	GET	Gets the Capture Schedule XML file associated with the live channel identified by {id}.
	DELETE	Disassociates the Capture Schedule file from the live channel identified by {id}.
/api/LiveChannels/{id}/ScheduleFiles/{id2}	POST	Associates the Capture Schedule XML file (identified by {id2}) with the live channel identified by {id}.

GET Capture Schedule File Response Example

Following is an example of a response to get the Capture Schedule file associated with a live channel with ID 396 (GET /api/LiveChannels/396/ScheduleFiles)

```

<!-- Data Model (response): GET the schedule file of a live channel -->
<LiveChannels uri="/api/LiveChannels/396/ScheduleFiles">
    <File name="schedule_file2.xml" id="415" uri="/api/FileMgmt/files?type=301/415"/>
</LiveChannels>

```

POST Capture Schedule File and DELETE Capture Schedule File

The request and response body to associate a Capture Schedule file with a live channel are empty. The HTTP response code indicates whether the association is successful or not.

POST /api/LiveChannels/{id}/ScheduleFiles/{id2}

DELETE /api/LiveChannels/{id}/ScheduleFiles

Publish Resources

The Publish Resources part of the Live Channel API provides parity for the following VOSM GUI page:

- **Services > Virtual Origin Services > Live Channel > Publish Resources**

Table 2-13 Live Channel Publish Resource API Calls

Resource URL	Method	Function Description
/api/LiveChannels/{id}/PublishResources	GET	Gets all publish resources for the live channel identified by {id}.
	POST	Creates a publish resource for the live channel identified by {id}.
/api/LiveChannels/{id}/PublishResources/{id2}	GET	Gets a publish resource.
	PUT	Modifies a publish resource.
	DELETE	Deletes a publish resource.
/api/LiveChannels/{id}/PublishResources/{id2}/control/start	POST	Starts or stops a publish resource.
/api/LiveChannels/{id}/PublishResources/{id2}/control/stop		

GET Publish Resource Response Example

Following is an example of a response to get a publish resource with ID 1342.

```
<!-- Data Model (response): GET a Publish Resource -->
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<PublishResources-row id="1342" uri="/api/LiveChannels/666/PublishResources/1342">
  <baseURI>apple/hls/index.m3u8</baseURI>
  <abrType>1</abrType>
  <passThruMode>0</passThruMode>
  <startTime></startTime>
  <endTime></endTime>
  <tsWindow>123</tsWindow>
  <inMemoryLive>0</inMemoryLive>
  <drmKeyProfileId></drmKeyProfileId>
  <description></description>
  <CaptureStreamList captureResourceId="1339">
    <CaptureStream id="1341">
      <name>11</name>
      <bitrate>22</bitrate>
    </CaptureStream>
  </CaptureStreamList>
</PublishResources-row>
```

GET List of Publish Resources Response Example

Following is an example of a response to get the list of publish resources for a live channel.

```
<!-- Data Model (response): GET list of Publish Resources -->
<PublishResources owner-id="521" uri="/api/LiveChannels/521/PublishResources/">
  <PublishResources-row owner-id="521" uri="/api/LiveChannels/521/PublishResources/612">
    <baseURI>apple/hls/index.m3u8</baseURI>
    <abrType>1</abrType>
    <passThruMode>0</passThruMode>
    <startTime>1330560000000</startTime>
    <endTime>1330995661000</endTime>
```

```

    <tsWindow>256</tsWindow>
    <inMemoryLive>0</inMemoryLive>
    <drmKeyProfileId></drmKeyProfileId>
    <description></description>
    <CaptureStreamList captureResourceId="608">
      <CaptureStream id="610">
        <name>11</name>
        <bitrate>22</bitrate>
      </CaptureStream>
    </CaptureStreamList>
  </PublishResources-row>
  <PublishResources-row owner-id="521" uri="/api/LiveChannels/521/PublishResources/618">
    <baseURI>publish2</baseURI>
    <abrType>1</abrType>
    <passThruMode>0</passThruMode>
    <startTime>1330560000000</startTime>
    <endTime>1332630662000</endTime>
    <tsWindow>256</tsWindow>
    <inMemoryLive>0</inMemoryLive>
    <drmKeyProfileId></drmKeyProfileId>
    <description></description>
    <CaptureStreamList captureResourceId="608">
      <CaptureStream id="611">
        <name>11</name>
        <bitrate>222</bitrate>
      </CaptureStream>
    </CaptureStreamList>
  </PublishResources-row>
</PublishResources>

```

POST or PUT Publish Resource Request Example, Passthrough Disabled (passThruMode 0)

Following is an example of a request to create or modify a publish resource with passthrough disabled.

```

<!-- Data Model (request): POST(create) or PUT(modify) a Publish Resource -->
<PublishResource >
  <baseURI>apple/hls/index.m3u8</baseURI>
  <abrType>1</abrType>
  <passThruMode>0</passThruMode>
  <startTime>3/1/2013 0:00:00</startTime>
  <endTime>3/2/2013 0:00:00</endTime>
  <tsWindow>22</tsWindow>
  <inMemoryLive>0</inMemoryLive>
  <drmKeyProfileId></drmKeyProfileId>
  <description></description>
  <CaptureStreamList captureResource-Id="619">
    <CaptureStream id="621">
      <name>11</name>
      <bitrate>512</bitrate>
    </CaptureStream>
    <CaptureStream id="622">
      <name>22</name>
      <bitrate>128</bitrate>
    </CaptureStream>
  </CaptureStreamList>
</PublishResource>

```

POST or PUT Publish Resource Request Example, Passthrough Enabled (passThruMode 1)

Following is an example of a request to create or modify a publish resource with passthrough enabled.

```

<!-- Data Model (request): POST(create) or PUT(modify) a Publish Resource -->
<PublishResource >
  <abrType>1</abrType>
  <passThruMode>1</passThruMode>

```

```

    <inMemoryLive>0</inMemoryLive>
    <CaptureStreamList captureResourceId="619"></CaptureStreamList>
</PublishResource>

```

POST or PUT Publish Resource Request Example, In-Memory Live Enabled (inMemoryLive 1)

Following is an example of a request to create or modify a publish resource with in-memory live enabled.



Note

The tsWindow setting must be deleted or empty when the inMemoryLive setting is set to 1.

```

<!-- Data Model (request): POST(create) or PUT(modify) a Publish Resource -->
<PublishResource >
  <baseURI>apple/hls/index.m3u8</baseURI>
  <abrType>1</abrType>
  <passThruMode>0</passThruMode>
  <startTime>3/1/2013 0:00:00</startTime>
  <endTime>3/2/2013 0:00:00</endTime>
  <inMemoryLive>0</inMemoryLive>
  <drmKeyProfileId></drmKeyProfileId>
  <description></description>
  <CaptureStreamList captureResource-Id="619">
    <CaptureStream id="621">
      <name>11</name>
      <bitrate>512</bitrate>
    </CaptureStream>
    <CaptureStream id="622">
      <name>22</name>
      <bitrate>128</bitrate>
    </CaptureStream>
  </CaptureStreamList>
</PublishResource>

```

Add, Modify, or Delete Capture Streams by Updating a Publish Resource

Following is an example of adding, modifying, and deleting capture streams by updating a publish resource.

```

<!-- Data Model (request): PUT(modify) a Publish Resource -->
<PublishResource >
  <baseURI>apple/hls/index.m3u8</baseURI>
  <abrType>1</abrType>
  <passThruMode>0</passThruMode>
  <startTime>3/1/2012 0:00:00</startTime>
  <endTime>3/2/2012 0:00:00</endTime>
  <inMemoryLive>1</inMemoryLive>
  <drmKeyProfileId></drmKeyProfileId>
  <description></description>
  <CaptureStreamList captureResourceId="619">
    <CaptureStream> <!-- add a new stream when id is empty -->
      <name>11</name>
      <bitrate>512</bitrate>
    </CaptureStream>
    <CaptureStream id="622"> <!-- update an existing stream -->
      <name>22</name>
      <bitrate>128</bitrate>
    </CaptureStream>
    <!-- remove all other streams that are missing from this stream list -->
  </CaptureStreamList>
</PublishResource>

```

Assign Acquisition Nodes

The Assign Acquisition Nodes part of the Live Channel API provides parity for the following VOSM GUI page:

- **Services > Virtual Origin Services > Live Channel > Assign Acquisition Nodes**

Table 2-14 Live Channel Assign Acquisition Nodes API Call

Resource URL	Method	Function Description
/api/LiveChannels/{id}/SEs	GET	Gets the Acquisition nodes assigned to the live channel identified by {id}.
	POST	Assigns Acquisition nodes to or unassigns Acquisition nodes from the live channel identified by {id}.

GET All Acquisition Nodes Response Example

Following is an example of a response to get all the Acquisition nodes for a live channel with ID 241.

```
<!-- Data Model (response): GET all Acquisition Nodes assigned to a Live Channel -->
<SEs uri="/api/LiveChannels/241/SEs">
  <SE id="215" uri="/api/LiveChannels/SEs/215">
    <name>SPCDN-DE-1830000</name>
    <status>Online</status>
    <isServerOffload>false</isServerOffload>
    <ipAddress>18.0.101.161</ipAddress>
    <location>SPCDN-DE-18-location</location>
    <version>3.0.0.b.85</version>
  </SE>
</SEs>
```

POST Acquisition Nodes (assign) Request Example

Following is an example of a request to assign Acquisition node with IDs 215, 216, 217, and 218 to a live channel.

```
<!-- Data Model (request): POST (assign) Acquisition Nodes (Service Engines) to a Live Channel -->
<assign>
  <assignIds>215,216,217,218</assignIds>
</assign>
```

POST Acquisition Nodes (unassign) Request Example

Following is an example to unassign Acquisition nodes with IDs 215 and 216 from a live channel.

```
<!-- Data Model (request): POST (unassign) Acquisition Nodes (Service Engines) from a Live Channel -->
<assign>
  <removeIds>215,216</removeIds>
</assign>
```

API for File Management

The File Management API provides support to register a configuration file with the VOSM by using the import or upload method. The File Management API provides parity for the following VOSM GUI pages:

- **Services > Virtual Origin Services > Live Channel > Capture Schedule**
- **System > Configuration > Service Routing > Coverage Zone File Registration**

Using Multipart/Form-Data Request to Upload a File

For the upload method, a multipart/form-data request is used. Following is an example of the upload import method for registering a file that uses the curl utility to upload a Coverage Zone file:

```
curl -k -v -X POST -H "Accept: application/xml" -H "Content-Type: multipart/form-data" -F
"file=@covl.xml;type=application/xml" -F "request=@covl.xml;type=application/xml" -u
admin:default "https://10.74.61.199:8443/api/FileMgmt/files;type=1"
```

In this example, the curl utility uploads the file covl.xml.

If the curl utility is used, another way to upload the file is to use the option -F "file=!sourceFile.xml," which can upload the original file, sourceFile.xml, as a multipart/form-data request.

Supported curl Version

In very old versions of curl (for example, 7.11.0 January 2004), the HTTP multipart/form-data request sent by curl misses the boundary Content-Type header. This does not occur in versions 7.15.x or later.

Add ";type=application/xml" to the -F argument for XML files. For example:

```
curl -k -v -X POST -H "Accept: application/xml" -H "Content-Type: multipart/form-data" -F
"file=@tstvl.xml;type=application/xml" -F
"request=@tstvl.xml;type=application/xml" -u admin:default
"https://1.2.3.4:8443/api/FileMgmt/files;type=301"
```



Note

The curl version 7.19 or later detects an XML file and adds the "type=application/xml" itself.

For more information on the **curl** utility, see the manual (man) pages located at the following URL:

<http://curl.haxx.se/docs/manpage.html>

Table 2-15 File Management API Calls

Resource URL	Method	Function Description
/api/FileMgmt/types	GET	Lists all file types supported by VDS-OS. The response returns the file type value for each kind of supported file.

Table 2-15 File Management API Calls (continued)

Resource URL	Method	Function Description
/api/FileMgmt/files?type={type}	GET	Lists all registered files of type {type}.
	POST	Registers a file of type {type} using the import method (imports the file from an external HTTP or FTP file server).
	POST (multipart/form-data)	Registers a file of type {type} using the upload method (uploads the file from the local disk of the API client). The request Content-Type must be multipart/form-data. See the “POST Upload a File Example” section on page 2-23.
/api/FileMgmt/files?type={type}/{id}	GET	Gets the registered file of type {type}, and file ID {id}. Both {type} and {id} are integer numbers, allocated by VOSM.
	PUT	Modifies a file using the import method.
	PUT (multipart/form-data)	Modifies a file using the upload method. The request Content-Type is multipart/form-data. See the “PUT Modify a File Example (Upload Method)” section on page 2-23.
	DELETE	Deletes a file from VOSM.
/api/FileMgmt/validate?type={type}	POST	Validates a file on a file server, using import method. The file is not registered at this point, so there is no file ID. The file is fetched from the file server and then validated, but VOSM does not store it.
/api/FileMgmt/validate?type={type}/{id}	POST	Validate an existing file that is registered to the VOSM and has a file ID.
/api/FileMgmt/refetch?type={type}/{id}	POST	Refetches a registered file from the file server. If the file on the file server has changes, the file stored on the VOSM is updated.

GET All File Types Response Example

Following is an example of a response to get all the file types for VDS-OS (GET /api/FileMgmt/types).

```
<!-- Data Model (response): GET all file types -->
<fileMgmt uri="/api/fileMgmt">
  <result message="Show all supported file types" status="success">
    <types>
      <type value="17">Geo/IP File</type>
      <type value="1">Coverage Zone File</type>
      <type value="19">CDN Selector File</type>
      <type value="20">Service Rule File</type>
      <type value="22">NAS File</type>
      <type value="26">Root CA File</type>
      <type value="301">Capture Schedule File</type>
      <type value="302">Asset Resolver File</type>
      <type value="304">Cross Domain File</type>
    </types>
  </result>
</fileMgmt>
```

GET All Files of Specified Type Response Example

Following is an example of a response to get all Coverage Zone files; the type is 1 (GET /api/FileMgmt/files;type=1).

```
<!-- Data Model (response): GET files by file type -->
<fileMgmt uri="/api/fileMgmt">
  <result message="2 Coverage Zone Files are displayed." status="success">
    <file type="1" id="401">
      <originUrl>czFile1.xml</originUrl>
      <destName>czFile1.xml</destName>
      <username />
      <password />
      <ttl>10</ttl>
    </file>
    <file type="1" id="402">
      <originUrl>czFile2.xml</originUrl>
      <destName>czFile2.xml</destName>
      <username />
      <password />
      <ttl>10</ttl>
    </file>
  </fileMgmt>
```

GET a Single File Response Example

Following is an example of a response to get a Coverage Zone file (type = 1) with ID 401 (GET /api/FileMgmt/files;type=1/401)

```
<!-- Data Model (response): GET a single file -->
<fileMgmt uri="/api/fileMgmt">
  <result message="1 Coverage Zone File is displayed." status="success">
    <file type="1" id="401">
      <originUrl>czFile1.xml</originUrl>
      <destName>czFile1.xml</destName>
      <username />
      <password />
      <ttl>10</ttl>
    </file>
  </fileMgmt>
```

POST Import a File Request Example

Following is an example of a request to register a Coverage Zone file by the import method (POST /api/FileMgmt/files;type=1). The ID allocated to the file is returned in the response body as the ID attribute of the <file> element.

```
<!-- Data Model (request): register a file by import method -->
<file>
  <originUrl>http://1.2.3.4/pub/czFile1.xml</originUrl>
  <destName>czFile1.xml</destName>
  <username>admin</username>
  <password>password</password>
  <ttl>10</ttl>      <!-- update interval, in minutes -->
</file>
```

PUT Modify a File Request Example (Import Method)

Following is an example of a request to modify a Coverage Zone file with ID 401 by using the import method (PUT /api/FileMgmt/files;type=1/401).

```
<!-- Data Model (request): modify a file by import method -->
<file>
  <originUrl>http://1.2.3.4/pub/czFile1.xml</originUrl>
  <destName>czFile1.xml</destName>
```

```

    <username>admin</username>
    <password>password</password>
    <ttml>20</ttml>    <!-- update interval, in minutes -->
</file>

```

POST Upload a File Example

Following is an example of registering a file by using the upload method (POST `/api/FileMgmt/files?type={type}` with multipart/form-data request).

In the following example, the Capture Schedule XML file (type = 301) is uploaded to the VOSM, and the `scheduleFileInfo.xml` specifies the parameters (also known as file information metadata) used to register the Capture Schedule file.

```

# POST multipart/form-data: register a file by upload method
curl -k -X POST -H "Accept: application/xml" -H "Content-Type: multipart/form-data" -F
"file=@schedule.xml" -F "request=@scheduleFileInfo.xml" -u username:password
"https://1.2.3.4:8443/api/FileMgmt/files?type=301"

```

Currently, only the `destName` element (destination name) needs to be provided for the file metadata.

```

# scheduleFileInfo.xml
<FileInfo>
    <destName>schedule_file2.xml</destName>
</FileInfo>

```

PUT Modify a File Example (Upload Method)

Following is an example of modifying a registered file by using the import method (PUT `/api/FileMgmt/files?type={type}/{id}` with multipart/form-data request).

For example, PUT `/api/FileMgmt/files?type=301/501`

In the following example, the Capture Schedule XML file (type = 301 and ID = 501) is uploaded to the VOSM, and the `scheduleFileInfo.xml` specifies the parameters (also known as file information metadata) used to modify the schedule file.

```

# PUT multipart/form-data: modify a file by upload method
curl -k -X PUT -H "Accept: application/xml" -H "Content-Type: multipart/form-data" -F
"file=@schedule.xml" -F "request=@scheduleFileInfo.xml" -u username:password
"https://1.2.3.4:8443/api/FileMgmt/files?type=301/501"

```

Currently, only `destName` element (destination name) needs to be provided for the file metadata.

```

# scheduleFileInfo.xml
<FileInfo>
    <destName>schedule_file2.xml</destName>
</FileInfo>

```

Set a File as the Default

Following is an example of modifying a registered file to default

(PUT `/api/FileMgmt/default?type=type/id`) with an "Accept: application/xml" request).

For example:

```
PUT /api/FileMgmt/default?type=301/196
```

In the following example, the Default Asset Resolver XML file (type = 302 and ID = 196) is uploaded to the VOSM.

```

curl -k -X PUT -H "Accept: application/xml" -H "Content-Type: application/xml" -d
"<fileinfo><isdefault>1</isdefault></fileinfo>" -u admin:default
"https://127.0.0.1:8443/api/FileMgmt/default?type=302/196"

```

DELETE a File Example

Following is an example of a response to delete a Capture Schedule file (type = 301) with ID 501 (DELETE /api/FileMgmt/files;type=301/501).

The request body is empty. The response returns the deletion result.

```
<!-- Data Model (response): delete a file. -->
<fileMgmt uri="/api/fileMgmt">
  <result message="The file is deleted." status="success">
</fileMgmt>
```

POST Validate a File Example (Import Method)

Following is an example of a request to validate a file by using the import method (POST: /api/FileMgmt/files;type={type}). The file has not been registered with the VOSM; therefore it does not have a file ID.

```
<!-- Data Model (request): validate a file by import method -->
<file>
  <originUrl>http://1.2.3.4/pub/czFile1.xml</originUrl>
  <destName>czFile1.xml</destName>
  <username>admin</username>
  <password>password</password>
</file>
```

POST Validate a File Example (Upload Method)

Following is an example of a response to validate a registered Coverage Zone file with ID 408 by using the upload method (POST: /api/FileMgmt/validate;type=1/408). The request body is empty. The response returns the validation result.

```
<!-- Data Model (response): validation result. -->
<fileMgmt uri="/api/fileMgmt">
  <result message="Valid file" status="success">
  <warning message="Warning: SE(s) U1-2G2-4, spcdn-de-9 does not exist at CDN">
</fileMgmt>
```

POST Refetch a File Response Example

Following is an example of a response to refetch a Capture Schedule file with ID 501 (POST /api/FileMgmt/refetch;type=301/501). The request body is empty. The response returns the refetch operation result.

```
<!-- Data Model (response): refetch operation result -->
<fileMgmt uri="/api/fileMgmt">
  <result message="The file will be refetched soon." status="success">
</fileMgmt>
```

API for Channel and Resource Status

The Channel and Resource Status API provides the status of a live channel and the capture and publish resources. The Channel and Resource Status API provides parity of the Status column on the following VOSM GUI pages:

- **Services > Virtual Origin Services > Live Channels**
- **Services > Virtual Origin Services > Live Channels > Capture Resources**
- **Services > Virtual Origin Services > Live Channels > Publish Resources**

Table 2-16 Live Channel and Resource Status API Calls

Resource URL	Method	Function Description
/api/LiveChannels/status	GET	Queries the live channel status and its capture and publish resources from the VOSM.
/api/LiveChannels/status/force	POST	Force full status updates for each Service Engine. (Includes live channel, capture resource, publish resource.)
/api/LiveChannels/{id}/status	GET	Queries the live channel status from the VOSM (the status aggregated from all Acquisition nodes for the same channel).
/api/LiveChannels/{id}/CaptureResources/{id2}/status	GET	Queries the capture resource status
/api/LiveChannels/{id}/CaptureResources/status	GET	Queries the all capture resources status
/api/LiveChannels/{id}/PublishResources/{id2}/status	GET	Queries the publish resource status
/api/LiveChannels/{id}/PublishResources/status	GET	Queries the all publish resources status

GET Live Channel Status Response Example

Following is an example of a response to get the status of a live channel.

```
<LiveChannel base-uri="http://sr.vos.com/ch1" id="325">
  <Control state="Stop">
    <Message>SPCDN-DE-29: Offline</Message>
    <Message>spcdn-de-23: Time out</Message>
  </Control>
  <Status aggregate-status="Not Available">
    <AcquisitionNode status="Not Available" id="210" reason="" />
    <AcquisitionNode status="Not Available" id="1374" reason="" />
  </Status>
</LiveChannel>
```

GET Capture Resource Status Response Example

Following is an example of a response to get the status of a capture resource.

```
<!-- Data Model (response): control/status result -->
<CaptureResources>
  <CaptureResource base-uri="http://sr.vos.com/ch1/cap-1" id="326">
    <Control disabled="true" state="Stop">
      <Message>SPCDN-DE-29: Offline</Message>
      <Message>spcdn-de-23: Time out</Message>
    </Control>
    <Status>
      <AcquisitionNode status="Stopped" id="243" reason="" />
      <AcquisitionNode status="Failed" id="228" reason="Stream1 Timeout" />
    </Status>
  </CaptureResource>
</CaptureResources>
```

GET Publish Resource Status Response Example

Following is an example of a response to get the status of a publish resource.

```
<!-- Data Model (response): control/status result -->
<PublishResources>
  <PublishResource base-uri="http://sr.vos.com/ch1/publish" id="328">
    <Control disabled="true" state="Stop">
      <Message>SPCDN-DE-29: Offline</Message>
      <Message>spcdn-de-23: Time out</Message>
    </Control>
    <Status>
      <AcquisitionNode status="Stopped" id="243"/>
      <AcquisitionNode status="Failed" id="228" reason="Stream1 Timeout" />
    </Status>
  </PublishResource>
</PublishResources>
```

APIs for External Storage

The External Storage APIs support external storage management and have the following APIs:

- [API for Network Storage Shares](#)
- [API for Mount Option Profiles](#)

The Storage Shares API supports external storage management consisting of Network Storage Share provisioning and association with origin services and live channels.

The Mount Profile API supports configuration of the optional mount profiles and association with Service Engines.

API for Network Storage Shares

The Storage Shares API provides parity for the following VOSM GUI page:

- **System > Configuration > Network Storage Shares**

Table 2-17 Storage Shares API Calls

Resource URL	Method	Function Description
/api/StorageShares	GET	Lists all Network Storage Shares (NSS).
	POST	Creates a NSS.
/api/StorageShares/{id}	PUT	Modifies a NSS identified by the {id}.
	DELETE	Deletes a NSS.

GET All Network Storage Shares Response Example

Following is an example of a response to get all the Network Storage Shares. In this example, there is only one Network Storage Share configured.

```
<!-- Data Model (response): GET list of all Storage Shares -->
<StorageShares uri="/api/StorageShares">
  <StorageShares-row id="354" uri="/api/StorageShares/354">
    <name>nss2</name>
    <type>NFS</type>
    <sharedDir>/aa/bb/cc</sharedDir>
```

```

    <mountPath>movie1</mountPath>
    <numOfMounts>3</numOfMounts>
    <shareAddr1>1.1.1.1</shareAddr1>
    <shareAddr2></shareAddr2>
    <ipRange1></ipRange1>
    <ipRange1End></ipRange1End>
    <ipRange2></ipRange2>
    <ipRange2End></ipRange2End>
    <ipRange3></ipRange3>
    <ipRange3End></ipRange3End>
    <ipRange4></ipRange4>
    <ipRange4End></ipRange4End>
    <locationId>246</locationId>
  </StorageShares-row>
</StorageShares>

```

POST or PUT Network Storage Share Request Example

Following is an example of a request to POST (create) or PUT (modify) a Network Storage Share.

```

<!-- Data Model (request): POST (create) / PUT (modify) a Storage Share -->
<StorageShares-row>
  <name>nss2</name>
  <type>NFS</type>
  <sharedDir>/aa/bb/cc</sharedDir>
  <mountPath>movie1</mountPath>
  <numOfMounts>3</numOfMounts>
  <shareAddr1>1.1.1.1</shareAddr1>
  <shareAddr2></shareAddr2>
  <ipRange1></ipRange1>
  <ipRange1End></ipRange1End>
  <ipRange2></ipRange2>
  <ipRange2End></ipRange2End>
  <ipRange3></ipRange3>
  <ipRange3End></ipRange3End>
  <ipRange4></ipRange4>
  <ipRange4End></ipRange4End>
  <locationId>246</locationId>
</StorageShares-row>

```

Assign Network Storage Share to Origin Service

Assigning a Network Storage Share to an origin service is part of the Origin Service API, which provides parity of the following VOSM GUI page:

- **Services > Virtual Origin Services > Service Definition > Assign Network Storage Shares**



Note

At least one Service Engine must be assigned to the Origin Service first, before assigning Network Storage Shares. The SEs assigned to the origin service are associated with a location; only Network Storage Shares associated with the same location can be assigned to the origin service. Only one location is allowed; therefore, all SEs and Network Storage Shares have the same location.

Table 2-18 *Origin Service Assign Network Storage Shares API Call*

Resource URL	Method	Function Description
/api/OriginServices/{id}/StorageShares	GET	Lists all NSSs assigned to the origin service identified by the {id}.
	PUT	Assigns NSSs to the origin service identified by the {id} and unassigns NSSs from the origin service identified by the {id}.

GET Network Storage Shares Response Example

Following is an example of the response to get the Network Storage Shares assigned to an origin service.

```
<!-- Data Model (response): GET Storage Shares assigned to an Origin Service -->
<StorageShares uri="/api/OriginServices/401/StorageShares">
  <StorageShare name="nss1" id="415" uri="/api/StorageShares/415"/>
  <StorageShare name="nss2" id="416" uri="/api/StorageShares/416"/>
</StorageShares>
```

PUT Assign Network Storage Share Response Example

Following is an example of the response to assign a Network Storage Share to an origin service (PUT /api/OriginServices/{id}/StorageShares).

```
<!-- Data Model (response): assign NSSs to an Origin Service -->
<assign>
  <assignIds>415,416</assignIds>
</assign>
```

PUT Remove Network Storage Share Response Example

Following is an example of the response to remove a Network Storage Share from an origin service (PUT /api/OriginServices/{id}/StorageShares).

```
<!-- Data Model (response): assign NSSs to an Origin Service -->
<assign>
  <removeIds>415,416</removeIds>
</assign>
```

Assign Network Storage Share to Live Channel

Assigning a Network Storage Share to a live channel is part of the Live Channel API, which provides parity of the following VOSM GUI page:

- **Services > Virtual Origin Services > Live Channels > Assign Network Storage Shares**

Only one Network Storage Share can be assigned to a live channel.

Table 2-19 *Live Channel Assign Storage Shares API Calls*

Resource URL	Method	Function Description
/api/LiveChannels/{id}/StorageShares	GET	Gets the NSS assigned to the live channel identified by the {id}.
/api/LiveChannels/{id}/StorageShares/{nssId}	POST	Assigns the NSS with {nssId} to the live channel identified by the {id}.
/api/LiveChannels/{id}/StorageShares	DELETE	Unassigns the NSS from the live channel identified by the {id}.

GET Network Storage Share Response Example

Following is an example of a response to get the Network Storage Share associated with a live channel with ID 396 (GET: /api/LiveChannels/396/StorageShares).

```
<!-- Data Model (response): GET the schedule file of a live channel -->
<StorageShares uri="/api/LiveChannels/396/StorageShares">
  <StorageShare name="nss1" id="416" uri="/api/StorageShares/416"/>
</StorageShares>
```

POST Network Storage Share Example

Use POST /api/LiveChannels/{id}/StorageShares/{nssId} to assign a Network Storage Share to a live channel. The request body and response body are both empty. The HTTP response code indicates whether the association is successful or not.

DELETE a Network Storage Share Example

Use DELETE: /api/LiveChannels/{id}/StorageShares to disassociate a Network Storage Share from a live channel. Both request body and response body are empty. The HTTP response code indicates whether the association is successful or not.

API for Mount Option Profiles

The Mount Profile API provides parity of the following VOSM GUI page:

- **System > Configuration > Mount Option Profiles**

Table 2-20 Mount Profile API Calls

Resource URL	Method	Function Description
/api/MountProfiles	GET	Lists all mount profiles.
	POST	Creates a mount profile.
/api/MountProfiles/{id}	PUT	Modifies a mount profile identified by the {id}.
	DELETE	Deletes a mount profile.

GET All Mount Profiles Response Example

Following is an example of a response to get all mount profiles configured (GET /api/MountProfiles). In this example, there is only one configured mount profile (ID=388).

```
<!-- Data Model (response): GET list of all Mount Option Profiles -->
<MountProfiles uri="/api/MountProfiles">
  <MountProfiles-row id="388" uri="/api/MountProfiles/388">
    <name>profile1</name>
    <nfsAccessMode>ro</nfsAccessMode>
    <nfsReadBlockSize>65536</nfsReadBlockSize>
    <nfsWriteBlockSize>65536</nfsWriteBlockSize>
    <nfsTimeout>30</nfsTimeout>
    <nfsRetrans>30</nfsRetrans>
    <nfsRetry>30</nfsRetry>
  </MountProfiles-row>
</MountProfiles >
```

GET a Mount Profile Response Example

Following is an example of a response to get the configuration settings for a mount profile with ID 388 (GET /api/MountProfiles/388).

```
<!-- Data Model (response): GET a single Mount Option Profile -->
<MountProfiles-row id="388" uri="/api/MountProfiles/388">
  <name>profile1</name>
  <nfsAccessMode>ro</nfsAccessMode>
  <nfsReadBlockSize>65536</nfsReadBlockSize>
  <nfsWriteBlockSize>65536</nfsWriteBlockSize>
  <nfsTimeout>30</nfsTimeout>
  <nfsRetrans>30</nfsRetrans>
  <nfsRetry>30</nfsRetry>
</MountProfiles-row>
```

POST or PUT Mount Profile Request Example

Following is an example of a request to create (POST /api/MountProfiles) or modify (PUT /api/MountProfiles/388) the configuration settings for a mount profile with ID 388.

```
<!-- Data Model (request): POST (create) or PUT (modify) a Mount Option Profile -->
<MountProfiles-row id="388" uri="/api/MountProfiles/388">
  <name>profile1</name>
  <nfsAccessMode>ro</nfsAccessMode>
  <nfsReadBlockSize>65536</nfsReadBlockSize>
  <nfsWriteBlockSize>65536</nfsWriteBlockSize>
  <nfsTimeout>30</nfsTimeout>
  <nfsRetrans>30</nfsRetrans>
  <nfsRetry>30</nfsRetry>
</MountProfiles-row>
```

Assign Mount Option Profile to SE

This part of the SE API is used to assign the mount profile to the SE and has parity with the following VOSM GUI page:

- **Devices > Devices (SE) > Device Activation**

Table 2-21 SE API Call for Mount Profile

Resource URL	Method	Function Description
/api/SEs	GET	Lists all Service Engines.
/api/SEs/{id}	PUT	Modifies a Service Engine, including the Mount Option Profile element <nssMountProfile>.
	DELETE	Deletes an SE.

GET all Service Engines Response Example

Following is an example of a response to get a list of all the Service Engines (SEs).

```
<!-- Data Model (response): GET the list of all Service Engines -->
<SEs uri="/api/SEs">
  <SE id="245" uri="/api/SEs/245">
    <Name>SPCDN-DE-20</Name>
    <Description>n/a</Description>
    <locationId>246</locationId>
    <status>Online</status>
    <isActive>true</isActive>
    <isServerOffload>>false</isServerOffload>
  </SE>
</SEs>
```

```

    <setDefaultCZ>true</setDefaultCZ>
    <FileInfos>-1</FileInfos>
    <MgmtIncomingType>1</MgmtIncomingType>
    <MgmtIncomingIP></MgmtIncomingIP>
    <nssMountProfile>389</nssMountProfile>
  </SE>
</SEs>

```

PUT Assign a Mount Profile to an SE Request Example

Following is an example of a request to modify the SE configuration by assigning a mount profile with ID 389 to the SE. The `<nssMountProfile>` element is used to specify the mount profile ID.

```

<!-- Data Model (request): PUT (modify) a Service Engine -->
<SE id="245" uri="/api/SEs/245">
  <Name>SPCDN-DE-20</Name>
  <Description>n/a</Description>
  <locationId>246</locationId>
  <status>Online</status>
  <isActive>true</isActive>
  <isServerOffload>false</isServerOffload>
  <setDefaultCZ>true</setDefaultCZ>
  <FileInfos>-1</FileInfos>
  <MgmtIncomingType>1</MgmtIncomingType>
  <MgmtIncomingIP></MgmtIncomingIP>
  <nssMountProfile>389</nssMountProfile>
</SE>

```

API for Service Router

The Service Router API provides parity on the following VOSM GUI page:

- **Devices > Devices (SR) > Device Activation**

Table 2-22 Service Router API Calls

Resource URL	Method	Function Description
/api/SRs	GET	Lists all Service Routers.
/api/SRs/{id}	PUT	Modifies the configuration settings of a Service Router.
	DELETE	Deletes an SR.

GET All Service Router Response Example

Following is an example of a response to get a list of all the Service Routers in the system. In this example, there is only one Service Router in the system.

```

<!-- Data Model (response): GET the list of all Service Routers -->
<SRs uri="/api/SRs">
  <SR id="685" uri="/api/SRs/685">
    <Name>SPCDN-DE-8</Name>
    <Description>n/a</Description>
    <isActive>true</isActive> <!--active flag -->
    <status>Online</status>
    <coverageZoneFileId>0</coverageZoneFileId>
    <locationId>319</locationId>
    <replaceable>0</replaceable>
    <MgmtIncomingType>1</MgmtIncomingType>
  </SR>
</SRs>

```

```

    <MgmtIncomingIP></MgmtIncomingIP>
    <MgmtIncomingPort>443</MgmtIncomingPort>
    <WorkType>1</WorkType>
    <isServerOffload>>false</isServerOffload>
  </SR>
</SRs>

```

PUT Modify a Service Router Request Example

Following is an example of a request to modify the settings of a Service Router with ID 685.

```

<!-- Data Model (request): PUT (modify) a Service Router -->
<SR id="685" uri="/api/SRs/685">
  <Name>SPCDN-DE-8</Name>
  <Description>n/a</Description>
  <isActive>>false</isActive> <!--active flag -->
  <coverageZoneFileId>0</coverageZoneFileId>
  <locationId>319</locationId>
  <replaceable>0</replaceable>
  <MgmtIncomingType>1</MgmtIncomingType>
  <MgmtIncomingIP></MgmtIncomingIP>
  <isServerOffload>>false</isServerOffload>
</SR>

```