



HAIVISION

Haivision Gateway 3.7
REST API Integrator's Reference

HVS-ID-INT-MGW-3.7

Edition Notice

© 2015-2023 Haivision. All rights reserved.

This edition and the products it describes contain proprietary and confidential information. No part of this content may be copied, photocopied, reproduced, translated or reduced to any electronic or machine-readable format without prior written permission of Haivision. If this content is distributed with software that includes an end-user agreement, this content and the software described in it, are furnished under license and may be used or copied only in accordance with the terms of that license. Except as permitted by any such license, no part of this content may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Haivision Systems, Inc. Please note that the content is protected under copyright law even if it is not distributed with software that includes an end-user license agreement.

About Haivision

Founded in 2004, Haivision is now a market leader in enterprise video and video streaming technologies. We help the world's top organizations communicate, collaborate and educate. Recognized as one of the most influential companies in video by Streaming Media and one of the fastest growing companies by Deloitte's Technology Fast 500, organizations big and small rely on Haivision solutions to deliver video. Headquartered in Montreal, Canada, and Chicago, USA, we support our global customers with regional offices located throughout the United States, Europe, Asia and South America.

Trademarks

The Haivision logo, Haivision, and certain other marks are trademarks of Haivision. CoolSign is a registered trademark licensed to Haivision Systems, Inc. All other brand or product names identified in this document are trademarks or registered trademarks of their respective companies or organizations.

Disclaimer

The information contained herein is subject to change without notice. Haivision assumes no responsibility for any damages arising from the use of this content, including but not limited to, lost revenue, lost data, claims by third parties, or other damages.

If you have comments or suggestions, please contact infodev@haivision.com.

While every effort has been made to provide accurate and timely information regarding this product and its use, Haivision Systems Inc. shall not be liable for errors or omissions contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Contents

Edition Notice	2
About Haivision	2
Trademarks	2
Disclaimer	2
Contents	3
About This Document	5
Conventions	5
Typographic Conventions and Elements	5
Action Alerts.....	5
Obtaining Documentation.....	6
Getting Service Support	6
Introduction	7
Introduction to REST API	7
Informational Links.....	7
Introduction to JSON.....	7
HMG/HSG Concepts.....	8
Resources	8
Accounts	8
REST API Concepts.....	10
Required Authorization.....	10
Default Success HTTP Status Code.....	10
JSON Properties.....	10
HTTP Request Headers	10
Errors	11
REST API Reference	12
Login and Device Info.....	12
Initiate Session.....	13
Get Session Information.....	14
Delete Existing Session	15
Get Device Info.....	16
Get Device Configuration.....	18
Managing and Configuring Routes	22
Get List of Routes.....	22
Get Route Configuration.....	23
Create a Route.....	23
Update a Route	25
Delete a Route	27
Start or Stop Routes.....	28
Start or Stop a Route's Destination.....	29
Export System Preset	31
Import System Preset	33
Example: Stopping an Individual Destination of a Route	34
Statistics.....	38
Get Route Statistics	38
Get Source Statistics.....	41
Get Destination Statistics.....	42
Get SRT Client Statistics.....	43
Object Model Reference	44
Route Object Model.....	44

Source Object Model.....	45
Destinations Object Model.....	52
Source Statistics Object Model.....	59
Destinations Statistics Object Model.....	60
SRT Statistics Connections Object Model	62

Warranties 68

1-Year Limited Hardware Warranty.....	68
EXCLUSIONS AND LIMITATIONS	68
OBTAINING WARRANTY SERVICE.....	69
APPLICABLE LAW	69
EULA - End User License Agreement.....	70
READ BEFORE USING	70
SLA - Service Level Agreement.....	70
1. Introduction.....	70
2. Definitions	70
3. Service Levels for the Video Content Management System	70
4. Exceptions to Availability for the VCMS	71
5. Credits for Downtime for the VCMS.....	72
6. Support Services for the VCMS	72
7. Service Levels for Haivision Streaming Media Service	73
8. Credits for Outages of Haivision Streaming Media Service	73
9. No Secondary End User Support	73

Getting Help 74

About This Document

Conventions

The following conventions are used to help clarify the content.

Typographic Conventions and Elements

<i>Italics</i>	Used for the introduction of new terminology, for words being used in a different context, and for placeholder or variable text.
bold	Used for strong emphasis and items that you click, such as buttons.
Monospaced	Used for code examples, command names, options, responses, error messages, and to indicate text that you enter.
>	In addition to a math symbol, it is used to indicate a submenu. For instance, File > New where you would select the New option from the File menu.
...	Indicates that text is being omitted for brevity.

Action Alerts

The following alerts are used to advise and counsel that special actions should be taken.



Tip

Indicates highlights, suggestions, or helpful hints.



Note

Indicates a note containing special instructions or information that may apply only in special cases.



Important

Indicates an emphasized note. It provides information that you should be particularly aware of in order to complete a task and that should not be disregarded. This alert is typically used to prevent loss of data.

⚠ Caution

Indicates a potentially hazardous situation which, if not avoided, may result in damage to data or equipment. It may also be used to alert against unsafe practices.

⚠ Warning

Indicates a potentially hazardous situation that may result in physical harm to the user.

Obtaining Documentation

This document was generated from the Haivision InfoCenter. To ensure you are reading the most up-to-date version of this content, access the documentation online at <https://doc.haivision.com>. You may generate a PDF at any time of the current content. See the footer of the page for the date it was generated.

Getting Service Support

For more information regarding service programs, training courses, or for assistance with your support requirements, contact Haivision Technical Support using our Support Portal at: <https://support.haivision.com>.

Introduction

The Haivision Media Gateway/SRT Gateway (HMG/HSG) REST API is an application programming interface that allows you to create custom applications for HMG/HSG.

For an introduction to HMG/HSG, as well as information on managing the server from the Web interface, please see the [User's Guide](#).

Topics Discussed

- [Introduction to REST API](#)
- [HMG/HSG Concepts](#)
- [REST API Concepts](#)

Introduction to REST API

The HMG/HSG REST API is a Representational State Transfer (REST) Application Programming Interface (REST API). REST is a style of software architecture for distributed hyper-media systems such as the World Wide Web. REST provides a set of rules (constraints) to which an architecture should conform. This is in contrast to an "unconstrained architecture" in which services are free to define their own idiosyncratic interfaces.

REST provides a uniform interface between components, allowing them to communicate in a standard way. Requests use the standard HTTP methods; GET, PUT and DELETE requests can do only what is expected.

The effect is that your services are accessible through standard tools, and it is safe for other services and utilities to use yours in ways you did not predict.

Informational Links

Following are some useful external references to learn more about REST:

- [Architectural Styles and the Design of Network-based Software Architectures](#) (dissertation by Roy Fielding)
- [Representational State Transfer](#) (Wikipedia entry)
- [How to Create a REST Protocol](#)
- [REST Anti-Patterns](#)

Introduction to JSON

The REST API uses JSON (JavaScript Object Notation) to format API requests and responses. JSON is a lightweight data-interchange format that is easy for both humans to read and write, and machines to parse and generate. It is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, and Python.

JSON is built on two universal data structures, which are supported by nearly all modern programming languages:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

A full description of JSON can be found at <http://www.json.org>.

HMG/HSG Concepts

This section introduces concepts that you should understand to use the REST API effectively.

Resources

The REST API can create and manipulate the routes used on the HMG/HSG system. This section provides a brief description of their functions.

Identifiers

Each unique HMG/HSG API object is assigned a randomly generated identifier (ID) when it is created. The REST API uses these IDs to identify specific resources.

Route

A *route* is an source-to-multiple-destination resource for distributing a video stream.

Sources

A *source* is the incoming video stream of a route.

Destination

A *destination* is the outgoing video stream of a route. Each route may contain multiple destinations.

Accounts

To simplify setup and security, there are three built-in user accounts available: haiadmin, operator, and user.

Note

Default passwords for each account are provided in the *Important Notice* document that is shipped with your hardware, or available on the [Support Portal](#).

To view the available user accounts:

1. In the side menu under Administration, click **Access Control**.
2. Click **Accounts** in the navigation toolbar.

The available accounts are listed grouped by their roles. The account names and roles assigned to the accounts cannot be changed.

Role	Account Name	Description
Administrator	haiadmin	All access rights and administrator privileges.

Role	Account Name	Description
Operator	operator	All rights to create and configure routes. Does not include rights to the administration screens.
Observer	user	Read-only access to the system. Does not include the rights to the administration screens.

See [REST API Concepts](#) for details on using the available accounts to access the REST API.

REST API Concepts

This section introduces REST API concepts underlying the HMG/HSG REST API.

- [Required Authorization](#)
- [Default Success HTTP Status Code](#)
- [JSON Properties](#)
- [HTTP Request Headers](#)
- [Errors](#)

Required Authorization

The REST API requires user authentication. Execute the [Initiate Session](#) command to start an API session, using the login/password of the appropriate user role as described in [HMG/HSG Concepts](#).

Each API command listed in [REST API Reference](#) includes an "Authorizations" heading that lists the user role necessary to execute the command. If you make an API call with a user role that is not authorized to execute the command, the following response is returned:

```
{
  "error": {
    "type": "SessionAuthorization",
    "message": "Operation not authorized for roles: [User role]"
  }
}
```

where User role is role of the current session's user.

Default Success HTTP Status Code

Unless otherwise specified, the standard response for successful HTTP requests is 200 OK.

JSON Properties

JSON properties are either optional, required, immutable, or ignored. The definitions are as follows:

Property	Description
Optional	If the property is not provided on a PUT or POST, the server uses the default value for PUT requests or the current value stored on the server for POST.
Required	If the property is not provided on a PUT or POST, an error is returned stating that the property must be provided.
Ignored	Ignored properties are informational only in responses and are ignored for PUT and POST requests.
Immutable	Immutable properties cannot be changed. If a PUT or POST request attempts to change or assign an immutable property, an error is returned.

HTTP Request Headers

On a POST or PUT request, the content type request header should be one of the content types supported for that specific API endpoint, either:

- `application/json`

- application/octet-stream
- multipart/form-data

If the content type is different, the HTTP error 415 Unsupported Media Type is returned.

Errors

In case of an error on an API call, the API returns a response with the appropriate HTTP status code and an error message encapsulated in the body:

```
{
  "error": {
    "type": "[string]",
    "message": "[string]"
  }
}
```

REST API Reference

This API command reference lists and describes the available resources for the Haivision Media Gateway/SRT Gateway REST API. This is a list of API calls that can be used to create and manage stream routes on your HMG/HSG.

Topics Discussed

- **Login and Device Info**
 - **Initiate Session**
 - **Get Session Information**
 - **Delete Existing Session**
 - **Get Device Info**
 - **Get Device Configuration**
- **Managing and Configuring Routes**
 - **Get List of Routes**
 - **Get Route Configuration**
 - **Create a Route**
 - **Update a Route**
 - **Delete a Route**
 - **Start or Stop Routes**
 - **Start or Stop a Route's Destination**
 - **Export System Preset**
 - **Import System Preset**
 - **Example: Stopping an Individual Destination of a Route**
- **Statistics**
 - **Get Route Statistics**
 - **Get Source Statistics**
 - **Get Destination Statistics**
 - **Get SRT Client Statistics**
- **Object Model Reference**
 - **Route Object Model**
 - **Source Object Model**
 - **Destinations Object Model**
 - **Source Statistics Object Model**
 - **Destinations Statistics Object Model**
 - **SRT Statistics Connections Object Model**

Login and Device Info

- **Initiate Session**
- **Get Session Information**
- **Delete Existing Session**
- **Get Device Info**
- **Get Device Configuration**

Initiate Session

Initiating a session requires a user name and a password. The configured users are the same as configured for the web application. When you issue the Initiate Session request, a sessionID cookie is returned in the response.

Authorizations: Administrator, Operator, Observer

Note

All subsequent API requests must be made with a valid sessionID cookie in the request header.

Requests

```
POST /api/session

{
  "username" : "[user name]",
  "password" : "[password]"
}
```

Parameters

Name	Type	Description
username	<i>string</i>	Valid user.
password	<i>string</i>	Valid password.

Response

```
{
  "response": {
    "type": "Session",
    "message": "Session successfully started for haiadmin",
    "sessionID": "[Session ID]",
    "lastLoginDate": 1536777877871,
    "numLoginFailures": 0
  }
}
```

Get Session Information

Get existing session information.

Authorizations: Administrator, Operator, Observer

Requests

```
GET /api/session
cookie: sessionId: [Session ID]
```

Parameters

N/A

Response

```
{
  "sessionId": "[Session ID]",
  "displayName": "Administrator",
  "email": "haiadmin@localhost",
  "roles": [
    "Administrator"
  ],
  "startAt": 1536937838919,
  "expireAt": 1536938857529,
  "lastLoginDate": null,
  "numLoginFailures": null,
  "isLicensed": true
}
```

Name	Type	Description
sessionId	string	ID of the session.
displayName	string	User name.
email	string	Email address of user.
roles	object	List of strings of user roles.
startAt	number	User session started at in Unix time.
expireAt	number	When current user session will expire in Unix time.
lastLoginDate	number	Last login date in Unix time.
numLoginFailures	number	Login failure count.
isLicensed	Boolean	Indicates if Haivision Gateway license is valid.

Delete Existing Session

Delete existing session.

Authorizations: Administrator, Operator, Observer

Requests

```
DELETE /api/session
cookie: sessionId: [Session ID]
```

Parameters

N/A

Response

```
{
  "type" : "Session",
  "message" : "[success message]"
}
```

Get Device Info

Retrieves information about the appliance. Use the Device ID value from the `_id` property to reference the device in subsequent API calls.

Authorizations: Administrator, Operator, Observer

Requests

```
GET /api/devices
cookie: sessionId: [Session ID]
```

Parameters

N/A

Response

```
[
  {
    "_id": "[Device ID]",
    "type": "gateway",
    "ip": "127.0.0.1",
    "name": "Haivision Gateway",
    "lastConnectedAt": [Date/time shown in Unix time],
    "statusCode": "ok",
    "status": "Online",
    "statusDetails": "Connection has been established in the last 1 minutes.",
    "serialNumber": null,
    "firmware": "5.0.180611.1530",
    "hasAdminError": false,
    "pendingSync": false,
    "lastConnection": "<1m"
  }
]
```

Name	Type	Description
<code>_id</code>	<i>string</i>	Device ID.
<code>type</code>	<i>string</i>	Device type.
<code>ip</code>	<i>string</i>	IP address of the device. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>⚠ Note The IP address is from the front end to the backend, so 127.0.0.1 is expected.</p> </div>
<code>name</code>	<i>string</i>	Name of device.
<code>lastConnectedAt</code>	<i>number</i>	Last time connected to the device in Unix time.
<code>statusCode</code>	<i>string</i>	Device status code: warn, error, ok, or unknown.

Name	Type	Description
status	<i>string</i>	Device status: Online, Device error, Offline, Never, or Communication error.
statusDetails	<i>string</i>	Device status details.
serialNumber	<i>string</i>	Device serial number.
firmware	string	Device firmware version.
hasAdminError	Boolean	Indicates if device has an error.
pendingSync	Boolean	Indicates if there is a device pending a sync task.
lastConnection	<i>string</i>	Last time connected to the device in string format (e.g., "<1m").

Get Device Configuration

Retrieves the configuration of the device, including the list of routes. The Device ID used in the command is retrieved via the [Get Device Info](#) command.

Authorizations: Administrator, Operator, Observer

Requests

```
GET /api/devices/[Device ID]
cookie: sessionId: [Session ID]
```

Parameters

N/A

Response

Note

Use the config > routes > id properties on subsequent API calls to start/stop the corresponding routes, as shown in [Start or Stop Routes](#).

```
{
  "_id": "[Device ID]",
  "type": "gateway",
  "ip": "127.0.0.1",
  "name": "Haivision Media Gateway",
  "lastConnectedAt": [Date/time in Unix time],
  "lastConfigReadAt": [Date/time in Unix time],
  "statusCode": "ok",
  "status": "Online",
  "statusDetails": "Connection has been established in the last 1 minutes.",
  "serialNumber": null,
  "firmware": "5.0.180611.1530",
  "hasAdminError": false,
  "config": {
    "routes": [ { <Route Object> } ],
    "settings": {
      "networkAdapters": [
        {
          "name": "eth0",
          "address": "10.67.12.128"
        },
        {
          "name": "idrac",
          "address": "169.254.0.2"
        }
      ]
    }
  },
  "calypsos": [
    {
      "id": "Lh5ZNDm7LspssseJ3pg82jw",
      "gatewayName": "MG 141",
      "address": "dev.haivision.com",
      "httpPort": 80,
      "httpsPort": 443,
      "passcode": "[Pairing passcode]",
      "lastConnectedAt": 1536759437398,
      "status": "accepted",
      "statusDetails": "pairing accepted",
      "proxyAddress": "10.69.12.141",
      "proxyHttpPort": 80,
      "proxyHttpsPort": 443,
      "lastConnection": "<1m"
    }
  ],
  "pendingSync": false,
  "lastAcceptedUpdate": 1530219677119,
  "lastConnection": "<1m",
  "gateway": {
    "port": 1080
  },
  "suggestedPollingInterval": {
    "ms": 10000
  }
}
```

Name	Type	Description
_id	string	Device ID.
type	string	Device type.

Name	Type	Description
ip	string	IP address of the device. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Note</p> <p>The IP address is from the front end to the backend, so 127.0.0.1 is expected.</p> </div>
name	string	Name of device.
lastConnectedAt	number	Last time connected to the device in Unix time.
lastConfigReadAt	number	Last time a read config occurred in Unix time.
statusCode	string	Device status code.
status	string	Device status: Online, Device error, Offline, Never, or Communication error.
statusDetails	string	Device status details: warn, error, ok, or unknown.
serialNumber	string	Device serial number.
firmware	string	Device firmware version.
hasAdminError	Boolean	Indicates if device has an error.
config	object	Device configuration object, including route objects (see Route Object Model for definition), networkAdapters objects (see following), and calypsos object (see following).
pendingSync	Boolean	Indicates if there is a device pending a sync task.
lastAcceptedUpdate	number	Last time a device update was accepted.
lastConnection	string	Last time connected to the device in string format (e.g., "<1m").
gateway	object	Gateway device settings: port number.
suggestedPollingInterval	object	Suggested polling device interval setting. Value in milliseconds.

NetworkAdapters Object Model

Name	Type	Destination
name	string	Adapter name.
address	string	IP address of network adapter.

Calypsos Object Model

Name	Type	Destination
address	string	IP address of paired HMP.
gatewayName	string	Gateway name in the relationship
httpPort	number	HTTP port of paired HMP.
httpsPort	number	HTTPS port of paired HMP.
id	string	HMP relationship ID.

Name	Type	Destination
lastConnectedAt	<i>number</i>	Last connected time.
lastConnection	<i>string</i>	Last connection string.
passcode	<i>string</i>	Pairing passcode.
proxyAddress	<i>string</i>	Proxy address.
proxyHttpPort	<i>number</i>	Proxy HTTP port.
proxyHttpsPort	<i>number</i>	Proxy HTTPS port.
status	<i>string</i>	Relationship status: accepted, pair pending, pairing error, pairing timeout, pairing denied, poll timeout, or poll error.
statusDetails	<i>string</i>	Relationship status detail.

Managing and Configuring Routes

- [Get List of Routes](#)
- [Get Route Configuration](#)
- [Create a Route](#)
- [Update a Route](#)
- [Delete a Route](#)
- [Start or Stop Routes](#)
- [Start or Stop a Route's Destination](#)
- [Export System Preset](#)
- [Import System Preset](#)
- [Example: Stopping an Individual Destination of a Route](#)

Get List of Routes

Use this command to get details of all routes. Retrieve the Device ID via the [Get Device Info](#) command.

Active for Version: 3.2+

Authorizations: Administrator, Operator, Observer

✓ Tip

Routes are paged and have a `numPages` attribute in the response. If the `numPages` value is greater than 1, use the `page` and/or `pageSize` parameters to access the rest of the data. For example, `GET /api/.../routes?page=2&pageSize=10` fetches routes 11-20 from the collection.

Requests

```
GET /api/gateway/[Device ID]/routes
cookie: sessionId: [Session ID]
```

Parameters

N/A

Response

```
{
  "data": { [
    <Array of Route Objects>
  ] },
  "numPages": 1,
  "numResults": 3,
  "numActiveOutputConnections": 1,
  "pendingRouteCreates": 1
}
```

Name	Type	Description
data	<i>object</i>	Data object, including route objects (see Route Object Model for definition).
numPages	<i>number</i>	Number of pages of routes. See the above tip on using the numPages attribute.
numResults	<i>number</i>	Number of available routes.
numActiveOutputConnections	<i>number</i>	Number of currently active outputs.
pendingRouteCreates	<i>number</i>	Number of routes that are pending creation, and that are not yet listed.

Get Route Configuration

Use this command to get details of an individual route. Retrieve the Route ID via the [Get Device Configuration](#) command.

Active for Version: 3.1+

Authorizations: Administrator, Operator, Observer

Requests

```
GET /api/gateway/[Device ID]/routes/[Route ID]
cookie: sessionId: [Session ID]
```

Parameters

N/A

Response

Note

See [Route Object Model](#) for definition of the response.

```
{
  <Route Object>
}
```

Create a Route

Use this command to create an individual route.

Authorizations: Administrator, Operator

Requests

```
POST /api/devices/[Device ID]/updates
cookie: sessionId: [Session ID]
```

```
{
  "action": "create",
  "deviceID": "[Device ID]",
  "elementType": "route",
  "fields": {
    "name": "[Route name]",
    "startRoute": [true,false],
    "source": {
      <Source object>
    },
    "destinations": [
      <Destination object list>
    ]
  }
}
```

Parameters

Name	Type	Description
action	• create	Create the specified element type.
deviceID	<i>string</i>	Device ID retrieved via the Get Device Info command.
elementType	• route	Create a route element.
name	<i>string</i>	Name of route.
startRoute	<i>Boolean</i>	If route is started after it is created
source	<i>object</i>	Source object. See Source Object Model for definition.
destinations	<i>object list</i>	Optional. List of route destinations. See Destinations Object Model for definition.

Response

```
{
  "status": "[success message]"
}
```


Update a Route

Use this command to update an individual route.

Authorizations: Administrator, Operator

⚠ Note

You must include all route details (source and destinations configurations) with your update. That is, if destinations are excluded, they will be deleted from the route.

Requests

```
POST /api/devices/[Device ID]/updates
cookie: sessionID: [Session ID]
```

```
{
  "action": "update",
  "deviceID": "[Device ID]",
  "elementType": "route",
  "elementID": "[Route ID]",
  "fields": {
    "name": "[Route name]",
    "source": {
      <Source object>
    },
    "destinations": [
      <Destination object list>
    ]
  }
}
```

Parameters

Name	Type	Description
action	• update	Update the specified element type.
deviceID	string	Device ID retrieved via the Get Device Info command.
elementType	• route	Update the route element.
elementID	string	Route ID retrieved via the Get Device Configuration command
name	string	Name of route.
source	object	Source object model. See Source Object Model for definition.
destinations	string	Optional, if no destinations are desired in the route. Destinations object model. See Destinations Object Model for definition.

Response

```
{  
  "status": "[success message]"  
}
```

Delete a Route

Use this command to delete an individual route.

Authorizations: Administrator, Operator

Requests

```
POST /api/devices/[Device ID]/updates
cookie: sessionId: [Session ID]

{
  "action": "delete",
  "deviceID": "[Device ID]",
  "elementType": "route",
  "elementID": "[Route ID]"
}
```

Parameters

Name	Type	Description
action	<ul style="list-style-type: none">delete	Delete the specified element type.
deviceID	<i>string</i>	Device ID retrieved via the Get Device Info command.
elementType	<ul style="list-style-type: none">route	Delete a route element as specified by elementID.
elementID	<i>string</i>	Route ID to delete. Retrieved via the Get Device Configuration command.

Response

```
{
  "status": "[success message]"
}
```

Start or Stop Routes

Use this command to start or stop individual routes.

Authorizations: Administrator, Operator

Requests

```
POST /api/devices/[Device ID]/commands
cookie: sessionId: [Session ID]

{
  "deviceID": "[Device ID]",
  "command": "[command]",
  "parameters": {
    "routeID": "[Route ID]"
  }
}
```

Parameters

Name	Type	Description
deviceID	string	Device ID retrieved via the Get Device Info command.
command	<ul style="list-style-type: none"> start-route stop-route 	Command to issue to the route.
routeID	string	Route ID retrieved via the Get Device Configuration command.

Response

```
[
  {
    "action": "command",
    "command": "start-route",
    "parameters": {
      "routeID": "[Route ID]"
    },
    "deviceID": "[Device ID]",
    "createdAt": [Date/time in Unix time],
    "completedAt": 0,
    "result": null,
    "state": "pending",
    "_id": "a5x4-7KEApdS0UuAUUCSog"
  }
]
```

Note

Status of the state property can be polled via the [Get Device Configuration](#) command. It may take several seconds to complete the command.

Start or Stop a Route's Destination

Use this command to start or stop a destination within a route.

Authorizations: Administrator, Operator

Note

You must include all route destinations, even those not being started or stopped. If destinations are excluded, they will be deleted from the route.



Requests

```
POST /api/devices/[Device ID]/updates
cookie: sessionID: [Session ID]

{
  "action": "update",
  "deviceID": "[Device ID]",
  "elementType": "route",
  "elementID": "[Route ID]",
  "fields":
  {
    "name": "[Route name]",
    "source":
    {
      "name": "[Source name]",
      "id": "[Source ID]",
      "address": "0.0.0.0",
      "port": 1234
    },
    "destinations": [
      {
        "name": "[Destination name]",
        "id": "[Destination ID]",
        "action": "[Command]",
        "address": "10.69.12.148",
        "port": 4900
      },
      (other destinations not being started/stopped)
    ]
  }
}
```

Parameters

Name	Type	Description
action	• update	Update the specified element type.
deviceID	string	Device ID retrieved via the Get Device Info command.
elementType	• route	Update a route element.

Name	Type	Description
elementID	<i>string</i>	Route ID retrieved via the Get Device Configuration command.
name	<i>string</i>	Name of route.
source	<i>object</i>	Source object model. See Source Object Model for definition. <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p> Tip Only the fields shown above are required.</p> </div>
destinations	<i>object</i>	Destination object model. See Destinations Object Model for definition. <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p> Tip Only the fields shown above are required.</p> </div>

Response

```
{
  "status": "[success message]"
}
```

Export System Preset

Retrieves a preset of the entire device, including route configurations.

Authorizations: Administrator

Requests

```
GET /api/gateway/[Device ID]/system/preset
cookie: sessionId: [Session ID]
```

Parameters

N/A

Response

Response type is a downloadable file with `Content-Disposition: attachment` where the contents of the file are of type `application/json`. This file can be POSTed back to the Media Gateway/SRT Gateway using the [Import System Preset](#) command to return the system to the state captured by the preset at the moment of export.

```
{
  "preset_version": "1.0",
  "product_version": "3.0.0.79",
  "routes": [
    {
      "name": "Test route",
      "source": {
        "name": "Source Name",
        "networkInterface": "",
        "mode": "unicast",
        "address": "0.0.0.0",
        "protocol": "udp",
        "port": 1234,
        "encryption": "none",
        "srtLatency": null,
        "srtPassPhrase": null
      },
      "destinations": [
        {
          "name": "Destination Name",
          "protocol": "udp",
          "port": 7890,
          "mode": "unicast",
          "networkAddress": null,
          "networkInterface": "",
          "address": "1.2.3.4",
          "ttl": 64,
          "mtu": 1496,
          "tos": 184,
          "encryption": "none",
          "fec": "none",
          "shaping": false,
          "maxBitrate": 10000
        }
      ]
    }
  ]
}
```


Import System Preset

Loads a preset of the entire device, previously downloaded using the [Export System Preset](#) command.

Authorizations: Administrator

Requests

```
POST /api/gateway/[Device ID]/system/preset
cookie: sessionID: [Session ID]
Content-Type: multipart/form-data
```

A previously downloaded Gateway preset (.hmg file) should be posted using the multipart formdata file upload pattern also used by web browser file upload controls. Implementations of multipart file uploads are available in libraries for most programming/scripting languages.

Note

The format may vary depending on the REST library in use.

Parameters

N/A

Response

```
{
  "type": "postSystemPreset",
  "message": "Successfully loaded preset",
  "pendingCreation": [
    "[Route Name 1]",
    "[Route Name 2]"
  ]
}
```

Example: Stopping an Individual Destination of a Route

Stopping an individual destination of an already existing route requires multiple API calls:

1. Login
2. Get device ID.
3. Get list of routes.
4. Stop individual destination.
5. Confirm the destination is stopped.

Step 1: Login

If you are not already logged in, use the [Initiate Session](#) command to login.

```
POST https://192.168.0.168/api/session
{
  "username" : "[user name]",
  "password" : "[password]"
}
```

Step 2: Get Device ID

Use the [Get Device Info](#) commands to obtain the device ID.

```
GET https://192.168.0.168/api/devices
```

The server returns the device information. Copy the "_id" value for use in the next step.

```
[
  {
    "_id": "9cr86XGNGaxtGyS0tyjPDA",
    "type": "gateway",
    "ip": "127.0.0.1",
    "name": "Haivision Gateway",
    "lastConnectedAt": 1603791478291,
    "statusCode": "ok",
    "status": "Online",
    "statusDetails": "Connection has been established in the last 1 minutes.",
    "serialNumber": null,
    "firmware": "5.5.201009.1610",
    "hasAdminError": false,
    "pendingSync": false,
    "lastConnection": "<1m"
  }
]
```

Step 3: Get List of Routes

Assuming the route already exists, using the id value from the previous step, get the list of all routes to find the route you wish to change. (See [Get List of Routes](#) for details.)

```
GET https://10.69.10.175/api/gateway/9cr86XGNGaxtGyS0tyjPDA/routes
```

The server returns a list of all sources. Find the route you wish to stop the destinations of, and copy its contents for use in the next step.

```
...
{
  "name": "StopDestAPIExample",
  "elapsedTime": "00:22:24",
  "id": "5912f2cd-00cb-41e9-b369-b6f1e139542b",
  "state": "running",
  "source": {
    "name": "SourceName",
    "id": "bdc42cc1-b982-4821-9d2f-4687f5b58cca",
    "networkInterface": "",
    "mode": "unicast",
    "address": "0.0.0.0",
    "protocol": "udp",
    "port": 1234,
    "sourceAddress": "",
    "encryption": "none",
    "usedBandwidth": "0.338",
    "state": "connected",
    "srtRcvBuf": 1024000,
    "srtPassPhrase": null,
    "srtGroupMode": "none",
    "summaryStatusCode": "ok",
    "summaryStatusDetails": "connected"
  },
  "destinations": [ {
    "name": "Destination1Name",
    "id": "dcb09ae4-8a1a-488a-a850-cc37da8b8ce8",
    "protocol": "udp",
    "port": 1111,
    "started": true,
    "mode": "unicast",
    "networkAddress": null,
    "networkInterface": "",
    "address": "10.0.65.10",
    "ttl": 64,
    "mtu": 1496,
    "tos": 136,
    "state": "connected",
    "encryption": "none",
    "fec": "none",
    "usedBandwidth": "0.338",
    "summaryStatusCode": "ok",
    "summaryStatusDetails": "connected",
    "shaping": false,
    "maxBitrate": 10000
  },
  {
    "name": "Destination2Name",
    "id": "8728c3d5-b0bf-4108-9a78-21c12c355d61",
    "protocol": "udp",
    "port": 2222,
    "started": true,
    "mode": "unicast",
    "networkAddress": null,
    "networkInterface": "",
    "address": "10.0.65.10",
    "ttl": 64,
    "mtu": 1496,
    "tos": 136,
    "state": "connected",
    "encryption": "none",
```

```

    "fec": "none",
    "usedBandwidth": "0.338",
    "summaryStatusCode": "ok",
    "summaryStatusDetails": "connected",
    "shaping": false,
    "maxBitrate": 10000
  },
  {
    "name": "Destination3Name",
    "id": "d07db32a-7e82-47b9-bcc3-18e5d5037f7b",
    "protocol": "udp",
    "port": 3333,
    "started": true,
    "mode": "unicast",
    "networkAddress": null,
    "networkInterface": "",
    "address": "10.0.65.10",
    "ttl": 64,
    "mtu": 1496,
    "tos": 136,
    "state": "connected",
    "encryption": "none",
    "fec": "none",
    "usedBandwidth": "0.338",
    "summaryStatusCode": "ok",
    "summaryStatusDetails": "connected",
    "shaping": false,
    "maxBitrate": 10000
  }
],
"summaryStatusCode": "ok",
"summaryStatusDetails": "running"
}
...

```

Step 4: Stop Individual Destination

Using the route update command, send the contents of the route above removing the read-only values and adding an "stop" action on the destination that you wish to stop. See [Start or Stop a Route's Destination](#) for the proper format.

Note

If you only include the destination you wish to stop, all other route destinations will be deleted.

```
POST /api/devices/9cr86XNGGaxtGyS0tyjPDA/updates
```

```

{
  "action": "update",
  "deviceID": "9cr86XNGGaxtGyS0tyjPDA",
  "elementType": "route",
  "elementID": "5912f2cd-00cb-41e9-b369-
b6f1e139542b",
  "fields": {
    "name": "StopDestAPIExample",
    "source": {
      "name": "SourceName",
      "id": "bdc42cc1-
b982-4821-9d2f-4687f5b58cca",
      "networkInterface": "",
      "address": "0.0.0.0",
      "protocol": "udp",

```

```
    "port": 1234,
    "sourceAddress": ""
  },
  "destinations": [ {
    "name": "Destination1Name",
    "id": "dcb09ae4-8a1a-488a-a850-
cc37da8b8ce8",
    "action": "stop",
    "protocol": "udp",
    "port": 1111,
    "networkInterface": "",
    "address": "10.0.65.10",
    "ttl": 64,
    "mtu": 1496,
    "tos": 136,
    "encryption": "none",
    "fec": "none",
    "shaping": false,
    "maxBitrate": 10000
  },
  {
    "name": "Destination2Name",
    "id": "8728c3d5-
b0bf-4108-9a78-21c12c355d61",
    "protocol": "udp",
    "port": 2222,
    "networkInterface": "",
    "address": "10.0.65.10",
    "ttl": 64,
    "mtu": 1496,
    "tos": 136,
    "encryption": "none",
    "fec": "none",
    "shaping": false,
    "maxBitrate": 10000
  },
  {
    "name": "Destination3Name",
    "id": "d07db32a-7e82-47b9-
bcc3-18e5d5037f7b",
    "protocol": "udp",
    "port": 3333,
    "networkInterface": "",
    "address": "10.0.65.10",
    "ttl": 64,
    "mtu": 1496,
    "tos": 136,
    "encryption": "none",
    "fec": "none",
    "shaping": false,
    "maxBitrate": 10000
  }
]
}
```

Step 4: Confirm the Destination is Stopped

Confirm the action by issuing the [Get List of Routes](#) command again.

```
GET https://10.69.10.175/api/gateway/9cr86XGNGaxtGyS0tyjPDA/routes
```

The result shows that only the specified destination in the previous step was stopped.

```
...
{
  "name": "StopDestAPIExample",
  "elapsedTime": "00:54:26",
  "id": "5912f2cd-00cb-41e9-b369-b6f1e139542b",
  "state": "running",
  "source": {
    "name": "SourceName",
    ...
  },
  "destinations": [
    {
      "name": "Destination1Name",
      ...
      "state": "disconnected",
      ...
    },
    {
      "name": "Destination2Name",
      ...
      "state": "connected",
      ...
    },
    {
      "name": "Destination3Name",
      ...
      "state": "connected",
      ...
    }
  ],
  "summaryStatusCode": "ok",
  "summaryStatusDetails": "running"
}
...
}
```

Statistics

- [Get Route Statistics](#)
- [Get Source Statistics](#)
- [Get Destination Statistics](#)
- [Get SRT Client Statistics](#)

Get Route Statistics

Retrieves statistics about a specific route.

Authorizations: Administrator, Operator, Observer

Requests

```
GET /api/gateway/[Device ID]/statistics?routeID=[Route ID]
cookie: sessionID: [Session ID]
```

Parameters

Name	Type	Description
Device ID	<i>string</i>	Device ID retrieved via the Get Device Info command.
Route ID	<i>string</i>	Route ID retrieved via the Get Device Configuration command.

Response

Response varies depending on which protocol is used.

Note

The Source ID and Destination ID can be used in subsequent API calls to obtain their specific statistics. See [Get Source Statistics](#) and [Get Destination Statistics](#) for more information.

```
{
  "collectedAt": [Date/time in Unix time],
  "route": {
    "name": "[Route Name]",
    "elapsedRunningTime": "00:00:14",
    "id": "[Route ID]",
    "state": "running",
    "source": {
      <Source Statistics object>
    },
    "destinations": [
      {
        <Destination Statistics object>
      }
    ]
  }
}
```

Parameters

Name	Type	Description
collectedAt	<i>string</i>	Timestamp when statistics were collected.
route	<i>object</i>	Refer to the table below for the route object model.

The route object has the following fields:

Name	Type	Description
id	<i>string</i>	Unique identifier for the route.
name	<i>string</i>	Name of the route.
elapsedRunningTime	<i>number</i>	Either an empty string (for idle routes), or a string in HH:MM:SS format (e.g. 00:03:46).
state	<i>string</i>	Whether the route is running, idle, or stopped.
source	<i>object</i>	See Source Statistics Object Model for the source statistics object model.
destinations	<i>object list</i>	See Destinations Statistics Object Model for the destination statistics object model.

Get Source Statistics

Retrieves statistics about a specific route's source. **Authorizations:** Administrator, Operator, Observer

Requests

```
GET /api/gateway/[Device ID]/statistics?routeID=[Route ID]&sourceID=[Source ID]
cookie: sessionId: [Session ID]
```

Parameters

Name	Type	Description
Device ID	<i>string</i>	Device ID retrieved via the Get Device Info command.
Route ID	<i>string</i>	Route ID retrieved via the Get Device Configuration command.
Source ID	<i>string</i>	Source ID retrieved via the Get Route Statistics command.

Response

Response varies depending on which protocol is used. See [Source Statistics Object Model](#) for parameter definitions of the source statistics object.

```
{
  "collectedAt": [Date/time in Unix time],
  "source": {
    <Source Statistics Object>
  }
}
```

Get Destination Statistics

Retrieves statistics about a specific route's destination.

Authorizations: Administrator, Operator, Observer

Tip

With HMG/HSG version 3.7.4+, you can use the destination name in the request to return its statistics. If special characters are used in the destination name (for example, +, ?, &), ensure you use a URL-encoded format in the request.

Requests

```
GET /api/gateway/[Device ID]/statistics?routeID=[Route ID]&destinationID=[Destination ID]
cookie: sessionID: [Session ID]
```

or

```
GET /api/gateway/[Device ID]/statistics?routeID=[Route ID]&destinationName=[Destination Name]
cookie: sessionID: [Session ID]
```

Parameters

Name	Type	Description
Device ID	<i>string</i>	Device ID retrieved via the Get Device Info command.
Route ID	<i>string</i>	Route ID retrieved via the Get Device Configuration command.
Destination ID	<i>string</i>	Destination ID retrieved via the Get Route Statistics command.
Destination Name	<i>string</i>	Destination name.

Response

Response varies depending on which protocol is used. See [Destinations Statistics Object Model](#) for parameter definitions of the destination statistics object.

```
{
  "collectedAt": [Date/time in Unix time],
  "destination": {
    <Destination Statistics Object>
  }
}
```

Get SRT Client Statistics

Retrieves statistics for a specific client connected to an SRT listener destination.

Authorizations: Administrator, Operator, Observer

Requests

```
GET /api/gateway/[Device ID]/statistics/client?routeID=[Route ID]&destinationID=[Destination ID]&clientAddress=[Client Address]&clientPort=[Client Port]
cookie: sessionId: [Session ID]
```

Parameters

Name	Type	Description
Device ID	<i>string</i>	Device ID retrieved via the Get Device Info command.
Route ID	<i>string</i>	Route ID retrieved via the Get Device Configuration command.
Destination ID	<i>string</i>	Destination ID retrieved via the Get Route Statistics command.
Client Address	<i>string</i>	Address of client to retrieve statistics.
Client Port	<i>string</i>	Optional. Port of client to retrieve statistics.

Response

See Client Statistics Model in [Destinations Statistics Object Model](#) for parameter definitions of the client statistics object.

```
{
  "collectedAt": [Date/time in Unix time],
  "clientStat": [
    <Client Statistics Object>
  ]
}
```

Object Model Reference

This section defines object models that are used across various Haivision Gateway API commands.

Route Object Model

The route object model differs when used in API requests and responses.

POST API Requests

Use the following route model when issuing the [Create a Route](#) and [Update a Route](#) API requests.

Name	Type	Description
name	<i>string</i>	Name of route.
id	<i>string</i>	Optional when creating source, required when updating. Route ID.
source	<i>object</i>	Route source object. See Source Object Model for definition.
destinations	<i>object list</i>	Route destinations list. See Destinations Object Model for definition.

GET API Responses

The following route model is returned after issuing the [Get Device Configuration](#) and [Get Route Configuration](#) API commands.

Name	Type	Description
name	<i>string</i>	Name of route.
id	<i>string</i>	Route ID.
source	<i>object</i>	Route source object. See Source Object Model for definition.
destinations	<i>object list</i>	Route destinations list. See Destinations Object Model for definition.
elapsedTime	<i>string</i>	Elapsed running time of route.
state	<i>string</i>	Route state: idle or running.
pendingUpdates	<i>object</i>	Route update object that lists the values that are pending an update. Only appears while a route update or a stop/start action is occurring.
summaryStatusCode	<i>string</i>	Route status summary code: warn, error, ok, or unknown.
summaryStatusDetails	<i>string</i>	Route status details.
hasPendingDelete	<i>boolean</i>	If the route has a pending delete task.

Source Object Model


The source object model differs when used in API requests and responses.

POST API Requests

Use the following source model when issuing the [Create a Route](#), [Update a Route](#), and [Start or Stop a Route's Destination](#) API requests. Definition of each source depends on the source protocol.

[UDP and RTP](#) [SRT](#) [RTMP](#) [RTSP](#)

UDP or RTP Source

Name	Type	Description
name	<i>string</i>	Name of source for route. (Unique name with length 1-60.)
id	<i>string</i>	Optional when creating source, required when updating. Source ID.
address	<i>string</i>	IP address of route source. (0.0.0.0 for unicast.)
protocol	<i>string</i>	Source protocol: udp or rtp
port	<i>number</i>	Port number of route source: 1-65535.
networkInterface	<i>string</i>	Optional. Network interface name. (Empty string if auto.)
retainHeader	<i>Boolean</i>	Optional. RTP only . To retain headers for RTP tunneling through SRT.
sourceAddress	<i>string</i>	Optional. IGMP v3 source address.
fec	<i>string</i>	Optional. For RTP inputs, PRO-MPEG FEC enable: none or prompeg. <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 5px;"> PRO-MPEG FEC is available only on Haivision SRT Gateway.</div>

[UDP and RTP](#) [SRT](#) [RTMP](#) [RTSP](#)

SRT Source

Name	Type	Description
name	<i>string</i>	Name of source for route. (Unique name with length 1-60.)
id	<i>string</i>	Optional when creating source, required when updating. Source ID.

Name	Type	Description
address	<i>string</i>	IP address of route source. (0.0.0.0 for listener.) If SRT path redundancy is used, this value is not needed. The <code>srtNetworkBondingParams</code> object contains the individual address assignments.
protocol	<i>string</i>	Source protocol: srt
port	<i>number</i>	Port number of route source: 1-65535. If SRT path redundancy is used, this value is not needed. The <code>srtNetworkBondingParams</code> object contains the individual port assignments.
networkInterface	<i>string</i>	Optional. Network interface name. (Empty string if auto.) If SRT path redundancy is used, this value is not needed. The <code>srtNetworkBondingParams</code> object contains the individual network interfaces.
srtPassPhrase	<i>string</i>	Optional. SRT passphrase.
srtLatency	<i>number</i>	Optional. SRT latency.
srtMode	<i>string</i>	Optional. SRT mode: caller, listener, or rendezvous.
srtRcvBuf	<i>number</i>	Optional. SRT receive buffer size in bytes.
srtStreamID	<i>string</i>	Optional. SRT Caller mode only. Stream ID string to identify the listener. <div style="border: 1px solid #f0e68c; padding: 5px; margin-top: 10px;">Note Only available for Haivision SRT Gateway.</div>
useFec	<i>Boolean</i>	Optional. SRT Caller mode only. If FEC is enabled: true or false
srtFecCols	<i>number</i>	Optional. SRT Caller mode only. If FEC is enabled, number of columns.
srtFecRows	<i>number</i>	Optional. SRT Caller mode only. If FEC is enabled, number of rows.
srtFecLayout	<i>string</i>	Optional. SRT Caller mode only. If FEC is enabled, FEC layout: saircase or even.
srtFecArq	<i>string</i>	Optional. SRT Caller mode only. If FEC is enabled, FEC ARQ: never or onreq.
srtGroupMode	<i>string</i>	Optional. SRT path redundancy mode: none, broadcast, backup, or any.
srtNetworkBondingParams	<i>object list</i>	If <code>srtGroupMode</code> is not set to none, array of SRT network bonding parameters.

SRT Network Bonding Parameters Model

In SRT Listener or Caller mode with `srtGroup` mode not set to none, use the following SRT Network Bonding Parameters model when issuing the [Create a Route](#), [Update a Route](#), and [Start or Stop a Route's Destination](#) API requests for each defined network path.

Name	Type	Description
address	<i>string</i>	IP address of route source.
port	<i>number</i>	Port number of route source: 1-65535.
networkInterface	<i>string</i>	Network interface name. (Empty string if auto.)

[UDP and RTP](#)
[SRT](#)
[RTMP](#)
[RTSP](#)

RTMP Source

Name	Type	Description
name	<i>string</i>	Name of source for route. (Unique name with length 1-60.)
id	<i>string</i>	Optional when creating source, required when updating. Source ID.
address	<i>string</i>	Optional in publisher mode, required in consumer mode. IP address of route source.
protocol	<i>string</i>	Source protocol: rtmp
port	<i>number</i>	Port number of route source: 1-65535.
networkInterface	<i>string</i>	Optional. Network interface name. (Empty string if auto.)
streamName	<i>string</i>	RTMP stream name
rtmpMode	<i>string</i>	RTMP mode: PUBLISHER or CONSUMER.

[UDP and RTP](#)
[SRT](#)
[RTMP](#)
[RTSP](#)

RTSP Source

Name	Type	Description
name	<i>string</i>	Name of source for route. (Unique name with length 1-60.)
id	<i>string</i>	Optional when creating source, required when updating. Source ID.
address	<i>string</i>	IP address of route source.
protocol	<i>string</i>	Source protocol: rtsp
port	<i>number</i>	Port number of route source: 1-65535.
networkInterface	<i>string</i>	Optional. Network interface name. (Empty string if auto.)
fec	<i>string</i>	Optional. FEC enable: none or vf
username	<i>string</i>	Optional. RTSP username.
password	<i>string</i>	Optional. RTSP password.

GET API Responses

The following source model is returned after issuing the [Get Device Configuration](#) and [Get Route Configuration](#) API commands. Definition of each source depends on the source protocol.

[UDP and RTP](#) [SRT](#) [RTMP](#) [RTSP](#)

UDP or RTP Source

Name	Type	Description
name	string	Name of source for route.
id	string	Source ID.
address	string	IP address of route source. (0.0.0.0 for unicast.)
protocol	string	Source protocol: udp or rtp
port	number	Port number of route source: 1-65535.
networkInterface	string	Network interface name. (Empty string if auto.)
retainHeader	Boolean	RTP only. To retain headers for RTP tunneling through SRT.
sourceAddress	string	IGMP v3 source address.
fec	string	RTP only. Optional. For RTP inputs, PRO-MPEG FEC enable: none or prompeg. <div style="border: 1px solid #ffc107; padding: 5px;"> <p>Note PRO-MPEG FEC is available only on Haivision SRT Gateway.</p> </div>
state	string	Source state: disconnected, connecting, connection established, or connected.
summaryStatusCode	string	Source status summary code: warn, error, ok, or unknown.
summaryStatusDetails	string	Source status details.

[UDP and RTP](#) [SRT](#) [RTMP](#) [RTSP](#)

SRT Source

Name	Type	Description
name	string	Name of source for route.
id	string	Source ID.
address	string	IP address of route source. (0.0.0.0 for listener.) If path redundancy is used, pipe-delimited string of addresses.
protocol	string	Source protocol: srt

Name	Type	Description
port	<i>number or string</i>	Port number of route source: 1-65535. If path redundancy is used, pipe-delimited string of ports.
networkInterface	<i>string</i>	Network interface name. (Empty string if auto.) If path redundancy is used, pipe-delimited string of interface names.
srtPassPhrase	<i>string</i>	SRT Passphrase.
srtLatency	<i>number</i>	SRT latency.
srtRcvBuf	<i>number</i>	SRT receive buffer size in bytes.
srtStreamID	<i>string</i>	SRT Caller mode only. Stream ID string to identify the listener. <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;">Note Only available for Haivision SRT Gateway.</div>
useFec	<i>Boolean</i>	SRT Caller mode only. If FEC is enabled: true or false.
srtFecCols	<i>number</i>	SRT Caller mode only. If FEC is enabled, number of columns.
srtFecRows	<i>number</i>	SRT Caller mode only. If FEC is enabled, number of rows.
srtFecLayout	<i>string</i>	SRT Caller mode only. If FEC is enabled, FEC layout: saircase or even.
srtFecArq	<i>string</i>	SRT Caller mode only. If FEC is enabled, FEC ARQ: never or onreq.
mode	<i>string</i>	SRT mode: caller, listener, or rendezvous.
state	<i>string</i>	Source state: disconnected, connecting, connection established, or connected.
summaryStatusCode	<i>string</i>	Source status summary code: warn, error, ok, or unknown.
summaryStatusDetails	<i>string</i>	Source status details.
srtGroupMode	<i>string</i>	SRT path redundancy mode: none, broadcast, backup, or any. <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;">Note Backup mode is an experimental feature and will be fully supported in a future version.</div>
srtNetworkBondingParams	<i>object list</i>	If srtGroupMode is not set to none, array of SRT network bonding parameters.

SRT Network Bonding Parameters Model

In SRT Listener or Caller mode with srtGroup mode not set to none, the following SRT Network Bonding Parameters model is returned as part of the response to a [Get Device Configuration](#) or [Get Route Configuration](#) API command for each defined network path.

Name	Type	Description
address	<i>string</i>	IP address of route source.
port	<i>number</i>	Port number of route source: 1-65535.
networkInterface	<i>string</i>	Network interface name. (Empty string if auto.)

[UDP and RTP](#)
[SRT](#)
[RTMP](#)
[RTSP](#)

RTMP Source

Name	Type	Description
name	<i>string</i>	Name of source for route.
id	<i>string</i>	Source ID.
address	<i>string</i>	IP address of route source. (127.0.0.1/live for publisher.)
protocol	<i>string</i>	Source protocol: rtmp
port	<i>number</i>	Port number of route source: 1-65535.
networkInterface	<i>string</i>	Network interface name. (Empty string if auto.)
streamName	<i>string</i>	RTMP stream name
rtmpMode	<i>string</i>	RTMP mode: PUBLISHER or CONSUMER.
state	<i>string</i>	Source state: disconnected, connecting, connection established, or connected.
summaryStatusCode	<i>string</i>	Source status summary code: warn, error, ok, or unknown.
summaryStatusDetails	<i>string</i>	Source status details.

[UDP and RTP](#)
[SRT](#)
[RTMP](#)
[RTSP](#)

RTSP Source

Name	Type	Description
name	<i>string</i>	Name of source for route.
id	<i>string</i>	Source ID.
address	<i>string</i>	IP address of route source.
protocol	<i>string</i>	Source protocol: rtsp

Name	Type	Description
port	<i>number</i>	Port number of route source: 1-65535.
networkInterface	<i>string</i>	Network interface name. (Empty string if auto.)
fec	<i>string</i>	FEC enable: none or vf.
username	<i>string</i>	RTSP username.
password	<i>string</i>	RTSP password.
state	<i>string</i>	Source state: disconnected, connecting, connection established, or connected.
summaryStatusCode	<i>string</i>	Source status summary code: warn, error, ok, or unknown.
summaryStatusDetails	<i>string</i>	Source status details.

Destinations Object Model

The destinations object model differs when used in API requests and responses.

POST API Requests

Use the following destinations model when issuing the [Create a Route](#), [Update a Route](#), and [Start or Stop a Route's Destination](#) API requests. Definition of each destination depends on the protocol.

[UDP and RTP](#) SRT HLS

UDP or RTP Destination

Name	Type	Description
name	<i>string</i>	Name of destination for route. (Unique name with length 1-60.)
id	<i>string</i>	Optional when creating destination, required when updating. Destination ID.
address	<i>string</i>	IP address of route destination.
protocol	<i>string</i>	Destination protocol: udp or rtp.
port	<i>number</i>	Port number of route destination: 1-65535.
networkInterface	<i>string</i>	Optional. Network interface name. (Empty string if auto.)
retainHeader	<i>Boolean</i>	Optional. RTP only . To retain headers for RTP tunneling through SRT.
action	<i>string</i>	Optional. Destination action: start, stop, or none.
mtu	<i>number</i>	Optional. Destination MTU. Range = 280-1500.
ttl	<i>number</i>	Optional. Destination TTL. Range = 1-255.
tos	<i>number</i>	Optional. Destination ToS. Range = 0-255.
fec	<i>string</i>	Optional. <ul style="list-style-type: none"> For UDP, destination FEC enable: none or vf. For RTP, destination PRO-MPEG FEC enable: none or prompeg. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Note PRO-MPEG FEC available only on Haivision SRT Gateway.</p> </div>
prompegFecLevel	<i>string</i>	Optional. RTP only . PRO-MPEG FEC level: a or b.
prompegFecIsBlockAligned	<i>Boolean</i>	Optional. RTP only . PRO-MPEG FEC if block is aligned: true or false.
prompegFecColumns	<i>number</i>	Optional. RTP only . PRO-MPEG FEC number of columns: 1-20.
prompegFecRows	<i>number</i>	Optional. RTP only . PRO-MPEG FEC number of rows: 4-20.
shaping	<i>Boolean</i>	Optional. Destination shaping: true or false.
maxBitrate	<i>number</i>	Optional. Destination maximum bitrate.

SRT Destination

Name	Type	Description
name	<i>string</i>	Name of destination for route. (Unique name with length 1-60.)
id	<i>string</i>	Optional when creating destination, required when updating. Destination ID.
address	<i>string</i>	IP address of route destination. (0.0.0.0 for listener.) If SRT path redundancy is used, this value is not needed. The <code>srtNetworkBondingParams</code> object contains the individual address assignments.
protocol	<i>string</i>	Destination protocol: srt.
port	<i>number</i>	Port number of route destination: 1-65535. If SRT path redundancy is used, this value is not needed. The <code>srtNetworkBondingParams</code> object contains the individual port assignments.
networkInterface	<i>string</i>	Optional. Network interface name. (Empty string if auto.) If SRT path redundancy is used, this value is not needed. The <code>srtNetworkBondingParams</code> object contains the individual network interface names.
retainHeader	<i>Boolean</i>	Optional. To retain headers for RTP tunneling through SRT.
action	<i>string</i>	Optional. Destination action: start, stop, or none.
mtu	<i>number</i>	Optional. Destination MTU. Range = 280-1500. If SRT path redundancy is used, this value is not needed. The <code>srtNetworkBondingParams</code> object contains the individual MTU values.
ttl	<i>number</i>	Optional. Destination TTL. Range = 1-255. If SRT path redundancy is used, this value is not needed. The <code>srtNetworkBondingParams</code> object contains the individual TTL values.
tos	<i>number</i>	Optional. Destination ToS. Range = 0-255. If SRT path redundancy is used, this value is not needed. The <code>srtNetworkBondingParams</code> object contains the individual ToS values.
srtEncryption	<i>string</i>	Optional. Encryption mode: AES128, AES256, or None.
srtPassPhrase	<i>string</i>	Optional. SRT Passphrase.
srtLatency	<i>number</i>	Optional. SRT latency.
srtMode	<i>string</i>	Optional. SRT mode: caller, listener, or rendezvous.
srtOverhead	<i>string</i>	Optional. Overhead used for SRT.
srtStreamID	<i>string</i>	Optional. SRT Caller mode only. Stream ID string to identify the listener. <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;">Note Only available for Haivision SRT Gateway.</div>
useFec	<i>Boolean</i>	SRT Caller mode with path redundancy disabled only. Enable FEC: true or false

Name	Type	Description
srtFecCols	<i>number</i>	If useFEC is enabled, number of columns.
srtFecRows	<i>number</i>	If useFEC is enabled, number of rows.
srtFecLayout	<i>string</i>	If useFEC is enabled, FEC layout: saircase or even.
srtFecArq	<i>string</i>	If useFEC is enabled, FEC ARQ: never or onreq.
srtConnectionLimit	<i>string</i>	SRT Listener mode only. SRT Caller connection limit.
srtGroupMode	<i>string</i>	Optional. SRT path redundancy mode: none, broadcast, backup, or any.
srtNetworkBondingParams	<i>object list</i>	If srtGroupMode is not set to none, array of SRT network bonding parameters.

SRT Network Bonding Parameters Model

In SRT Listener or Caller mode with srtGroupmode not set to none, use the following SRT Network Bonding Parameters destinations model when issuing the [Create a Route](#), [Update a Route](#), and [Start or Stop a Route's Destination](#) API requests for each defined network path.

Name	Type	Description
address	<i>string</i>	IP address of route destination.
port	<i>number</i>	Port number of route destination: 1-65535.
networkInterface	<i>string</i>	Network interface name. (Empty string if auto.)
mtu	<i>number</i>	Optional. Destination MTU. Range = 280-1500
ttl	<i>number</i>	Optional. Destination TTL. Range = 1-255.
tos	<i>number</i>	Optional. Destination ToS. Range = 0-255.

UDP and RTP SRT [HLS](#)

HLS Destination

Name	Type	Description
name	<i>string</i>	Name of destination for route. (Unique name with length 1-60.)
id	<i>string</i>	Optional when creating destination, required when updating. Destination ID.
protocol	<i>string</i>	Destination protocol: hls.
action	<i>string</i>	Optional. Destination action: start, stop, or none.
segmentDuration	<i>number</i>	Optional. HLS segment duration.
useEncryption	<i>Boolean</i>	Optional. Encryption enable: true or false.
segmentsPerKey	<i>number</i>	Optional. Number of HLS segments per key.
multitrackSegmenter	<i>Boolean</i>	Optional. HLS variant playlist enable: true or false.

GET API Responses

The following destinations model is returned after issuing the [Get Device Configuration](#) and [Get Route Configuration](#) API commands. Definition of each destination depends on the protocol.

[UDP and RTP](#) SRT HLS

UDP or RTP Destination

Name	Type	Description
name	<i>string</i>	Name of destination for route.
id	<i>string</i>	Destination ID.
address	<i>string</i>	IP address of route destination. (0.0.0.0 for unicast.)
protocol	<i>string</i>	Destination protocol: udp or rtp.
port	<i>number</i>	Port number of route destination: 1-65535.
networkInterface	<i>string</i>	Network interface name. (Empty string if auto.)
retainHeader	<i>Boolean</i>	RTP only. To retain headers for RTP tunneling through SRT.
mtu	<i>number</i>	Destination MTU. Range = 280-1500
ttl	<i>number</i>	Destination TTL. Range = 1-255.
tos	<i>number</i>	Destination ToS. Range = 0-255.
fec	<i>string</i>	Optional. <ul style="list-style-type: none"> For UDP, destination FEC enable: none or vf. For RTP, destination PRO-MPEG FEC enable: none or prompeg. <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;"> <p>Note PRO-MPEG FEC available only on Haivision SRT Gateway.</p> </div>
prompegFecLevel	<i>string</i>	Optional. RTP only. PRO-MPEG FEC level: a or b.
prompegFecIsBlockAligned	<i>Boolean</i>	Optional. RTP only. PRO-MPEG FEC if block is aligned: true or false.
prompegFecColumns	<i>number</i>	Optional. RTP only. PRO-MPEG FEC number of columns: 1-20.
prompegFecRows	<i>number</i>	Optional. RTP only. PRO-MPEG FEC number of rows: 4-20.
shaping	<i>Boolean</i>	Destination shaping: true or false.
maxBitrate	<i>number</i>	Destination maximum bitrate.
state	<i>string</i>	Destination state: disconnected, connecting, connection established, or connected.
summaryStatusCode	<i>string</i>	Destination status summary code: warn, error, ok, or unknown.
summaryStatusDetails	<i>string</i>	Destination status details.

Name	Type	Description
started	<i>Boolean</i>	If destination is started: true or false.

UDP and RTP [SRT](#) [HLS](#)

SRT Destination

Name	Type	Description
name	<i>string</i>	Name of destination for route.
id	<i>string</i>	Destination ID.
address	<i>string</i>	IP address of route destination. (0.0.0.0 for listener.) If path redundancy is used, pipe-delimited string of addresses.
protocol	<i>string</i>	Destination protocol: srt.
port	<i>number or string</i>	Port number of route destination: 1-65535. If path redundancy is used, pipe-delimited string of ports.
networkInterface	<i>string</i>	Network interface name. (Empty string if auto.) If path redundancy is used, pipe-delimited string of network interfaces.
retainHeader	<i>Boolean</i>	Optional. To retain headers for RTP tunneling through SRT.
mtu	<i>number or string</i>	Destination MTU. Range = 280-1500. If path redundancy is used, pipe-delimited string of MTU values.
ttl	<i>number or string</i>	Destination TTL. Range = 1-255. If path redundancy is used, pipe-delimited string of TTL values.
tos	<i>number or string</i>	Destination ToS. Range = 0-255. If path redundancy is used, pipe-delimited string of ToS values.
srtEncryption	<i>string</i>	Encryption mode: AES128, AES256, or None.
srtPassPhrase	<i>string</i>	SRT Passphrase.
useFEC	<i>Boolean</i>	SRT Caller mode only. If FEC is enabled: true or false
srtFecCols	<i>number</i>	SRT Caller mode only. If FEC is enabled, number of columns.
srtFecRows	<i>number</i>	SRT Caller mode only. If FEC is enabled, number of rows.
srtFecLayout	<i>string</i>	SRT Caller mode only. If FEC is enabled, FEC layout: staircase or even.
srtFecArq	<i>string</i>	SRT Caller mode only. If FEC is enabled, FEC ARQ: never or onreq.
srtConnectionLimit	<i>string</i>	SRT Listener mode only. SRT Caller connection limit.
srtLatency	<i>number</i>	SRT latency.
mode	<i>string</i>	SRT mode: caller, listener, or rendezvous.
srtOverhead	<i>string</i>	Overhead used for SRT.
state	<i>string</i>	Destination state: disconnected, connecting, connection established, or connected.
summaryStatusCode	<i>string</i>	Destination status summary code: warn, error, ok, or unknown.

Name	Type	Description
summaryStatusDetails	<i>string</i>	Destination status details.
started	<i>Boolean</i>	If destination is started: true or false.
srtStreamID	<i>string</i>	SRT Caller mode only. Stream ID string to identify the listener. <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;">Note Only available for Haivision SRT Gateway.</div>
srtGroupMode	<i>string</i>	SRT path redundancy mode: none, broadcast, backup, or any.
srtNetworkBondingParameters	<i>object list</i>	If srtGroupMode is not set to none, array of SRT network bonding parameters.

SRT Network Bonding Parameters Model

In SRT Listener or Caller mode with srtGroupMode not set to none, the following SRT Network Bonding Parameters model is returned as part of the response to a [Get Device Configuration](#) or [Get Route Configuration](#) API command for each defined network path.

Name	Type	Description
address	<i>string</i>	IP address of route destination.
port	<i>number</i>	Port number of route destination: 1-65535.
networkInterface	<i>string</i>	Network interface name. (Empty string if auto.)
mtu	<i>number</i>	Optional. Destination MTU. Range = 280-1500
ttl	<i>number</i>	Optional. Destination TTL. Range = 1-255.
tos	<i>number</i>	Optional. Destination ToS. Range = 0-255.

[UDP and RTP](#)
[SRT](#)
[HLS](#)

HLS Destination

Name	Type	Description
name	<i>string</i>	Name of destination for route.
id	<i>string</i>	Destination ID.
protocol	<i>string</i>	Destination protocol: http.
address	<i>string</i>	HLS playlist path. (Construct the URI to m3u8 playlist for this client by using: https://[IP address] + [HLS playlist path].) Playlist requests may require authorization.
segmentDuration	<i>string</i>	HLS segment duration.
useEncryption	<i>boolean</i>	Encryption enable: true or false.
segmentsPerKey	<i>number</i>	Number of HLS segments per key.

Name	Type	Description
multitrackSegmenter	<i>boolean</i>	HLS variant playlist enable: true or false.
state	<i>string</i>	Destination state: disconnected, connecting, connection established, or connected.
summaryStatusCode	<i>string</i>	Destination status summary code: warn, error, ok, or unknown.
summaryStatusDetails	<i>string</i>	Destination status details.
started	<i>Boolean</i>	If destination is started: true or false.

Source Statistics Object Model

The following source statistics model is returned after issuing the [Get Device Configuration](#) and [Get Source Statistics](#) API commands. The definition depends on the source protocol.

[UDP, RTP, RTMP, and RTSP](#) [SRT](#)

UDP, RTP, RTMP, and RTSP Source

Name	Type	Description
name	string	Name of the source.
id	string	Unique identifier for the source.
mode	string	Unicast or multicast.
elapsedRunningTime	string	Either an empty string (for idle routes), or a string in HH:MM:SS format (e.g. 00:03:46).
signalLosses	number	Number of signal losses.
sendRate	number	Packet send rate in Mbits/s.
numPackets	number	Number of packets sent.
usedBandwidth	number	Bandwidth used in Mbits/s.
bitrate	number	Stream bitrate in Mbits/s.
state	string	Source state: disconnected, connecting, connection established, or connected.
fecLostPackets	number	Number of lost FEC packets.
fecRecoveredPackets	number	Number of recovered FEC packets.
fecUnrecoveredPackets	number	Number of unrecovered FEC packets.
fecReorderedPackets	number	Number of reordered FEC packets.

[UDP, RTP, RTMP, and RTSP](#) [SRT](#)

SRT Source

Name	Type	Description
name	string	Name of the source.
id	string	Unique identifier for the source.
mode	string	SRT mode: caller, listener, or rendezvous.
elapsedRunningTime	string	Either an empty string (for idle routes), or a string in HH:MM:SS format (e.g. 00:03:46).
signalLosses	number	Number of signal losses.

Name	Type	Description
sendRate	<i>number</i>	Packet send rate in Mbits/s.
numPackets	<i>number</i>	Number of packets sent.
usedBandwidth	<i>number</i>	Bandwidth used in Mbits/s.
bitrate	<i>number</i>	Stream bitrate in Mbits/s.
state	<i>string</i>	Source state: disconnected, connecting, connection established, or connected.
srtNumLostPackages	<i>number</i>	SRT number of lost packages.
srtRetransmitRate	<i>number</i>	SRT retransmit rate in bits/s.
srtRoundTripTime	<i>number</i>	SRT round trip time in ms.
srtNegotiatedLatency	<i>number</i>	SRT latency in ms.
srtLatency	<i>number</i>	SRT latency in ms.
srtEncryption	<i>string</i>	SRT encryption.
srtPacketLossRate	<i>number</i>	SRT packet loss rate.
srtNumSkippedPackets	<i>number</i>	SRT number of skipped packet.
srtDecryptionState	<i>string</i>	SRT decryption state: <empty>, active, initializing, inactive (no passphrase), inactive (invalid passphrase).
srtBufferLevel	<i>number</i>	SRT buffer time in ms.
connections	<i>object</i>	An array of SRT source connections objects. See SRT Statistics Connections Object Model .

Destinations Statistics Object Model

The following destination statistics model is returned after issuing the [Get Route Statistics](#) and [Get Destination Statistics](#) API commands. The definition depends on the destination protocol.

[UDP, RTP, and HLS](#) SRT

UDP, RTP, and HLS Destinations

Name	Type	Description
name	<i>string</i>	Name of the destination.
id	<i>string</i>	Unique identifier for the destination.
mode	<i>string</i>	Destination mode: unicast or multicast.
state	<i>string</i>	Destination state: disconnected, connecting, connection established, or connected.
elapsedRunningTime	<i>string</i>	Either an empty string (for idle routes), or a string in HH:MM:SS format (e.g. 00:03:46).
bitrate	<i>number</i>	Stream bitrate in Mbits/s.

Name	Type	Description
signalLosses	<i>number</i>	Number of signal losses.
usedBandwidth	<i>number</i>	Bandwidth used in Mbits/s.
sendRate	<i>number</i>	Packet send rate in Mbits/s.
numPackets	<i>number</i>	Number of packets.

UDP, RTP, and HLS [SRT](#)

SRT Destinations

Name	Type	Description
name	<i>string</i>	Name of the destination.
id	<i>string</i>	Unique identifier for the destination.
mode	<i>string</i>	Destination SRT mode: caller, listener, or rendezvous.
protocol	<i>string</i>	Protocol: srt.
state	<i>string</i>	Destination state: disconnected, connecting, connection established, or connected.
elapsedRunningTime	<i>string</i>	Either an empty string (for idle routes), or a string in HH:MM:SS format (e.g. 00:03:46).
bitrate	<i>number</i>	Stream bitrate in Mbits/s.
signalLosses	<i>number</i>	Number of signal losses.
usedBandwidth	<i>number</i>	Bandwidth used in Mbits/s.
sendRate	<i>number</i>	Packet send rate in Mbits/s.
numPackets	<i>number</i>	Number of packets.
srtNumLostPackets	<i>number</i>	SRT number of lost (but recovered) packages.
srtPacketLossRate	<i>number</i>	SRT packet loss rate in percent.
srtNumSkippedPackets	<i>number</i>	(Receiver only) Missing packets skipped, because they were not recovered in time.
srtDroppedPackets	<i>number</i>	(Sender only) Number of dropped packets.
srtRoundTripTime	<i>number</i>	SRT round trip time in ms.
srtBufferLevel	<i>number</i>	SRT buffer time in ms.
srtNegotiatedLatency	<i>number</i>	SRT maximum latency in ms.
srtLatency	<i>number</i>	(Receiver only) SRT latency.
srtDecryptionState	<i>string</i>	(Receiver only) SRT receiver decryption state: <empty>, active, initializing, inactive (no passphrase), or inactive (invalid passphrase).
srtPeerDecryptionState	<i>string</i>	(Sender only) SRT peer decryption state: <empty>, active, initializing, inactive (no passphrase), or inactive (invalid passphrase).

Name	Type	Description
srtEncryption	<i>string</i>	(Receiver only) Indicates the cipher used in the received stream: <empty>, AES128, or AES256.
srtMaxBandwidth	<i>number</i>	(Sender only) SRT maximum bandwidth used in Mbits/s.
srtRetransmitRate	<i>number</i>	SRT retransmit rate in bits/s.
srtEstimatedBandwidth	<i>number</i>	(Sender only) SRT estimated path max bandwidth in bits/s.
clientStat	<i>object list</i>	(SRT listener only) Array of route client statistics objects. See below for the client statistics model.
connections	<i>object list</i>	(SRT caller and rendezvous only). Array of destination connections objects. See SRT Statistics Connections Object Model for the definition.


Client Statistics Model

The following client statistics model is returned as part of the response to a [Get Destination Statistics](#) or [Get SRT Client Statistics](#) API command for SRT Listener outputs.

Field	Type	Description
label	<i>string</i>	Listener output label.
address	<i>string</i>	Client address.
port	<i>number</i>	Client connection port.
bitrate	<i>number</i>	Stream bitrate.
signalLosses	<i>number</i>	Number of signal losses.
srtVersion	<i>string</i>	SRT protocol version of the listener.
SRTPeerVersion	<i>string</i>	SRT protocol version of the client.
usedBandwidth	<i>number</i>	Bandwidth used in Mbits/s.
connections	<i>object list</i>	Array of destination connections objects. See SRT Statistics Connections Object Model for the definition.

SRT Statistics Connections Object Model

The following statistics connections model is returned after issuing the [Get Route Statistics](#) and [Get Destination Statistics](#) API commands for SRT routes. The definition depends on the SRT type.

Field	Type	SRT Source	SRT Destination		Description
			Listener Client	Caller/Rendezvous	
SRTPeerVersion	<i>string</i>				Peer SRT protocol version.

Field	Type	SRT Source	SRT Destination		Description
			Listener Client	Caller/Rendezvous	
address	string	✓	✓	✓	Network address.
bitrate	number			✓	Stream bitrate in Mbits/s.
label	string			✓	Label.
localAddress	string	✓	✓	✓	Network local address.
port	number	✓	✓	✓	Network port.
signalLosses	number			✓	Number of signal losses.
localPort	number	✓	✓	✓	Network local port.
networkInterface	string	✓	✓	✓	Network interface.
numPackets	number		✓	✓	Number of packets.

Field	Type	SRT Source	SRT Destination		Description
			Listener Client	Caller/Rendezvous	
srtBufferLevel	number	✓	✓	✓	SRT buffer time in ms.
srtCurrentBandwidth	number	✓	✓	✓	SRT current bandwidth.
srtDecryptionState	string	✓			SRT decryption state: <empty>, active, initializing, inactive (no passphrase), or inactive (invalid passphrase).
srtDroppedPackets	number		✓	✓	Number of dropped packets.
srtDroppedPacketsDiff	number		✓	✓	Number of dropped packets diff.
srtEncryption	string	✓	✓	✓	Indicates the cipher used in the stream: None, none, AES128, or AES256.
srtEstimatedBandwidth	number	✓	✓	✓	SRT estimated bandwidth.
srtFec	string	✓	✓	✓	SRT FEC.
srtFecArq	string	✓	✓	✓	SRT FEC ARQ: always, onreq, or never.

Field	Type	SRT Source	SRT Destination		Description
			Listener Client	Caller/Rendezvous	
srtFecCols	number	✓	✓	✓	SRT FEC columns.
srtFecLayout	string	✓	✓	✓	SRT FEC layout: even or staircase.
srtFecPacketLoss	number	✓			SRT FEC packet loss.
srtFecRecoveredPackets	number	✓			SRT FEC recovered packets.
srtFecRows	number	✓	✓	✓	SRT FEC rows.
srtFecTotalPacketLoss	number	✓			SRT FEC total packet loss.
srtFecTotalRecoveredPackets	number	✓			SRT FEC total recoverd packets.
srtGroupMemberStatus	string	✓	✓	✓	SRT group member status.
srtGroupMemberWeight	number	✓	✓	✓	SRT group member weight.

Field	Type	SRT Source	SRT Destination		Description
			Listener Client	Caller/Rendezvous	
srtGroupMode	string	✓	✓		SRT group mode: <empty>, broadcast, backup, balance, or any.
srtMaxBandwidth	number	✓	✓	✓	SRT maximum bandwidth used in Mbits/s.
srtNegotiatedLatency	number	✓	✓	✓	SRT negotiated latency in ms.
srtNumLostPackets	number	✓	✓	✓	SRT number of lost (but recovered) packets.
srtNumPackets	number	✓	✓	✓	SRT number of pacakets.
srtPacketLossRate	number	✓	✓	✓	SRT packet loss rate in percent.
srtPeerDecryptionState	string		✓	✓	SRT peer decryption state: <empty>, active, initializing, inactive (no passphrase), or inactive (invalid passphrase).
srtRetransmitRate	number	✓	✓	✓	SRT retransmit rate in bits/s.
srtRoundTripTime	number	✓	✓	✓	SRT round trip time in ms.

Field	Type	SRT Source	SRT Destination		Description
			Listener Client	Caller/Rendezvous	
srtSkippedPackets	number	✓			SRT number of skipped packets, because they were not recovered in time.
srtSkippedPacketsDiff	number	✓			SRT number of skipped packets diff.
srtVersion	string			✓	SRT protocol version.
state	string	✓	✓	✓	Source connection state.
usedBandwidth	number			✓	Bandwidth used in Mbits/s.

Warranties

1-Year Limited Hardware Warranty

Haivision warrants its hardware products against defects in materials and workmanship under normal use for a period of ONE (1) YEAR from the date of equipment shipment ("Warranty Period"). If a hardware defect arises and a valid claim is received within the Warranty Period, at its option and to the extent permitted by law, Haivision will either (1) repair the hardware defect at no charge, or (2) exchange the product with a product that is new or equivalent to new in performance and reliability and is at least functionally equivalent to the original product. A replacement product or part assumes the remaining warranty of the original product or ninety (90) days from the date of replacement or repair, whichever is longer. When a product or part is exchanged, any replacement item becomes your property and the replaced item becomes Haivision's property.

EXCLUSIONS AND LIMITATIONS

This Limited Warranty applies only to hardware products manufactured by or for Haivision that can be identified by the "Haivision" trademark, trade name, or logo affixed to them. The Limited Warranty does not apply to any non-Haivision hardware products or any software, even if packaged or sold with Haivision hardware. Manufacturers, suppliers, or publishers, other than Haivision, may provide their own warranties to the end user purchaser, but Haivision, in so far as permitted by law, provides their products "as is".

Haivision does not warrant that the operation of the product will be uninterrupted or error-free. Haivision does not guarantee that any error or other non-conformance can or will be corrected or that the product will operate in all environments and with all systems and equipment. Haivision is not responsible for damage arising from failure to follow instructions relating to the product's use.

This warranty does not apply:

- (a) to cosmetic damage, including but not limited to scratches, dents and broken plastic on ports;
- (b) to damage caused by accident, abuse, misuse, flood, fire, earthquake or other external causes;
- (c) to damage caused by operating the product outside the permitted or intended uses described by Haivision;
- (d) to a product or part that has been modified to alter functionality or capability without the written permission of Haivision; or
- (e) if any Haivision serial number has been removed or defaced.

TO THE EXTENT PERMITTED BY LAW, THIS WARRANTY AND REMEDIES PROVIDED ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, REMEDIES AND CONDITIONS, WHETHER ORAL OR WRITTEN, STATUTORY, EXPRESS OR IMPLIED. AS PERMITTED BY APPLICABLE LAW, HAIVISION SPECIFICALLY DISCLAIMS ANY AND ALL STATUTORY OR IMPLIED WARRANTIES,

INCLUDING, WITHOUT LIMITATION, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND WARRANTIES AGAINST HIDDEN OR LATENT DEFECTS. IF HAIVISION CANNOT LAWFULLY DISCLAIM STATUTORY OR IMPLIED WARRANTIES THEN TO THE EXTENT PERMITTED BY LAW, ALL SUCH WARRANTIES SHALL BE LIMITED IN DURATION TO THE DURATION OF THIS EXPRESS WARRANTY AND TO REPAIR OR REPLACEMENT SERVICE AS DETERMINED BY HAIVISION IN ITS SOLE DISCRETION. No Haivision reseller, agent, or employee is authorized to make any modification, extension, or addition to this warranty. If any term is held to be illegal or unenforceable, the legality or enforceability of the remaining terms shall not be affected or impaired.

EXCEPT AS PROVIDED IN THIS WARRANTY AND TO THE EXTENT PERMITTED BY LAW, HAIVISION IS NOT RESPONSIBLE FOR DIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY BREACH OF WARRANTY OR CONDITION, OR UNDER ANY OTHER LEGAL THEORY, INCLUDING BUT NOT LIMITED TO LOSS OF USE; LOSS OF REVENUE; LOSS OF ACTUAL OR ANTICIPATED PROFITS (INCLUDING LOSS OF PROFITS ON CONTRACTS); LOSS OF THE USE OF MONEY; LOSS OF ANTICIPATED SAVINGS; LOSS OF BUSINESS; LOSS OF OPPORTUNITY; LOSS OF GOODWILL; LOSS OF REPUTATION; LOSS OF, DAMAGE TO OR CORRUPTION OF DATA; OR ANY INDIRECT OR CONSEQUENTIAL LOSS OR DAMAGE HOWSOEVER CAUSED INCLUDING THE REPLACEMENT OF EQUIPMENT AND PROPERTY, ANY COSTS OF RECOVERING, PROGRAMMING, OR REPRODUCING ANY PROGRAM OR DATA STORED OR USED WITH HAIVISION PRODUCTS AND ANY FAILURE TO MAINTAIN THE CONFIDENTIALITY OF DATA STORED ON THE PRODUCT. THE FOREGOING LIMITATION SHALL NOT APPLY TO DEATH OR PERSONAL INJURY CLAIMS, OR ANY STATUTORY LIABILITY FOR INTENTIONAL AND GROSS NEGLIGENT ACTS AND/OR OMISSIONS.

OBTAINING WARRANTY SERVICE

Before requesting warranty service, please refer to the documentation accompanying this hardware product and the Haivision Support Portal <https://support.haivision.com>. If the product is still not functioning properly after making use of these resources, please contact Haivision or Authorized Reseller using the information provided in the documentation. When calling, Haivision or Authorized Reseller will help determine whether your product requires service and, if it does, will inform you how Haivision will provide it. You must assist in diagnosing issues with your product and follow Haivision's warranty processes.

Haivision may provide warranty service by providing a return material authorization ("RMA") to allow you to return the product in accordance with instructions provided by Haivision or Authorized Reseller. You are fully responsible for delivering the product to Haivision as instructed, and Haivision is responsible for returning the product if it is found to be defective. Your product or a replacement product will be returned to you configured as your product was when originally purchased, subject to applicable updates. Returned products which are found by Haivision to be not defective, out-of-warranty or otherwise ineligible for warranty service will be shipped back to you at your expense. All replaced products and parts, whether under warranty or not, become the property of Haivision. Haivision may require a completed pre-authorized form as security for the retail price of the replacement product. If you fail to return the replaced product as instructed, Haivision will invoice for the pre-authorized amount.

APPLICABLE LAW

This Limited Warranty is governed by and construed under the laws of the Province of Quebec, Canada.

This Limited Hardware Warranty may be subject to Haivision's change at any time without prior notice.

EULA - End User License Agreement

READ BEFORE USING

THE LICENSED SOFTWARE IS PROTECTED BY COPYRIGHT LAWS AND TREATIES. READ THE TERMS OF THE FOLLOWING END USER (SOFTWARE) LICENSE AGREEMENT ("AGREEMENT") CAREFULLY BEFORE ACCESSING THE LICENSED SOFTWARE. BY SCANNING THE QR CODE TO REVIEW THIS AGREEMENT AND/OR ACCESSING THE LICENSED SOFTWARE, YOU CONFIRM YOUR ACCEPTANCE OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THESE TERMS, HAIVISION IS UNWILLING TO LICENSE THE LICENSED SOFTWARE TO YOU AND YOU ARE NOT AUTHORIZED TO ACCESS THE LICENSED SOFTWARE.

Click the following link to view the Software End-User License Agreement: [Haivision EULA.pdf](#)

If you have questions, please contact legal@haivision.com

SLA - Service Level Agreement

1. Introduction

This Service Level and Support supplement forms a part of and is incorporated into the Service Agreement (the "Agreement") between You and Haivision Network Video Inc. ("Haivision"). Capitalized terms used but not otherwise defined in this supplement shall have the meaning ascribed to them in the Agreement. Haivision may, upon prior written notice to You, amend this supplement to incorporate improvements to the service levels and support commitments at no additional cost to You. This supplement applies only to those products and services set forth below.

2. Definitions

- "Audience Member" means an individual or entity that accesses Your Published Media Objects through a public URL.
- "Access Service" means the service provided by Haivision VCMS that verifies an Audience Member's credentials.
- "Digital Media File" means a computer file containing text, audio, video, or other content.
- "Outage" is a 12-minute period of consecutive failed attempts by all six agents to PING the domain on the Haivision Streaming Media network.
- "Published Media Object" means a Digital Media File with a public URL.
- "Transaction" means the creation of a right for an Audience Member to access a Media Object and the completion of an order logged in the order history service.

3. Service Levels for the Video Content Management System

The service levels in this [Section 3](#) apply only to the hosted version of Haivision VCMS and the Haivision VCMS development kit (collectively, the "Standard Hosted Components" of Haivision Video Cloud Services). Subject to the exceptions noted in [Section 4](#) below, the aforementioned components of Haivision Video Cloud Services will be available for use over the course of each calendar month as follows:

Type of Access	Definition	Availability Level
Write Functions	<ul style="list-style-type: none"> • Access to all functions through the administrative user interface. • Ability to add or modify objects and metadata through the application programming interface (“API”) • Ability of ingest service to check for new or updated files or feeds 	99.999%
Read-Only Functions	<ul style="list-style-type: none"> • Ability to retrieve data through the API • Ability for Audience Members to authenticate through the Access Service • Ability for Audience Members to play Published Media Objects • Ability for Audience Members to play Haivision VCMS-authenticated or entitled Published Media Objects • Ability to complete Transactions 	99.999%

4. Exceptions to Availability for the VCMS

The Standard Hosted Components may not be available for use under the following circumstances, and in such case such periods of unavailability shall not be counted against Haivision Video Cloud for purposes of calculating availability:

- a. Normal Maintenance, Urgent Maintenance and Upgrades as defined in the table below;
- b. Breach of the Agreement by You as defined in the Agreement;
- c. The failure, malfunction, or modification of equipment, applications, or systems not controlled by Haivision Video Cloud;
- d. Any third party, public network, or systems unavailability;
- e. Acts of Force Majeure as defined in the Agreement;
- f. Modification of software made available to You as part of Haivision Video Cloud Services by You or a third party acting on Your behalf; and
- g. Any third party product or service not incorporated into Haivision Video Cloud Services or any third party plug-in.

Haivision Video Cloud shall make commercially reasonable efforts to notify, or work with, applicable third parties to repair or restore Haivision VCMS functionality affected by such exceptions.

Type of Maintenance	Purpose	Write Functions Available	Read Functions Available	Maximum Time Per Month	Continuous Time in Mode (Max)	Window (Central Time)	Min Notice
Normal	<ul style="list-style-type: none"> • Preventive maintenance on the software/hardware components of Haivision VCMS • Addition of new features/functions • Repair errors that are not immediately affecting Your use of Haivision VCMS 	No	Yes	10 Hours	6 Hours	10:00p m - 5:00a m	48 Hours
Urgent	<ul style="list-style-type: none"> • Repair errors that are immediately affecting Your use of Haivision VCMS 	No	Yes	30 Minutes	15 Minutes	Any Time	3 Hours

Type of Maintenance	Purpose	Write Functions Available	Read Functions Available	Maximum Time Per Month	Continuous Time in Mode (Max)	Window (Central Time)	Min Notice
Upgrades	<ul style="list-style-type: none"> Perform upgrades on software or hardware elements necessary to the long term health or performance of Haivision VCMS, but which, due to their nature, require that certain components of Haivision VCMS to be shut down such that no access is possible 	No	No	1 Hour	1 Hour	12:00am - 4:00am M-F	5 Days

5. Credits for Downtime for the VCMS

Haivision Video Cloud will grant a credit allowance to You if You experience Downtime in any calendar month and you notify Haivision Video Cloud thereof within ten (10) business days after the end of such calendar month. In the case of any discrepancy between the Downtime as experienced by You and the Downtime as measured by Haivision Video Cloud, the Downtime as measured by Haivision Video Cloud shall be used to calculate any credit allowance set forth in this section. Such credit allowance shall be equal to the pro-rated charges of one-half day of Fees for each hour of Downtime or fraction thereof. The term “Downtime” shall mean the number of minutes that Standard Hosted Components are unavailable to You during a given calendar month below the availability levels thresholds in [Section 3](#), but shall not include any unavailability resulting from any of the exceptions noted in [Section 4](#). Within thirty (30) days after the end of any calendar month in which Downtime occurred below the availability levels thresholds in [Section 3](#), Haivision Video Cloud shall provide You with a written report detailing all instances of Downtime during the previous month. Any credit allowances accrued by You may be offset against any and all Fees owed to Haivision Video Cloud pursuant to the Agreement, provided that a maximum of one month of credit may be accrued per month.

6. Support Services for the VCMS

Support for Haivision Video Cloud Services as well as the Application Software (defined as the VCMS application software components that Haivision licenses for use in conjunction with the Video Cloud Services) can be reached at hvc-techsupport@haivision.com and shall be available for all Your support requests. Haivision Video Cloud will provide 24x7 monitoring of the Standard Hosted Components.

Cases will be opened upon receipt of request or identification of issue, and incidents will be routed and addressed according to the following:

Severity Level	Error State Description	Status Response Within	Incident Resolution within
1 - Critical Priority	Renders Haivision VCMS inoperative or causes Haivision VCMS to fail catastrophically.	15 minutes	4 hours
2 - High Priority	Affects the operation of Haivision VCMS and materially degrades Your use of Haivision VCMS.	30 minutes	6 hours
3 - Medium Priority	Affects the operation of Haivision VCMS, but does not materially degrade Your use of Haivision VCMS.	2 hours	12 hours

Severity Level	Error State Description	Status Response Within	Incident Resolution within
4 - Low Priority	Causes only a minor impact on the operation of Haivision VCMS.	1 business day	3 business days

7. Service Levels for Haivision Streaming Media Service

Haivision agrees to provide a level of service demonstrating 99.9% Uptime. The Haivision Streaming Media Service will have no network Outages.

The following methodology will be employed to measure Streaming Media Service availability:

Agents and Polling Frequency

- a. From six (6) geographically and network-diverse locations in major metropolitan areas, Haivision’s Streaming Media will simultaneously poll the domain identified on the Haivision Streaming Media network.
- b. The polling mechanism will perform a PING operation, sending a packet of data and waiting for a reply. Success of the PING operation is defined as a reply being received.
- c. Polling will occur at approximately 6-minute intervals.
- d. Based on the PING operation described in (b) above, the response will be assessed for the purpose of measuring Outages.

If an Outage is identified by this method, the customer will receive (as its sole remedy) a credit equivalent to the fees for the day in which the failure occurred.

Haivision reserves the right to limit Your use of the Haivision Streaming Media network in excess of Your committed usage in the event that Force Majeure events, defined in the Agreement, such as war, natural disaster or terrorist attack, result in extraordinary levels of traffic on the Haivision Streaming Media network.

8. Credits for Outages of Haivision Streaming Media Service

If the Haivision Streaming Media network fails to meet the above service level, You will receive (as your sole remedy) a credit equal to Your or such domain’s committed monthly service fee for the day in which the failure occurs, not to exceed 30 days of fees.

9. No Secondary End User Support

UNDER NO CIRCUMSTANCES MAY YOU PROVIDE CONTACT INFORMATION FOR HAIVISION SERVICES TO CUSTOMERS OR AUDIENCE MEMBERS OR OTHER THIRD PARTIES WITHOUT HAIVISION’S EXPRESS PRIOR WRITTEN CONSENT.

Getting Help

<p>General Support</p>	<p>North America (Toll-Free) 1 (877) 224-5445</p> <p>International 1 (514) 334-5445</p> <p><i>and choose from the following:</i> Sales - 1, Cloud Services - 3, Support - 4</p>
<p>Managed Services</p>	<p>U.S. and International 1 (512) 220-3463</p>
<p>Fax</p>	<p>1 (514) 334-0088</p>
<p>Support Portal</p>	<p>https://support.haivision.com</p>
<p>Product Information</p>	<p>info@haivision.com</p>

