HAIVISION

Kraken 3.0

REST API v1.0 Integrator's Guide

HVS-ID-INT-KRAK-30, Issue 01

Edition Notice

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

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

About This Document

Conventions

The following conventions are used to help clarify the content.

Typographic Conventions and Elements

Italics	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.

This guide describes the Representational State Transfer (REST) Application Programming Interface (API) v1.0 functions that can be used to interface third party management systems with the Kraken Video Encoder/Transcoder.

Important

Kraken release 3.0 introduces API v2.0, a modern REST API stack that provides all functionality from the Kraken Web GUI and is also harmonized with other Haivision appliances.
API v1.0 is still functioning and is not changed with this release. However, it is anomalous when compared to the other Haivision product APIs and is not used by the Kraken Web GUI.
The API Reference in this guide only covers API v1.0. To access the API v2.0 endpoint documentation, see **Accessing the REST API** (in the **Kraken User's Guide**).

Introducing the Kraken API

Haivision's Kraken™ Application Programming Interface (API) provides a means for third parties to create their own products that integrate with Kraken applications. The Kraken API is a Representational State Transfer (REST) API.

Note

To help keep applications stable with future versions of the API, please allow for, and ignore, unknown XML elements. When expanding the API, it may be necessary for Haivision to add elements to existing XML elements (without changing existing elements).

Tip

All communication with the REST API is done through the portal server. Your base URI for requests should match the root of your portal server. In the examples, replace "https://example.haivision.com/" with the address of your portal server.

Topics Discussed

REST 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

URI Structure

Note

`"foobar"` is a placeholder name intended to represent whatever is being discussed.

Method	URI	Notes
GET	/foobars	Returns a collection of the foobar entities. By default, items in the list are a minimal representation of a foobar entity. Note that we use the plural for the directory name.
POST	/foobars	Creates a foobar entity and returns a link to entity in the form /foobars/foobar- {id}.
GET	/foobars/foobar-{id}	Returns the full content of the foobar identified by the given id. Note that we use the singular for the entity name.
PUT	/foobars/foobar-{id}	Updates the contents of a foobar entity.
DELETE	/foobars/foobar-{id}	Deletes the foobar entity.

Sub-elements of a foobar entity are made available as sub-resources of /foobars/foobar- {id}, e.g.:

```
/foobars/foobar-{id}/bazs/baz-{id}/bloops/bloop-{id}
```

Note

The ID in each URI comes from the collection preceding it. When a resource contains multiple IDs, the notation does not imply that the IDs are identical. Refer to the collection to get the ID.

Required Authorization

The Kraken API v1.0 uses the OAuth (Open Authorization) standard for authorization when a third party application requests access. As defined in *The OAuth 1.0 Protocol* abstract:

> "OAuth provides a method for clients to access server resources on behalf of a resource owner (such as a different client or an end-user). It also provides a process for end-users to authorize third-party access to their server resources without sharing their credentials (typically, a username and password pair), using user-agent redirections."

OAuth is a standardized authentication mechanism that works by signing the HTTPS request using a shared secret. Kraken uses a two-legged implementation to control which applications can use the API. Two-legged OAuth does not provide user authentication, it only validates an application's identity.

Implementing OAuth

Implementing OAuth for the Kraken API (v1.0)is relatively simple and straightforward. However, it requires that both the server and client side behave the same way. Therefore, it is important to avoid even minor mistakes, which can lead to authentication errors. The following instructions provide an overview of the signature process. A great resource for understanding OAuth in action is the *RESTClient for Firefox*, listed below.

Note

Usage of Oauth with the Kraken API requires that calls be sent via HTTPS protocol.

OAuth Informational Links

- Official Site (Official Site)
- Oauth RFC (Official spec)
- Authoritative Guide to OAuth

- OAuth Libraries
- RESTClient for Firefox that supports OAuth (Only fill in consumer key and consumer secret to authenticate)

Preparing for OAuth

Before you can use the Kraken API (v1.0), you need to perform two steps from the Web Interface.

1. Enable API access.
2. Because OAuth uses a key pair authentication mechanism, you need to generate the credential (i.e., a key and secret pair).

For details, please refer to "Accessing the REST API" in the Kraken User's Guide.

When you have retrieved this API credential, proceed to the next step.

Generating the Request Base String

The next step is to generate OAuth headers.

1. Generate OAuth parameters:
   a. Generate a random nonce and store it as `oauth_nonce`.
   b. Generate a timestamp and store it as `oauth_timestamp`.
   c. Set `oauth_consumer_key` to the Consumer Key retrieved from the Web interface (see Preparing for OAuth).
   d. Set `oauth_signature_method` to "HMAC-SHA1". (No other methods are currently supported.)
2. Gather all parameters:
   - OAuth parameters
   - GET parameters
   - POST parameters
3. Encode the parameters using UTF-8 standards/functions.
4. Encode the parameters using URL standards/functions.
5. Normalize parameters (sort parameters alphabetically per http://tools.ietf.org/html/rfc5849#section-3.4.1.3.2).
6. Concatenate parameters together with an ampersand (&) between each, similar to HTTP GET requests.

REST API Responses

Responses to a request consist of two elements: the HTTP status code and the response content. An application can act initially upon the HTTP status code (sensing success or failure) and then act specifically upon the data of the response content.

Response content is usually returned as application/xml data, with the root level of `<response>`. Within the `<response>`, the content is context-specific. Individual API functions specify the type of response content later in this documentation.

If there is a problem processing or executing the request, the response content may contain an `<error>` element with a more application-specific error code.

For more details on the HTTP and error responses, see Error Codes.

Topics Discussed

Example Request and Success Response

The following request retrieves details about a specific transcoder:

```
GET https://example.haivision.com/apis/kraken/transcoders/transcoder-88ca6209-ca59-4226-a72c-2012231bf915
```

An example successful response is:

```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
  <transcoders>
    <transcoder>
      <name>SD-1</name>
      <resolution>
        <width>720</width>
        <height>480</height>
      </resolution>
      <frameRate>2</frameRate>
      <videoBitrate>2000</videoBitrate>
      <gopSize>30</gopSize>
      <metadata>off</metadata>
      <audio>on</audio>
      <audioBitrate>128</audioBitrate>
      <videoType>avc</videoType>
      <bFrame>-1</bFrame>
      <link rel="self" type="application/xml" href="https://10.6.60.203/
 apis/kraken/transcoders/SD-1" />
    </transcoder>
    <transcoder>
      <name>TR-2</name>
      <frameRate>2</frameRate>
      <videoBitrate>3000</videoBitrate>
      <gopSize>30</gopSize>
      <metadata>off</metadata>
      <audio>on</audio>
      <videoType>avc</videoType>
      <bFrame>-1</bFrame>
      <link rel="self" type="application/xml" href="https://10.6.60.203/
 apis/kraken/transcoders/TR-2" />
    </transcoder>
  </transcoders>
</response>
```

Example Error Response

The following is an example error response for a request with improperly formatted XML data:

```
HTTP/1.1 400 Bad Request
Content-Type: application/xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<response>
  <error>
    <code>1011</code>
```

```
    <message>Input XML data is poorly formatted</message>
  </error>
</response>
```

Sorting Response Content

Please note that the contents of a given container (for example, the tags within a <input> element, or the `<input>` elements inside a parent <inputs> element) are not guaranteed to be returned in any particular order.

To avoid unnecessary errors, do not assume that the order of elements follow those in the example response. If necessary, your application must sort the response.

XML Entities

> **Note**
>
> `{foobarID}` is used throughout these examples to denote the unique identifiers referenced within the XML data. Keep in mind that this is not the syntax used in the actual results for these elements.

In general, a request for a list of resources (`/apis/resources`) returns XML such as:

```
<resources>
  <resource>
    <id>abc</id>
  </resource>
  <resource>
    <id>xyz</id>
  </resource>
</resources>
```

A request for a single resource (`/apis/resources/resource-xyz`) returns XML such as:

```
<resource>
  <id>xyz</id>
</resource>
```

Generic entities you may come across are as follows:

```
<error>
  <code>1011</code>
  <message>Input XML data is poorly formatted</message>
</error>
```

```
<link rel="self" type="application/xml" href=
  "https://example.haivision.com/apis/kraken/inputs/input-{inputID}"/>
```

- `rel` : describes the relationship of the link to the current entity. Values vary depending on context.
- `type` : Content-Type of the linked data.

- `href`: REST-navigable link to the indicated entity.

API Reference

This API command reference lists and describes the available resources for the Kraken Video Encoder/Transcoder API.

Topics Discussed

Summary of Kraken API Resources

The Kraken API consists of the following resources:

API Resource	Description
Input Resources	Defines the input source (either a source URL, an SRT source, or an SDI or Analog Composite input).
Output Resources	Defines one or more output URLs.
Transcoder Resources	Defines audio and video characteristics to change in the outbound stream.
Stream Resources	**Note** Passthru stream routing option is available as of Version 2.5.
Configuration Resources	Saves, loads, deletes, or gets information about configurations in the system.
System Resources	Gets the Kraken version number and capture card information.
Server Resources	Gets information for the Kraken physical server and Network Interface Cards (NICs).

For a basic description of the XML entities referenced in these sections, see XML Entities.

Endpoints vs. Methods

To use this reference, keep in mind the following definitions:

- An **endpoint** is a URI (Uniform Resource Identifier) that points to a function or operation provided by the API, e.g., `/apis/kraken/inputs`.

- A method, for the purposes of this document, refers to the HTTP methods: GET, POST, PUT, or DELETE. An HTTP method acts on a Kraken API endpoint.

For a glossary of terms used in this document, see **Glossary**.

API Endpoint Reference

API Endpoint and Methods
/apis/kraken/inputs • GET — Get a List of Inputs • POST — Create a New Input
/apis/kraken/inputs/input-{id} • GET — Retrieve Information for a Specific Input • PUT — Edit an Input • DELETE — Remove an Input
/apis/kraken/outputs • GET — Get a List of Outputs • POST — Create a New Output
/apis/kraken/outputs/output-{id} • GET — Retrieve Information for a Specific Output • PUT — Edit an Output • DELETE — Remove an Output
/apis/kraken/transcoders • GET — Get a List of Transcoders • POST — Create a New Transcoder
/apis/kraken/transcoders/transcoder-{id} • GET — Retrieve Information for a Specific Transcoder • PUT — Edit a Transcoder • DELETE — Remove a Transcoder
/apis/kraken/streams • GET — Get a List of Streams • POST — Create a New Stream
/apis/kraken/streams/stream-{id} • GET — Retrieve Information for a Specific Stream • PUT — Edit, Start, or Stop a Stream • DELETE — Remove a Stream
/apis/kraken/configurations
• GET — Get a List of Configurations • POST — Create a New Resource
/apis/kraken/configurations/configuration-{id} • GET — Retrieve Information for a Specific Configuration • PUT — Edit a Configuration • DELETE — Remove a Configuration
/apis/kraken/system • GET — Get System Information
/apis/servers • GET — Get List of Servers
/apis/servers/server-{id} • GET — Get Specific Server Information
/apis/servers/server-{id}/nics • GET — Get List of Server NICs
/apis/servers/server-{id}/nics/nic-{id} • GET — Get Specific NIC Information

Syntax Conventions

The following syntax conventions are used in this reference:

Convention	Description
{ braces }	Indicates a placeholder for the name of a resource, e.g., ID part of a URL.
< >	Delineates an XML tag. The left angle bracket begins an XML element. The right angle bracket ends an XML element.
/	A slash before an element name ends the element definition.
/>	A slash followed by a right angle bracket ends the response element.
[ ]	Square brackets indicate optional elements or arguments.
x | y	A vertical bar separates items in a list of options from which you must select one. If options are not separated by |, you may use combinations.

Input Resources

The inputs API allows you to define and get information about the input source, either a source URL, an SRT source, or an SDI or analog composite input.

Input Properties and API Commands:

Input Properties

The properties within the input element used throughout this section are defined below. Additional properties are available depending on the type of stream.

General | DECKLINK streamType | V4L2 streamType | SRT streamType

General

Property	Description and Values
uuid	Identifies the input source, generated by Kraken.
name	Name of the input source.
url	The source URL for the input. For example, udp://239.100.100.100:5000. If streamType is RTSP, rtsp://hostname/path.
description	Description of the input.
streamType	The source type for the input. Possible values are: • MPEG2TS: MPEG2 Transport Stream over UDP (no RTP header, default). • SRT: Haivision's Secure Reliable Transport (TS over SRT). • DECKLINK: (Kraken Mini or SDI capture card must be installed) DeckLink Micro Recorder 1: Select to capture HD/SD-SDI video for baseband input encoding. • V4L2: (Kraken CR or Analog Composite capture card must be installed) Analog Capture 1: Select to capture Analog Composite Video for baseband input encoding. • RTSP: Select to configure Kraken to interoperate with ISR "sensors" such as wearable IP cameras, which are typically H.264 RTP/RTSP. • MJPEGRAW: Allows you to input a Motion JPEG (MJPEG) live stream and transcode the payload into a standard H.264 video within an MPEG Transport Stream.

Property	Description and Values
interface	Select the network (Ethernet) interface for the input, either: • Auto (default): Uses static route, if defined; otherwise uses the default • eth0 • **Note** Network Interface names for Ethernet interfaces may vary, such as eth0/eth1/..., pNp1/pNp2/..., or em1/em2/....
state	Indicates if the input is active or not: active, stopped, waiting, or unset (not part of a stream).

General DECKLINK streamType V4L2 streamType SRT streamType

DeckLink streamType

Property	Description and Values
decklinkPort	Specifies the capture card port: • 0: First port/card • 1+: More than one port/card
decklinkMode	Specifies the capture card mode: • -1: Auto resolution (default) • **Tip** Use Get System Information to discover port information and available modes if not using Auto.

General DECKLINK streamType V4L2 streamType SRT streamType

V4L2 streamType

Property	Description and Values
capturePort	Specifies the capture card port: • 0: First port/card • 1+: More than one port/card
captureMode	Specifies the capture card mode: • -1: Auto resolution (default) • **Tip** Use Get System Information to discover port information and available modes if not using Auto .

General DECKLINK streamType V4L2 streamType SRT streamType

SRT streamType

Property	Description and Values
srtMode	The SRT connection mode: • caller: Kraken acts like a client and connects to a server listening and waiting for an incoming call. • listener: Kraken acts like a server and listens and waits for clients to connect to it. • rendezvous: Allows calling and listening at the same time.
srtAddress	**Tip** You can also enter a Fully Qualified Domain Name (FQDN).
srtRemotePort	(srtMode must be caller or rendezvous) The UDP destination port for the SRT stream.
srtLocalPort	(srtMode must be listener) The UDP local port for the SRT stream.
srtPassphrase	(srtEncryption must be enabled on the output, see **Output Properties**) Specifies a string used to generate the encryption keys to protect the stream. Range: 10–79 UTF8 characters
srtLatency	Specifies the SRT receiver buffer that permits lost packet recovery. The size of this buffer adds up to the total latency. A minimum value must be 3 times the round-trip-time (RTT). The SRT buffer, configured by this value, is the time reserved in the decoder to recover missing packets. **Note** Latency does not include the capture, encoding, decoding, and display processes of the end-point devices.

Get a List of Inputs

Gets the list of inputs on the device. A successful response includes an inputs element containing multiple input elements.

Authorizations: All

Requests

```
GET https://example.haivision.com/apis/kraken/inputs
```

HTTP Return Codes

Code	Description
200	Results found.
404	No results found.

Example Response

```
<response>
  <inputs>
    <input>
      <uuid>f7615c6b-830d-4e73-a132-b315b2c29051</uuid>
      <name>China Lake</name>
      <link rel="self" type="application/xml" href= "https://localhost:20343/
      apis/kraken/inputs/input-f7615c6b-830d-4e73-a132-b315b2c29051" />
    </input>
    <input>
      <uuid>e2c41f15-369a-4b0d-b772-c4142defd83d</uuid>
      <name>bluray</name>
      <link rel="self" type="application/xml" href= "https://localhost:20343/
      apis/kraken/inputs/input-e2c41f15-369a-4b0d-b772-c4142defd83d" />
    </input>
  </inputs>
</response>
```

Create a New Input

Creates a new input. The body of the request must contain an input element, with the included contents dependent on the input type. The response data returns a link to the newly created input for use in the following commands: Retrieve Information for a Specific Input, Edit an Input, Remove an Input.

Authorizations: All

Requests

Stream Input | Capture Card Input | DeckLink Input

Stream Input

> **Note**
>
> The <url> element is required.

```
POST https://example.haivision.com/apis/kraken/inputs

<input>
  <name>Another input</name>
  <url>udp://239.207.1.3:9002</url>
  <description>Another note</description>
  <streamType>MJPEGRAW</streamType>
  <interface>https://10.6.60.202/apis/servers/serverbc305be293d3/
  nics/nic-eth1</interface>
</input>
```

> **Note**
>
> Populate the <interface> element with a full link to the network interface receiving the stream. Retrieve the link to use by issuing a Get List of Server NICs call.

Stream Input | Capture Card Input | DeckLink Input

Capture Card Input

```
POST https://example.haivision.com/apis/kraken/inputs

<input>
  <name>CAPTURE CARD</name>
  <streamType>V4L2</streamType>
  <capture_card>0</capture_card>
  <capture_mode>-1</capture_mode>
  <description>from channel 2</description>
</input>
```

Note

Use Get System Information to retrieve the <capture_card> and <capture_mode> element values.

Stream Input | Capture Card Input | DeckLink Input

DeckLink Input

```
POST https://example.haivision.com/apis/kraken/inputs

<input>
  <name>DECKLINK</name>
  <streamType>DECKLINK</streamType>
  <decklinkPort>0</decklinkPort>
  <decklinkMode>7</decklinkMode>
  <description>live camera</description>
</input>
```

Note

Use Get System Information to retrieve the <decklinkPort> and <decklinkMode> element values.

HTTP Return Codes

Code	Description
201	Created.
400	Bad Request.
500	Internal server error.

Response

```
<response>
  <link rel="self" type="application/xml" href= "https://10.6.60.202/apis/
  kraken/inputs/input-aaef0642-47ce-4935-8a49-ce6f2d5d1257" />
</response>
```

Retrieve Information for a Specific Input

Retrieves information for a specific input. Return data includes an <input> element.

Authorizations: All

Requests

```
GET https://example.haivision.com/apis/kraken/inputs/input-%7Bid%7D
```

HTTP Return Codes

Name	Description
200	Results found
404	Unknown ID

Response

Stream Input V4L2 SRT DeckLink

Stream Input

```
<response>
  <input>
    <uuid>f5684714-5ee6-4523-ab73-c61773769b53</uuid>
    <name>Makito Input 1</name>
    <url>udp://239.19.3.100:4900</url>
    <description>Makito 1 Input on Direct TV Feed.</description>
    <streamType>MJPEGRAW</streamType>
    <interface>https://10.6.60.202/apis/servers/server-bc305be293d3/
    nics/nic-eth1</interface>
    <state>stopped</state>
    <link rel="self" type="application/xml" href="https://example.haivision.com
    /apis/kraken/inputs/input-f5684714-5ee6-4523-ab73-c61773769b53" />
  </input>
</response>
```

Stream Input V4L2 SRT DeckLink

V4L2 Input

```
<response>
  <input>
    <uuid>8fb95969-73f5-4f79-86e2-a0063f0ef627</uuid>
    <name>Haivision2</name>
    <streamType>V4L2</streamType>
    <url></url>
    <description>Analog capture</description>
    <state>unset</state>
    <link rel="self" type="application/xml" href= "https://10.66.133.141/apis/
    kraken/inputs/input-8fb95969-73f5-4f79-86e2-a0063f0ef627" />
  </input>
</response>
```

Stream Input V4L2 SRT DeckLink

SRT Input

```
<response>
  <input>
    <uuid>8fb95969-73f5-4f79-86e2-a0063f0ef627</uuid>
    <name>airspace 356</name>
    <streamType>SRT</streamType>
    <srtMode>rendezvous</srtMode>
    <srtAddress>10.65.11.111</srtAddress>
    <srtRemotePort>9000</srtRemotePort>
    <srtLocalPort>9000</srtLocalPort>
    <srtPassphrase>serh21323402fsd0342</srtPassphrase>
    <srtLatency>125</srtLatency>
    <description>geoloc @46.3183348,-79.4469281</description>
    <state>unset</state>
    <link rel="self" type="application/xml" href="https://10.66.133.141/apis/
    kraken/inputs/input-8fb95969-73f5-4f79-86e2-a0063f0ef627" />
  </input>
</response>
```

Stream Input V4L2 SRT DeckLink

DeckLink Input

```
<response>
  <input>
    <uuid>85195d65-1100-4446-b14b-9a6e17bf2c1f</uuid>
    <name>capture</name>
    <streamType>DECKLINK</streamType>
    <decklinkPort>0</decklinkPort>
    <decklinkMode>7</decklinkMode>
    <state>unset</state>
    <link rel="self" type="application/xml" href="https://10.66.133.141/apis/
    kraken/inputs/input-8fb95969-73f5-4f79-86e2-a0063f0ef627" />
  </input>
</response>
```

Note

Only fields which are set are returned. For example, if `<description>` is not set, the element is not returned. Required fields, such as `<url>` for input, are always returned since they always have a value associated with them.

Edit an Input

Edits an `<input>` resource. Like the Create a New Input call, an input element is required in the request body, and the response returns a `<link>` element to updated resource.

Authorizations: All

Requests

See Create a New Input for examples of other possible input types.

```
PUT https://example.haivision.com/apis/kraken/inputs/input-{id}
<input>
  <name>New name</name>
  <url>udp://239.207.1.4:9004</url>
  <description>New note</description>
  <streamType>MPEG2TS</streamType>
  <interface>https://10.6.60.202/apis/servers/serverbc305be293d3/nics/
 nic-eth0</interface>
</input>
```

HTTP Return Codes

Name	Description
200	Input successfully updated
400	Bad Request
404	Unknown ID

Response

```
<response>
  <link rel="self" type="application/xml" href= "https://10.6.60.202/apis/
  kraken/inputs/input-aaef0642-47ce-4935-8a49-ce6f2d5d1257" />
</response>
```

Remove an Input

Removes an `<input>` resource. The response data includes a `<link>` element pointing to the inputs collection.

Authorizations: All

Requests

```
DELETE https://example.haivision.com/apis/kraken/inputs/input-{id}
```

HTTP Return Codes

Code	Description
200	Input successfully deleted.
400	Bad Request.
404	Unknown ID.
500	Server error.

Response

```
<response>
  <link rel="self" type="application/xml" href="https://example.haivision.com/
  apis/kraken/inputs" />
</response>
```

Output Resources

The outputs API allows you to define and get information about one or more output URLs.

Output Properties and API Commands:

Output Properties

The properties within the output element used throughout this section are defined below. Additional properties are available depending on the type of stream.

General | Default streamType | SRT streamType

General

Property	Description and Values
uuid	Identifies the output. Generated by Kraken.
name	Name of the output.
description	Description of the output.
streamType	The source type for the output, either leave empty (default) or: • SRT: Haivision's Secure Reliable Transport (TS over SRT).
state	Indicates if the input is active or not: active, stopped, waiting, or unset (not part of a stream).

General | Default streamType | SRT streamType

Default streamType

Property	Description and Values
url	The URL for the output. Examples of supported output formats: • `udp://239.100.100.100:4900` multicast UDP on 239.100.100.100 port 4900 • `udp://10.1.10.10:4900` sends unicast UDP to host 10.1.10.10 on port 4900
mtu	(Maximum Transmission Unit Size) Specifies the maximum allowed size of IP packets for the outgoing data stream. Range: 228–1500, Default: 1490

Property	Description and Values
ttl	(Time-to Live for stream packets) Specifies the number of router hops the Stream packet is allowed to travel/pass before it must be discarded. Range: 1–255, Default: 64
tos	(Type of Service) Specifies the desired quality of service (QoS). This value will be assigned to the Type of Service field of the IP Header for the outgoing streams. Range: 0–255 (decimal) or 0x00–0xFF (hex), Default: 184

General Default streamType SRT streamType

SRT streamType

Property	Description and Values
srtMode	The SRT Connection Mode: • caller: Kraken acts like a client and connects to a server listening and waiting for an incoming call. • listener: Kraken acts like a server and listens and waits for clients to connect to it. • rendezvous: Allows calling and listening at the same time.
srtAddress	**Tip** You can also enter a Fully Qualified Domain Name (FQDN).
srtRemotePort	(srtMode must be caller or rendezvous) The UDP destination port for the SRT stream.
srtLocalPort	(srtMode must be listener) The UDP local port for the SRT stream.
srtPassphrase	(srtEncryption must be enabled) Specifies a string used to generate the encryption keys to protect the stream. Range: 10–79 UTF8 characters
srtEncryption	Enables AES encryption and specifies the key length, either: None, aes128, or aes256
srtLatency	Specifies the SRT receiver buffer that permits lost packet recovery. The size of this buffer adds up to the total latency. A minimum value must be 3 times the round-trip-time (RTT). The SRT buffer, configured in this property, is the time reserved in the decoder to recover missing packets. Range: 20–8000 ms **Note** Latency is for the SRT protocol only and does not include the capture, encoding, decoding and display processes of the end-point devices.

Get a List of Outputs

Gets the list of outputs. The return data includes an `<outputs>` element containing multiple `<output>` elements. Each output element contains a link for use in the following commands: **Retrieve Information for a Specific Input**, **Edit an Output**, and **Remove an Output**

Authorizations: All

Requests

```
GET https://example.haivision.com/apis/kraken/outputs
```

HTTP Return Codes

Code	Description
200	Results found
404	No results found

Response

```
<response>
  <outputs>
    <output>
      <uuid>fcc7f606-c331-427c-bff9-644f8aa62d11</uuid>
      <name>Out 1</name>
      <link rel="self" type="application/xml" href="https://localhost:20343/
    apis/kraken/outputs/outputfcc7f606-c331-427c-bff9-644f8aa62d11" />
    </output>
    <output>
      <uuid>f6e11008-6a04-4172-9a85-653c081b8871</uuid>
      <name>Out 2</name>
      <link rel="self" type="application/xml" href="https://localhost:20343/
    apis/kraken/outputs/outputf6e11008-6a04-4172-9a85-653c081b8871" />
    </output>
  </outputs>
</response>
```

Create a New Output

Creates a new output. The body of the request must contain an output element, with a required url element. The response data returns a link to the newly created input for use in the following commands: Retrieve Information for a Specific Output, Edit an Output, and Remove an Output.

Authorizations: All

Requests

```
POST https://example.haivision.com/apis/kraken/outputs
<output>
  <name>Another output</name>
  <url>udp://239.202.1.3:4900</url>
  <description>A description</description>
  <mtu>1442</mtu>
  <ttl>16</ttl>
  <tos>128</tos>
  <sap>
    <transmitSap>on</transmitSap>
    <address>224.2.127.254</address>
    <port>9875</port>
    <sessionName>A name</sessionName>
    <sessionDescription>A description</sessionDescription>
    <keywords>A keyword</keywords>
    <author>An author</author>
  </sap>
  <interface>https://10.6.60.202/apis/servers/serverbc305be293d3/nics/
 nic-eth1</interface>
</output>
```

Note

Populate the <interface> element with a full link to the network interface receiving the stream. Retrieve the link to use by issuing a Get List of Server NICs call.

HTTP Return Codes

Name	Description
201	Created
400	Bad Request
500	Server error

Response

```
<response>
  <link rel="self" type="application/xml" href= "https://10.6.60.202/apis/
  kraken/outputs/output-d7948861-42b2-4731-9465-c1325a9b7a4d" />
</response>
```

Retrieve Information for a Specific Output

Retrieves information for a specific output. .
Return data includes an <output> element.

Authorizations: All

Requests

```
GET https://example.haivision.com/apis/kraken/outputs/output-{id}
```

HTTP Return Codes

Code	Description
200	Results found
404	Unknown ID

Response

```
<response>
  <output>
    <uuid>f5684714-5ee6-4523-ab73-c61773769b53</uuid>
    <name>Makito Output 1</name>
    <url>udp://239.1.10.29:4900/</url>
    <description>Makito Output 1</description>
    <state>stopped</state>
    <mtu>1490</mtu>
    <ttl>64</ttl>
    <tos>184</tos>
    <link rel="self" type="application/xml" href="https://example.haivision.com
    /apis/kraken/outputs/output- ef8da11d-af76-437b-bc2a-5b3afced8e3f" />
  </output>
</response>
```

Edit an Output

Edits an `<output>` resource. Like the Create a New Output call, output and url elements are required in the request body, and the response returns a `<link>` element to updated resource.

Authorizations: All

Requests

```
PUT https://example.haivision.com/apis/kraken/outputs/output-{id}
<output>
  <name>New name</name>
  <url>udp://239.202.1.4:4900</url>
  <description>New description</description>
  <mtu>500</mtu>
  <ttl>24</ttl>
  <tos>156</tos>
  <sap>
    <transmitSap>on</transmitSap>
    <address>224.2.127.254</address>
    <port>9875</port>
    <sessionName>A name</sessionName>
    <sessionDescription>A description</sessionDescription>
    <keywords>A keyword</keywords>
    <author>An author</author>
  </sap>
  <interface>https://10.6.60.202/apis/servers/serverbc305be293d3/nics/
 nic-eth1</interface>
</output>
```

Note

Populate the <interface> element with a full link to the network interface receiving the stream. Retrieve the link to use by issuing a Get List of Server NICs call.

HTTP Return Codes

Name	Description
200	Output successfully updated
400	Bad Request
404	Unknown ID
500	Server error

Response

```
<response>
  <link rel="self" type="application/xml" href="https://10.6.60.202/apis/
  kraken/outputs/output-d7948861-42b2-4731-9465-c1325a9b7a4d" />
</response>
```

Remove an Output

Removes an `<output>` resource. The response data includes a `<link>` element pointing to the outputs collection.

Authorizations: All

Requests

```
DELETE https://example.haivision.com/apis/kraken/outputs/output-{id}
```

HTTP Return Codes

Code	Description
200	Output successfully deleted
400	Bad Request
404	Unknown ID
500	Server error

Response

```
<response>
  <link rel="self" type="application/xml" href="https://example.haivision.com/
  apis/kraken/outputs" />
</response>
```

Transcoder Resources

The `transcoders` API allows you to define and get information about audio and video characteristics to change in the outbound stream.

Transcoder Properties and API Commands:

Transcoder Properties

The properties within the transcoder element used throughout this section are defined below. Additional properties are available depending on the type of stream.

Property	Description and Values
uuid	Identifies the transcoder. Generated by Kraken.
name	Name of the transcoder.
resolutionMode	The resolution for the outbound stream: 1: Custom (width/height as defined by resolution) 2: Auto (Detect on stream start) 3: Auto (Detect continuously). Default when omitted.
resolution	(resolutionMode must be Custom) The resolution for the outbound stream. Use a horizontal and vertical (Width x Height) resolution. For example, 1280x720.
frameRate	The coded picture frame rate per second (fps): -1: Auto (Detect Continuously) 0: Auto (Detect on Stream Start) 1–60: Actual fixed framerate Default: 30
videoBitrate	The video bitrate for the outbound stream in kbps, for example, 1024. Range: 150–15000, Default: 3000
gopSize	The GOP (Group of Pictures) Size for the outbound stream, for example, 30. Range: 0–1000, Default: 30
metadata	Enables (on) or disables (off) KLV metadata pass-through.
audio	Enables or disables audio on the outbound stream.

Property	Description and Values
videoType	The video format for the outbound stream, either: • avc: H.264 (Default) • hevc: H.265 • mpeg2video: (License required) The stream is transcoded to MPEG-2 Video and MPEG-1 or MPEG-2 audio with closed captioning pass-through. This allows Kraken to inter-operate with legacy systems.
audioBitrate	The audio bitrate for the outbound stream in kbps. Range: 14–576, Default: 256
bFrame	The number of B-frames and B reference frames per P-Frames to allow in the output stream: Range: -1–3, Default: -1
enableIntraRefresh	Enables (on) or disables (off) Intra Refresh for X.264 and X.265. This is an advanced feature that puts the encoder into a mode where it does not generate I-Frames. Instead the individual macro blocks are refreshed and over time the entire picture is refreshed. This eliminates the I-Frame bitrate spikes and smoots the bitrate over the GOP interval. The GOP parameter is still used as a basis for the refresh interval. Default: off
encodingProfile	Adjusts the desired quality level (balancing speed vs. quality). • vbr: Variable bit rate (AVC videoType only, Default) • cbr: Constrained bitrate • performance: Fastest and lowest quality • balanced: Provides a balance between speed and quality (Default for HEVC videoType) • quality: Slowest and highest quality.
encoderType	(Optional, to enable hardware acceleration on qualified hardware) Select the encoding format, either: • unset: Software CPU-based encoding (Default, Kraken legacy) • qsv: Intel Quick Sync Video (License required). Video encoding is hardware accelerated using the Quick Sync Video capabilities of the processor.
interleaveAdjustment Ms	The number of milliseconds to delay audio before multiplexing (muxing). Setting the value to -1 (Default) leaves it up to the transcoder to decide. Range: -5000–5000ms
jitterBufferMs	The jitter buffer for the inbound source/stream. A jitter may be applied to video streams coming in at irregular intervals to help output the video in a steady stream (Default: 250 ms). **Note** Transcoding latency is affected proportionately.
videoProfile	(videoType must be mpeg2video) The video profile for the encoder: • unset: Defaults to Main profile. • mpeg2_main: The output encoded video adheres to the ISO/IEC 13818-2 / MPEG-2 Main Profile. • mpeg2_simple: The output encoded video adheres to the ISO/IEC 13818-2 / MPEG-2 Simple Profile. **Tip** Leave at unset if videoType is hevc or avc.

Property	Description and Values
audioType	(videoType must be mpeg2 video) The audio compression algorithm: • unset: Defaults to mpeg1_layer2. • mpeg1_layer2: Encodes audio using the ISO/IEC 11172-3 / MPEG-1 Layer II algorithm. • mpeg2_aac_adts: Encodes audio using the ISO/IEC 13818-7 / MPEG-2 AAC-LC algorithm with an ADTS header. **Tip** Leave at unset if videoType is hevc or avc.

Get a List of Transcoders

Gets the list of transcoders. Returns a `<transcoders>` element containing multiple `<transcoder>` elements.

Each transcoder element contains a link for use in the following commands: **Retrieve Information for a Specific Transcoder**, **Edit a Transcoder**, and **Remove a Transcoder**.

Authorizations: All

Requests

```
GET https://example.haivision.com/apis/kraken/transcoders
```

HTTP Return Codes

Name	Description
200	Results found
404	No results found

Response

```
<response>
  <transcoders>
    <transcoder>
      <uuid>f00a2077-a6ff-4335-92a9-6a7fd543e20d</uuid>
      <name>SD</name>
      <link rel="self" type="application/xml"
    href="https://example.haivision.com/apis/kraken/transcoders/
    transcoder-f00a2077-a6ff-4335-92a9-6a7fd543e20d" />
    </transcoder>
  </transcoders>
</response>
```

Create a New Transcoder

Creates a new `<transcoder>` resource. The body of the request must contain an transcoder element, with a required name element. The response data returns a link to the newly created transcoder for use in the following commands: Retrieve Information for a Specific Transcoder, Edit a Transcoder, and Remove a Transcoder.

Authorizations: All

Requests

Without Traffic Shaping | With Traffic Shaping

Without Traffic Shaping

```
POST https://example.haivision.com/apis/kraken/transcoders

<transcoder>
  <name>SD-4</name>
  <resolution>
    <width>720</width>
    <height>480</height>
  </resolution>
  <frameRate>30</frameRate>
  <videoBitrate>3000</videoBitrate>
  <gopSize>30</gopSize>
  <metadata>off</metadata>
  <audio>on</audio>
  <audioBitrate>256</audioBitrate>
  <videoType>avc</videoType>
  <qualityLevel>6</qualityLevel>
  <bFrame>2</bFrame>
  <enableIntraRefresh>on</enableIntraRefresh>
  <encodingProfile>performance</encodingProfile>
</transcoder>
```

Without Traffic Shaping | With Traffic Shaping

With Traffic Shaping

```
POST https://example.haivision.com/apis/kraken/transcoders


<transcoder>
  <name>1000kbps</name>
  <resolutionMode>2</resolutionMode>
  <framerate>-1</framerate>
  <videoBitrate>1000</videoBitrate>
  <audioBitrate>60</audioBitrate>
  <gopSize>3444</gopSize>
  <metadata>on</metadata>
  <audio>on</audio>
  <videoType>hevc</videoType>
  <bFrame>80</bFrame>
  <enableIntraRefresh>on</enableIntraRefresh>
  <encodingProfile>quality</encodingProfile>
  <encoderType>software</encoderType>
  <useLegacyTranscoder>on</useLegacyTranscoder>
  <interleaveAdjustmentMs>1</interleaveAdjustmentMs>
  <jitterBufferMs>5</jitterBufferMs>
  <outputPacing>off</outputPacing>
  <outputPacingInterval>-1</outputPacingInterval>
  <shapingBitrate>100</shapingBitrate>
  <shapingMaxBitratePercent>80</shapingMaxBitratePercent>
  <shapingTargetBitratePercent>70</shapingTargetBitratePercent>
  <shapingVbvSizeMs>2000</shapingVbvSizeMs>
  <shapingBufferingIntervalMs>2260</shapingBufferingIntervalMs>
  <enablePcs>on</enablePcs>
</transcoder>
```

Note

As of Version 1.3, `framerate` is an absolute value, rather than a divisor.

Any fields that are not defined are set to their default value, which for resolution, GOP, and audio and video bitrates is zero.

HTTP Return Codes

Code	Description
201	Created
400	Bad Request
500	Server error

Response

```
<response>
  <link rel="self" type="application/xml" href= "https://example.haivision.com/
 apis/kraken/transcoders/transcoder-f00a2077-a6ff-4335-92a9-6a7fd543e20d" />
</response>
```

Retrieve Information for a Specific Transcoder

Retrieves information for a specific transcoder.
Return data includes a <transcoder> element.

Authorizations: All

Requests

```
GET https://example.haivision.com/apis/kraken/transcoders/transcoder-{id}
```

HTTP Return Codes

Code	Description
200	Results found
404	Unknown ID

Response

Without Traffic Shaping | With Traffic Shaping

Without Traffic Shaping

```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
  <transcoder>
    <uuid>b7fe6ed1-9082-4a22-b848-3673e0b56c23</uuid>
    <name>new from aP8</name>
    <resolutionMode>3</resolutionMode>
    <frameRate>-1</frameRate>
    <videoBitrate>9999</videoBitrate>
    <metadata>off</metadata>
    <audio>on</audio>
    <videoType>mpeg2video</videoType>
    <bFrame>-1</bFrame>
    <enableIntraRefresh>off</enableIntraRefresh>
    <encodingProfile>vbr</encodingProfile>
    <encoderType>unset</encoderType>
    <interleaveAdjustmentMs>-1</interleaveAdjustmentMs>
    <jitterBufferMs>250</jitterBufferMs>
    <outputPacing>on</outputPacing>
    <outputPacingInterval>100</outputPacingInterval>
    <shapingBitrate>-1</shapingBitrate>
    <shapingMaxBitratePercent>-1</shapingMaxBitratePercent>
    <shapingTargetBitratePercent>-1</shapingTargetBitratePercent>
    <shapingVbvSizeMs>-1</shapingVbvSizeMs>
    <shapingBufferingIntervalMs>-1</shapingBufferingIntervalMs>
    <enablePcs>off</enablePcs>
    <videoProfile>mpeg2_main</videoProfile>
    <audioType>mpeg2_aac_adts</audioType>
    <state>unset</state>
    <link rel="self" type="application/xml" href= "https://10.65.11.223/apis/
 kraken/transcoders/transcoder-b7fe6ed1-9082-4a22-b848-3673e0b56c23" />
  </transcoder>
</response>
```

Without Traffic Shaping | With Traffic Shaping

With Traffic Shaping

```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
  <transcoder>
    <uuid>88ca6209-ca59-4226-a72c-2012231bf915</uuid>
    <name>frank</name>
    <resolutionMode />
    <framerate>-1</framerate>
    <gopSize>3444</gopSize>
    <metadata>on</metadata>
    <audio>on</audio>
    <videoType>hevc</videoType>
    <bFrame>80</bFrame>
    <enableIntraRefresh>on</enableIntraRefresh>
    <encodingProfile>1</encodingProfile>
    <encoderType>balanced</encoderType>
    <useLegacyTranscoder>on</useLegacyTranscoder>
    <interleaveAdjustmentMs>1</interleaveAdjustmentMs>
    <jitterBufferMs>5</jitterBufferMs>
    <outputPacing>on</outputPacing>
    <outputPacingInterval></outputPacingInterval>
    <shapingBitrate>100</shapingBitrate>
    <shapingMaxBitratePercent>4441</shapingMaxBitratePercent>
    <shapingTargetBitratePercent>80</shapingTargetBitratePercent>
    <shapingVbvSizeMs>80</shapingVbvSizeMs>
    <shapingBufferingIntervalMs>2260</shapingBufferingIntervalMs>
    <enablePcs>on</enablePcs>
    <state>unset</state>
    <link rel="self" type="application/xml" href= "https://10.66.133.141/apis/
 kraken/transcoders/transcoder-88ca6209-ca59-4226-a72c-2012231bf915" />
  </transcoder>
</response>
```

Edit a Transcoder

Edits a `<transcoder>` resource. Like the Create a New Transcoder call, transcoder and name elements are required in the request body, and the response returns a `<link>` element to updated resource.

Authorizations: All

Requests

```
PUT https://example.haivision.com/apis/kraken/transcoders/transcoder-{id}

<transcoder>
  <name>frank</name>
  <resolution>
    <width>800</width>
    <height>600</height>
  </resolution>
  <frameRate>0</frameRate>
  <videoBitrate>2000</videoBitrate>
  <gopSize>30</gopSize>
  <metadata>off</metadata>
  <audio>on</audio>
  <audioBitrate>128</audioBitrate>
  <videoType>avc</videoType>
  <qualityLevel>6</qualityLevel>
  <bFrame>2</bFrame>
  <enableIntraRefresh>on</enableIntraRefresh>
  <encodingProfile>performance</encodingProfile>
</transcoder>
```

Note

Editing a transcoder overwrites any fields that have not been set with their default value.

HTTP Return Codes

Code	Description
200	Transcoder successfully updated
400	Bad Request
404	Unknown ID
500	Server error

Response

```
<response>
  <link rel="self" type="application/xml" href="https://example.haivision.com/
 apis/kraken/transcoders/transcoder-f00a2077-a6ff-4335-92a9-6a7fd543e20d" />
</response>
```

Remove a Transcoder

Removes a `<transcoder>` resource. The response data includes a `<link>` element pointing to the transcoders collection.

Authorizations: All

Requests

```
DELETE https://example.haivision.com/apis/kraken/transcoders/transcoder-{id}
```

HTTP Return Codes

Code	Description
200	Transcoder successfully deleted
400	Bad request
404	Unknown ID
500	Server error

Response

```
<response>
  <link rel="self" type="application/xml" href="https://example.haivision.com/
  apis/kraken/transcoders"/>
</response>
```

Stream Resources

The `streams` API allows you to select from defined inputs, transcoders, outputs, and optionally, metadata sources to set up real-time stream-based transcoding. You can also get information about the streams in the system.

Stream Properties and API Commands:

Stream Properties

The properties within the stream element used throughout this section are defined below.

Property	Description and Values
uuid	Identifies the stream. Generated by Kraken.
name	Name of the stream.
auto_start	Enables (true) or disables (false) auto-starting this stream when a Preset is loaded via the Administration Preset page or applied after a reboot. A GET request indicates the auto-start status.
state	(GET) Indicates if the stream is active or not. It can return either: active, stopped, waiting, or unset (not part of a stream). (POST/PUT) To start and stop a stream, set the value to start or stop. **Tip** The state cannot be set with other properties and must be set by itself.
description	Description of the stream.
input	The input for the transcoded stream.
passthru	(POST/PUT) Sets the stream to "pass through" to the specified output URL. **Note** To not enable passthru, do not send this parameter or leave its value empty.
mode	(GET) Indicates the stream routing mode: • transcoder: (Default) The input is transcoded to one or more outputs. • passthru (Transcoder + Passthru): The input is transcoded and also re-transmitted to another destination. • bypass: The input is not transcoded, but simply copied (as is) to the outputs.

Property	Description and Values
passthru	(GET) (mode must be passthru) Indicates the name and URL of the output. name: name of the output link: REST API url of the output
transcoder	The uuid of the transcoder to apply to the stream.
outputs	The uuid of the output for the transcoded stream.
metadata	The uuid of the Metadata source for the transcoded stream.

Get a List of Streams

Gets the list of streams. The return data includes a `<streams>` element containing multiple `<stream>` elements. Each stream element contains a link for use in the following commands: **Retrieve Information for a Specific Stream**, **Edit, Start, or Stop a Stream**, and **Remove a Stream**.

Authorizations: All

Requests

```
GET https://example.haivision.com/apis/kraken/streams
```

HTTP Return Codes

Code	Description
200	Results found
404	No results found

Response

```
<response>
  <streams>
  <stream>
    <uuid>ee0d583a-e628-4541-971f-1b5d847fec31</uuid>
    <name>Session 1</name>
    <state>active</state>
    <link rel="self" type="application/xml" href="https://example.haivision.com
 /apis/kraken/streams/stream-ee0d583a-e628-4541-971f-1b5d847fec31" />
  </stream>
  </streams>
</response>
```

Create a New Stream

Creates a new stream. The body of the request must contain a stream element, with a required name element. The response data returns a link to the newly created stream for use in the following commands: Retrieve Information for a Specific Stream, Edit, Start, or Stop a Stream, and Remove a Stream.

Authorizations: All

Requests

```
POST https://example.haivision.com/apis/kraken/streams
<stream>
  <name>Stream 124</name>
  <auto_start>true</auto_start>
  <description>Stream Description</description>
  <input>https://10.1.40.89/apis/kraken/inputs/
 input-c769e651-b6a5-4547-94d5-f06bd2877a65</input>
  <transcoder>https://10.1.40.89/apis/kraken/transcoders/
 transcoder-9ff39a54-309c-4834-aa33-2b59549d0a1d</transcoder>
  <outputs>
    <output>https://10.1.40.89/apis/kraken/outputs/
 output-a87b141c-a4d2-4b04-bb20-d709b0b26225</output>
    <output>https://10.1.40.89/apis/kraken/outputs/
 output-ded3a62b-1a57-4352-a9bf-a336a97f4e15</output>
  </outputs>
  <metadatas>
    <metadata>https://10.65.11.209/apis/kraken/metadatas/
 metadata-e8b5a07d-4ca7-4651-8ae9-035e0849d235</metadata>
  </metadatas>
</stream>
```

Note

If state is not supplied, the stream is stopped. If state is start or auto-start is true, the stream is started.

HTTP Return Codes

Code	Description
201	Created
400	Bad Request
500	Server error

Response

```
<response>
  <link rel="self" type="application/xml" href="https://example.haivision.com/
 apis/kraken/streams/stream-ee0d583a-e628-4541-971f-1b5d847fec31" />
</response>
```

Retrieve Information for a Specific Stream

Retrieves information for a specific stream. Return data includes a <stream> element.

Authorizations: All

Requests

```
GET https://example.haivision.com/apis/kraken/streams/stream-{id}
```

HTTP Return Codes

Code	Description
200	Results found
404	Unknown ID

Response

Stream without Passthru | Stream with Passthru

Stream without Passthru

```
<response>
  <stream>
      <uuid>d4701f54-ac37-4768-8e8f-a1000ab59f05</uuid>
      <name>Stream1</name>
      <auto_start>true</auto_start>
      <state>stopped</state>
      <description>4mbps CBR</description>
      <input>
        <name>Mako720</name>
        <link rel="input" type="application/xml" href="https://10.1.40.89/
 apis/kraken/inputs/input-c769e651-b6a5-4547-94d5-f06bd2877a65" />
      </input>
      <transcoder>
        <name>Transcoder1 4mbps CBR</name>
        <link rel="transcoder" type="application/xml" href="https://10.1.40.89/
 apis/kraken/transcoders/transcoder-9ff39a54-309c-4834-aa33-2b59549d0a1d" />
      </transcoder>
      <outputs>
      <output>
        <name>Output1</name>
        <link rel="output" type="application/xml" href="https://10.1.40.89/
 apis/kraken/outputs/output-a87b141c-a4d2-4b04-bb20-d709b0b26225" />
      </output>
      <output>
        <name>Output2</name>
        <link rel="output" type="application/xml" href="https://10.1.40.89/
 apis/kraken/outputs/output-ded3a62b-1a57-4352-a9bf-a336a97f4e15" />
      </output>
    </outputs>
    <metadatas>
      <metadata>
        <name>meta1</name>
        <link rel="metadata" type="application/xml" href="https://10.65.11.209/
 apis/kraken/metadatas/metadata-e8b5a07d-4ca7-4651-8ae9-035e0849d235" />
      </metadata>
    </metadatas>
    <link rel="self" type="application/xml" href= "https://10.1.40.89/
 apis/kraken/streams/stream-d4701f54-ac37-4768-8e8f-a1000ab59f05" />
  </stream>
</response>
```

Stream without Passthru Stream with Passthru

Stream with Passthru

```
<response>
  <stream>
    <uuid>9276ea3f-ac21-4160-8760-8f2bb63d1e5d</uuid>
    <name>Stream with Passthru</name>
    <mode>passthru</mode>
    <auto_start>true</auto_start>
    <state>stopped</state>
    <input>
      <name>from multicast</name>
      <link rel="input" type="application/xml" href="https://10.65.11.151/
 apis/kraken/inputs/input-ee8ff55d-161d-4d48-a2f5-1550610cd141" />
    </input>
    <passthru>
      <name>to PC monitoring</name>
      <link rel="output" type="application/xml" href="https://10.65.11.151/
 apis/kraken/outputs/output-d730641f-6367-4d48-be65-4f365b086379" />
    </passthru>
    <transcoder>
      <name>Transcoder1</name>
      <link rel="transcoder" type="application/xml" href="https://10.65.11.151/
 apis/kraken/transcoders/transcoder-e7b75426-65ee-4363-be70-e57b5569b61d" />
    </transcoder>
    <outputs>
      <output>
        <name>alternate</name>
        <link rel="output" type="application/xml" href="https://10.65.11.151/
 apis/kraken/outputs/output-a87b141c-a4d2-4b04-bb20-d709b0b26225" />
      </output>
      <output>
        <name>Output2</name>
        <link rel="output" type="application/xml" href="https://10.65.11.151/
 apis/kraken/outputs/output-ded3a62b-1a57-4352-a9bf-a336a97f4e15" />
      </output>
    </outputs>
    <link rel="self" type="application/xml" href="https://10.65.11.151/
 apis/kraken/streams/stream-d4701f54-ac37-4768-8e8f-a1000ab59f05" />
  </stream>
</response>
```

Edit, Start, or Stop a Stream

Edits a `<stream>` resource. Also, starts or stops a stream. Like the Create a New Stream call, stream and name elements are required in the request body, and the response returns a `<link>` element to updated resource.

Authorizations: All

Requests

Edit a Stream | Start a Stream | Stop a Stream | Stream with Passthru

Edit a Stream

```
PUT https://example.haivision.com/apis/kraken/streams/stream-{id}

<stream>
  <name>Stream 124</name>
  <description>Stream Description</description>
  <input>https://10.1.40.89/apis/kraken/inputs/
 input-c769e651-b6a5-4547-94d5-f06bd2877a65</input>
  <transcoder>https://10.1.40.89/apis/kraken/transcoders/
 transcoder-9ff39a54-309c-4834-aa33-2b59549d0a1d</transcoder>
  <outputs>
    <output>https://10.1.40.89/apis/kraken/outputs/
 output-a87b141c-a4d2-4b04-bb20-d709b0b26225</output>
    <output>https://10.1.40.89/apis/kraken/outputs/
 output-ded3a62b-1a57-4352-a9bf-a336a97f4e15</output>
  </outputs>
</stream>
```

Edit a Stream | Start a Stream | Stop a Stream | Stream with Passthru

Starts a Stream

```
PUT https://example.haivision.com/apis/kraken/streams/stream-{id}

<stream>
  <state>start</state>
</stream>
```

Note

When using the <state> element, separate other non-state actions from the state action. Other changes in the same PUT request are ignored.

Edit a Stream | Start a Stream | Stop a Stream | Stream with Passthru

Stop a Stream

```
PUT https://example.haivision.com/apis/kraken/streams/stream-{id}

<stream>
  <state>stop</state>
</stream>
```

Note

When using the <state> element, separate other non-state actions from the state action. Other changes in the same PUT request are ignored.

Edit a Stream | Start a Stream | Stop a Stream | Stream with Passthru

Stream with Passthru

```
PUT https://example.haivision.com/apis/kraken/streams/stream-{id}

<stream>
  <name>Stream with Passthru</name>
  <mode>passthru</mode>
  <auto_start>true</auto_start>
  <input>https://10.65.11.151/apis/kraken/inputs/
 input-ee8ff55d-161d-4d48-a2f5-1550610cd141</input>
  <passthru>https://10.65.11.151/apis/kraken/outputs/
 output-d730641f-6367-4d48-be65-4f365b086379</passthru>
  <transcoder>https://10.65.11.151/apis/kraken/transcoders/
    transcoder-e7b75426-65ee-4363-be70-e57b5569b61d</transcoder>
  <outputs>
    <output>https://10.65.11.151/apis/kraken/outputs/
    output-bfa36a21-3ceb-466d-b041-e00b18b9f517</output>
    <output>https://10.65.11.151/apis/kraken/outputs/
    output-d84bef04-9290-496e-8c3b-fa62be09ed43</output>
  </outputs>
</stream>
```

HTTP Return Codes

Name	Description
200	Stream successfully updated
400	Bad request
404	Unknown ID
500	Server error

Response

```
<response>
  <link rel="self" type="application/xml" href= "https://example.haivision.com/
 apis/kraken/streams/stream-ee0d583a-e628-4541-971f-1b5d847fec31" />
</response>
```

Remove a Stream

Removes a `<stream>` resource. The response data includes a `<link>` element pointing to the streams collection.

Authorizations: All

Requests

```
DELETE https://example.haivision.com/apis/kraken/streams/stream-{id}
```

HTTP Return Codes

Code	Description
200	Stream successfully deleted
400	Bad request
404	Unknown ID
500	Server error

Response

```
<response>
  <link rel="self" type="application/xml" href= "https://example.haivision.com/
  apis/kraken/streams"/>
</response>
```

Configuration Resources

The `configurations` API allows you to save the current configuration, load saved configurations, set the default configuration to load on startup, or delete a saved configuration. You can also get information about the configurations in the system.

Configuration Properties and API Commands:

Configuration Resource Properties

The properties within the configuration element used throughout this section are defined below.

Property	Description and Values
uuid	Identifies the stream. Generated by Kraken.
name	Name of the stream.
defaultConfig	Indicates if the stream is the default.

Get a List of Configurations

Gets the list of configurations. The return data includes a `<configurations>` element containing multiple `<configuration>` elements. Each output element contains a link for use in the following commands: Retrieve Information for a Specific Configuration, Edit a Configuration, and Remove a Configuration.

Authorizations: All

Requests

```
GET https://example.haivision.com/apis/kraken/configurations
```

HTTP Return Codes

Name	Description
200	Results found
404	No results found

Response

```
<response>
  <configurations>
  <configuration>
    <uuid>83d0df6f-b1f9-421e-8942-06054703d265</uuid>
    <name>ConfigA</name>
    <defaultConfig>yes</defaultConfig>
    <link rel="self" type="application/xml"
     href="https://example.haivision.com/apis/kraken/configurations/
     configuration-83d0df6f-b1f9-421e-8942-06054703d265"/>
  </configuration>
  <configuration>
    <uuid>da8b2745-ba57-4b0d-bee9-be527d310961</uuid>
    <name>ConfigB</name>
    <defaultConfig>no</defaultConfig>
    <link rel="self" type="application/xml"
     href="https://example.haivision.com/apis/kraken/configurations/
     configuration-da8b2745-ba57-4b0d-bee9-be527d310961"/>
  </configuration>
  </configurations>
</response>
```

Create a New Resource

Authorizations: All

Creates a new resource (i.e., saves the current configuration). The body of the request must contain an configuration element, with a required name element. The response data returns a link to the newly created input for use in the following commands: Retrieve Information for a Specific Configuration, Edit a Configuration, and Remove a Configuration.

Requests

```
POST https://example.haivision.com/apis/kraken/configurations
<configuration>
  <name>ConfigA</name>
</configuration>
```

HTTP Return Codes

Name	Description
201	Created
400	Bad request
500	Server error

Response

```
<response>
  <link rel="self" type="application/xml"
   href="https://example.haivision.com/apis/kraken/configurations/
   configuration-83d0df6f-b1f9-421e-8942-06054703d265"/>
</response>
```

Retrieve Information for a Specific Configuration

Authorizations: All

Retrieves Information for a specific configuration. Return data includes an <configuration> element.

Requests

```
GET https://example.haivision.com/apis/kraken/configurations/configuration-{id}
```

HTTP Return Codes

Code	Description
200	Results found
404	Unknown ID

Response

```
<response>
  <configuration>
    <uuid>83d0df6f-b1f9-421e-8942-06054703d265</uuid>
    <name>ConfigA</name>
    <defaultConfig>yes</defaultConfig>
    <link rel="self" type="application/xml"
     href="https://example.haivision.com/apis/kraken/configurations/
     configuration-83d0df6f-b1f9-421e-8942-06054703d265"/>
  </configuration>
</response>
```

Edit a Configuration

Edits a `<configuration>` resource. The request data must include a `<configuration>` element, with one or both of the following required:

- `<activate>` to load the configuration, and/or
- `<default>` to make the configuration load upon bootup.

The response returns a `<link>` element to updated resource.

Authorizations: All

Requests

Load Configuration | Make Configuration the Default | Load Configuration and Make Default

To load the configuration:

```
PUT https://example.haivision.com/apis/kraken/configurations/configuration-{id}

<configuration>
  <activate>1</activate>
</configuration>
```

Load Configuration | Make Configuration the Default | Load Configuration and Make Default

To make the configuration the default upon bootup:

```
PUT https://example.haivision.com/apis/kraken/configurations/configuration-{id}

<configuration>
  <default>1</default>
</configuration>
```

Load Configuration | Make Configuration the Default | Load Configuration and Make Default

To load the configuration and make it the default upon bootup:

```
PUT https://example.haivision.com/apis/kraken/configurations/configuration-{id}

<configuration>
  <activate>1</activate>
  <default>1</default>
</configuration>
```

Note

The `<activate>` or `<default>` status is either 1 (yes) or 0 (no).

HTTP Return Codes

Name	Description
200	Configuration successfully updated
400	Bad request
404	Unknown ID
500	Server error

Response

```
<response>
  <link rel="self" type="application/xml"
   href="https://example.haivision.com/apis/kraken/configurations/
   configuration-83d0df6f-b1f9-421e-8942-06054703d265"/>
</response>
```

Remove a Configuration

Removes a `<configuration>` resource. The response data includes a `<link>` element pointing to the configurations collection.

Authorizations: All

Requests

```
DELETE https://example.haivision.com/apis/kraken/configurations/config-{id}
```

HTTP Return Codes

Code	Description
200	Configuration successfully deleted
400	Bad request
404	Unknown ID
500	Server error

Response

```
<response>
  <link rel="self" type="application/xml" href="https://example.haivision.com/
  apis/kraken/configurations"/>
</response>
```

System Resources

The `system` API allows you to get the Kraken software version number and capture card information. (GET only).

System Properties and API Commands:

- System Properties
- Get System Information

System Properties

The properties within the system element used throughout this section are defined below.

Property	Description
version	Identifies the Kraken version.
captureInfo	Displays current capture information, including: `keep_alive_counter`, `port_info_size`, `start_time`, `running`
capturePort	Displays capture port information and available modes for Analog Composite inputs, including: `port_number`, `config_mode`, `config_format`, `config_framerate`, `current_video_desc`, `available_modes`
decklinkInfo	Displays current capture information, including: `keep_alive_counter`, `port_info_size`, `start_time`, `running`
decklinkPort	Displays capture port information and available modes for DeckLink inputs, including: `port_number`, `config_mode`, `config_format`, `config_framerate`, `current_video_desc`, `available_modes`

Get System Information

Gets the Kraken version number, as well as current capture information, port information, and available modes for DeckLink or capture cards. Returns a system element.

Authorizations: All

Requests

```
GET https://example.haivision.com/apis/kraken/system
```

HTTP Return Codes

Name	Description
200	Results found
404	No results found

Response

```
<response>
  <system>
    <version>2.5.0</version>
    <captureInfo>
      <keep_alive_counter>34542</keep_alive_counter>
      <port_info_size>1</port_info_size>
      <start_time>2017-Jul-21 15:21:19.371990</start_time>
      <running>1</running>
    </captureInfo>
    <capturePort>
      <port_number>0</port_number>
      <config_mode>-2</config_mode>
      <config_format>unset</config_format>
      <config_framerate>unset</config_framerate>
      <current_video_desc>
        <width>720</width>
        <height>480</height>
        <mode_id />
      </current_video_desc>
      <available_modes>
        <mode>
          <width>-1</width>
          <height>-1</height>
          <mode_id>NTSC-M-JP</mode_id>
        </mode>
      ...
      </available_modes>
    </capturePort>

    <decklinkInfo>
      <keep_alive_counter>34527</keep_alive_counter>
      <port_info_size>1</port_info_size>
      <start_time>2017-Jul-21 15:23:15.732919</start_time>
      <running>1</running>
    </decklinkInfo>
    <decklinkPort>
      <port_number>0</port_number>
      <config_mode>-1</config_mode>
      <config_format />
      <config_framerate />
      <current_video_desc>
        <width>1920</width>
        <height>1080</height>
        <framerate>29.97</framerate>
        <interlaced>true</interlaced>
      </current_video_desc>
      <available_modes>
        <mode>
          <width>720</width>
          <height>576</height>
          <framerate>25.00</framerate>
          <interlaced>true</interlaced>
        </mode>
        <mode>
          <width>720</width>
          <height>486</height>
          <framerate>29.97</framerate>
          <interlaced>true</interlaced>
        </mode>
      ...
      </available_modes>
    </decklinkPort>
  </system>
</response>
```

Server Resources

The `servers` API allows you to get information for the Kraken physical server and Network Interface Cards (NICs).

Server Properties and API Commands:

- Server Resource Properties
- Get List of Servers
- Get Specific Server Information
- Get List of Server NICs
- Get Specific NIC Information

Server Resource Properties

Property	Description and Values
id	Identifies the Kraken physical server or NIC

Get List of Servers

Returns the server ID. The return data includes a `<servers>` element containing multiple `<server>` elements. Each server element contains a link for use in the following commands: Get Specific Server Information, Get List of Server NICs, and Get Specific NIC Information.

Authorizations: All

Requests

```
GET https://example.haivision.com/apis/servers
```

Response

```
<response>
  <servers>
    <server>
      <id>bc305be293d3</id>
      <link rel="self" type="application/xml" href="https://10.6.60.202/apis/
      servers/server-bc305be293d3" />
    </server>
  </servers>
</response>
```

Get Specific Server Information

Get specific server information. Response data includes link element to Get List of Server NICs.

Authorizations: All

Requests

```
GET https://example.haivision.com/apis/servers/server-{id}
```

Response

```
<response>
  <server>
    <id>bc305be293d3</id>
    <link rel="self" type="application/xml" href="https://10.6.60.202/apis/
      servers/server-bc305be293d3" />
    <link rel="nics" type="application/xml" href="https://10.6.60.202/apis/
      servers/server-bc305be293d3/nics" />
  </server>
</response>
```

Get List of Server NICs

Get list of server network interfaces. The return data includes a `<nics>` element containing multiple `<nic>` elements. Also includes a links to Get Specific NIC Information.

Authorizations: All

Requests

```
GET https://example.haivision.com/apis/servers/server-{id}/nics
```

Response

```
<response>
  <nics>
    <nic>
      <id>lo</id>
      <name>lo</name>
      <link rel="self" type="application/xml" href= "https://10.6.60.202/apis/
      servers/serverbc305be293d3/nics/nic-lo" />
    </nic>
    <nic>
      <id>eth0</id>
      <name>eth0</name>
      <link rel="self" type="application/xml" href= "https://10.6.60.202/apis/
      servers/serverbc305be293d3/nics/nic-eth0" />
    </nic>
    <nic>
      <id>eth1</id>
      <name>eth1</name>
      <link rel="self" type="application/xml" href= "https://10.6.60.202/apis/
      servers/serverbc305be293d3/nics/nic-eth1" />
    </nic>
  </nics>
</response>
```

Get Specific NIC Information

Get information about a specific network inferface. Response includes the NIC ID and name.

Authorizations: All

Requests

```
GET https://example.haivision.com/apis/servers/server-{id}/nics/nic-{id}
```

Response

```
<response>
  <nic>
    <id>eth0</id>
    <name>eth0</name>
    <link rel="self" type="application/xml" href="https://10.6.60.202/apis/
    servers/server-bc305be293d3/nics/niceth0" />
  </nic>
</response>
```

Error Codes

An error condition will return a specified HTTP status code on the requested action.

If the error is processed internally by the Kraken, the returned data may also contain an `<error>` entity (see XML Entities).

XML <error> Code	HTTP Status Code	Error Message	Common Cause
1000	500 Bad Request	Failed to create new resource	Submitted data was not sufficient, missing required data.
1001	404 Not Found	No results found	Request completed successfully, but no results were found.
1002	404 Not Found	Unknown id	A specific resource was requested but not found (deleted or bad ID specified).
1003	500 Internal Server Error	Error executing SQL query	An internal function failed while executing the request.
1004	500 Bad Request	Failed to update resource	Tried to update a nonexistent entity (bad ID)
1005	500 Bad Request	Failed to delete resource	Tried to delete a nonexistent entity (bad ID)
1006	501 Not Implemented	Unknown API function requested	Tried to access a nonexistent / inactive API location
1007	400 Bad Request	Unknown HTTP method	Tried to use a non-standard HTTP method (other than GET, POST, PUT, or DELETE)
1008	400 Bad Request	Unrecognized URI structure	
1009	501 Not Implemented	HTTP method not implemented	Requested an action not utilized by the target API (e.g., POST on a query-only API)
1010	501 Not Implemented	Function not implemented	
1011	400 Bad Request	Input XML data is poorly formatted	XML content submitted had syntax or validation problems.
1012	500 Internal Server Error	Error while executing	
1013	400 Bad Request	Unrecognized arguments	

XML <error> Code	HTTP Status Code	Error Message	Common Cause
1014	401 Not Authorized	Not Authorized	Authentication credentials are invalid, expired, or removed, or Accessing API site via HTTP protocol instead of HTTPS.
1015	403 Forbidden	API functions not enabled	API access is disabled in server configuration.
1016	503 Service Unavailable	Service provider for this API is unavailable	A server process that supports this API call was unable to be reached. It may be down, inaccessible, or overloaded.

Example Implementation

Following is a working PHP example that uses OAuth authentication to retrieve an input list.

```php
<?php
//OAuth library obtained from
// http://oauth.googlecode.com/svn/code/php/OAuth.php
require_once('OAuth.php');

// Establish an OAuth consumer based on our admin 'credentials'
$CONSUMER_KEY = 'U5k4J0/397WeUaRNWTs+CA';// See footnote 1
$CONSUMER_SECRET = 'iWU2RE2GYn5G8Fcj28+zIw';// See footnote 2
$consumer = new OAuthConsumer($CONSUMER_KEY, $CONSUMER_SECRET, NULL);

// Setup OAuth request based our previous credentials and query
$base_feed = 'https://example.haivision.com/apis/kraken/inputs';// See footnote 3
$params = array();
$request = OAuthRequest::from_consumer_and_token($consumer, NULL, 'GET', $base_feed, $params);

// Sign the constructed OAuth request using HMAC-SHA1
$request->sign_request(new OAuthSignatureMethod_HMAC_SHA1(), $consumer, NULL);

// Make signed OAuth request to the Contacts API server
$url = $base_feed;
echo send_request($request->get_normalized_http_method(), $url, $request->to_header());

/**
*  Makes an HTTP request to the specified URL
*  @param string $http_method The HTTP method (GET, POST, PUT, DELETE)
*  @param string $url Full URL of the resource to access
*  @param string $auth_header (optional) Authorization header
*  @param string $postData (optional) POST/PUT request body
*  @return string Response body from the server  */
function send_request($http_method, $url, $auth_header=null, $postData=null) {
  $curl = curl_init($url);
  curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
  curl_setopt($curl, CURLOPT_FAILONERROR, false);
  curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
  curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
  switch($http_method) {
    case 'GET':
     if ($auth_header) {
        curl_setopt($curl, CURLOPT_HTTPHEADER, array($auth_header));
     }
     break;
    case 'POST':
     curl_setopt($curl, CURLOPT_HTTPHEADER, array('Content-Type: application/atom+xml', $auth_header));
     curl_setopt($curl, CURLOPT_POST, 1);
     curl_setopt($curl, CURLOPT_POSTFIELDS, $postData);
     break;
    case 'PUT':
     curl_setopt($curl, CURLOPT_HTTPHEADER, array('Content-Type: application/atom+xml', $auth_header));
     curl_setopt($curl, CURLOPT_CUSTOMREQUEST, $http_method);
     curl_setopt($curl, CURLOPT_POSTFIELDS, $postData);
     break;
    case 'DELETE':
     curl_setopt($curl, CURLOPT_HTTPHEADER, array($auth_header));
```

```
      curl_setopt($curl, CURLOPT_CUSTOMREQUEST, $http_method);
      break;
  }
  $response = curl_exec($curl);
  if (!$response)
    $response = curl_error($curl);
  }
  curl_close($curl);
  return $response;
}

/**
*  Joins key:value pairs by inner_glue and each pair together by outer_glue
*  @param string $inner_glue The HTTP method (GET, POST, PUT, DELETE)
*  @param string $outer_glue Full URL of the resource to access
*  @param array $array Associative array of query parameters
*  @return string Urlencoded string of query parameters */
function implode_assoc($inner_glue, $outer_glue, $array) {
  $output = array();
  foreach($array as $key => $item) {
    $output[] = $key . $inner_glue . urlencode($item);
  }
  return implode($outer_glue, $output);
}
?>
```

1. Key generated from Web interface REST API page. See **Preparing for OAuth**.
2. Secret generated from Web interface REST API page. See **Preparing for OAuth**.
3. For the list of available URLs, see **API Reference**.

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, Havision 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	• 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	• 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	• 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	10:00pm - 5:00am	48 Hours
Urgent	• 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	• 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

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

info@haivision.com
www.haivision.com

North America: 1.877.224.5445
International: +1.514.334.5445

HAIVISION