# HAIVISION

Haivision Media Platform 3.1
Command Line API Integrator's Guide

HVS-ID-INT-HMP-CLAPI-31

# 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

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

## Disclaimer

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

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

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

# Contents

# 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 how to configure and manage the Haivision Media Platform (HMP) Command Line API to remotely control core HMP functionality such as the creation and management of sessions and recordings. The Command Line API is an alternative to the REST API.

# Command Line API Introduction

This content provides an introduction to the Command Line API for Haivision Media Platform (HMP). The HMP Command Line API provides streamlined remote control of core HMP functionality such as the creation and management of sessions and recordings. It is an alternative to the HMP REST API.

For information on the REST API, see the REST API Integrator's Guide.

**Topics Include**

- Why Use the Command Line API?
- HMP Concepts
  - HMP Objects
  - HMP Metadata
  - HMP Permissions
- Command Line API Syntax
  - Command Syntax
  - Response Syntax
  - Syntax Conventions
- Enabling Command Line API Access

## Why Use the Command Line API?

The Command Line API is designed to enable control system programmers to build custom workflows for HMP. Instead of using a JSON-based REST API, the Command Line API allows programmers to communicate with HMP using plain-text commands sent via a TCP/IP connection.

Using the Command Line API, programmers can perform tasks/actions such as the following:

- Create/Delete a new session
- Add/Remove a source to/from a session
- Create multi-source sessions
- Start/Pause/Resume/Stop a recording
- Get the status of a recording
- Share a session or recording with an HMP user or group
- Add metadata to a session or recording
- Set the category of a session or recording
- Add HotMarks on a recording
- Play a recording as a MPEG-TS stream on a specific multicast IP address and port number

You can communicate with the Command Line API by connecting to HMP's TCP/IP port 23233. During development, the Telnet utility may be used to connect to HMP on this port, allowing you to execute commands interactively.

⚠️ **Important**

Before using the Command Line API, access must be enabled in the HMP web interface. You cannot do so from the Command Line API. See **Enabling Command Line API Access** for details.

# HMP Concepts

Topics in this section:

- **HMP Objects**
- **HMP Metadata**
- **HMP Permissions**

## HMP Objects

The HMP Command Line API can create and manipulate a variety of objects that reside on the HMP system. This section provides a brief description of their functions.

### Identifiers

Each unique HMP object is assigned a randomly generated identifier (ID) when it is created. Commands use these IDs to refer to the specific objects they create or affect.

### Source

A *source* is an available (incoming) HMP-compatible MPEG-TS video stream that HMP content creators can use to generate a session.

### Session

A *session* is a collection of one to four video sources that are identified to provide live viewing, recording, and metadata storage. See **Sessions API**.

### Recording

A *recording* is a collection of one to four stored videos for playback and metadata storage.

| Recording Type | Description |
|---|---|
| Active Recording | An active recording is a recording that is currently being created by recording a session's sources. When it is paused, it becomes a paused recording. When it is stopped it becomes a finished recording. |
| Paused Recording | A paused recording is an active recording that has been suspended. When it is resumed, it becomes an active recording. When it is stopped, it becomes a finished recording. |
| Finished Recording | A finished recording is a recording that has been finalized and is available for playback in its entirety. It is no longer considered active and it may not be paused or resumed. |

See **Recordings API**.

### Stream

A *stream* is an HMP-compatible MPEG-TS video stream that is generated by HMP as it restreams a finished recording.

# HMP Metadata

HMP uses metadata to provide meaningful information about sessions and recordings. This section provides an overview of the kinds of predefined and custom metadata that HMP supports.

## Common Metadata

HMP stores common metadata for sessions and recordings. This type of metadata is defined on all HMP systems (as opposed to custom metadata which is defined by an administrator). The following types of common metadata are used by HMP.

| Metadata Type | Description |
|---|---|
| title | Name given to a session or recording. |
| description | Free-form text account of a session or recording. |
| createdby | Username of the person or entity responsible for making a session or recording.<br><br>⚠️ **Note**<br>When a session or recording is created from the HMP Web Interface, `createdby` is automatically set to the HMP user's display name. However, when a session or recording is created from the Command Line API, `createdby` is not set automatically. See **Update Recording Metadata** and **Update Session Metadata**. |

HMP's common metadata corresponds with the Dublin Core Metadata Element Set:

| HMP Common Metadata | Dublin Core Element |
|---|---|
| title | title |
| description | description |
| createdby | creator |

> ⚠️ **Note**
>
> For more information about the Dublin Core Metadata Element Set refer to *IETF RFC 5013*.

## Custom Metadata

HMP administrators may define keys and values to identify and store custom metadata for sessions and recordings. From the Command Line API, programmers can assign these keys and values to sessions and recordings. See **Set Recording Category** and **Set Session Category**. Viewers can then select key/value pairs to filter the displayed list of sessions or recordings.

## HotMarks

HMP stores HotMarks that indicate points of interest in a recording. Each HotMark is a metadata bookmark that allows viewers to jump to that time in the recording. From the Command Line API, programmers can add HotMarks to annotate an active recording. See **Add a HotMark**.

# HMP Permissions

HMP uses a combination of share permissions and roles with permissions to provide users or groups with controlled access to sessions, videos, and sources. If you do not have permission to at least view a session, video or source, it does not appear in the portal. From the Command Line API, programmers can assign these permissions to users or groups.

User can customize share permissions by granting and revoking specific actions on any available share permission. HMP comes with the default pre-configured share permissions listed below.

For details on share permissions and how to customize them, see "Managing Roles" in the *Haivision Media Platform Administrator's Guide*.

**Session Default Share Permissions**

| Permission Type | Description |
|---|---|
| OWN | The default OWN permission grants the rights to watch, edit, share, record, change sources, and delete a session. |
| EDIT | The default EDIT permission grants the rights to watch and edit a session. |
| WATCH | The default WATCH permission grants the right to watch a session. |

See **Share a Session with a User** and **Share a Session with a Group**.

**Recording Permissions**

| Permission Type | Description |
|---|---|
| OWN | The default OWN permission grants the rights to watch, edit, share, re-stream, trim, download, and delete a video. <br> The OWN recording permission grants a user/group the EDIT recording permission with the additional abilities to give permissions on the recording to other users/groups, to control the recording's expiry behavior, and to delete the session. |
| EDIT | The EDIT recording permission grants a user/group the VIEW recording permission with the additional abilities to edit the recording's metadata, as well as create, edit, and delete HotMarks on Active, Paused, and Finished recordings. |
| WATCH | The WATCH recording permission grants a user/group the abilities to view a recording's metadata and to play back its stored video. |

See **Share a Recording with a User** and **Share a Recording with a Group**.

# Command Line API Syntax

The HMP Command Line API provides a typical line-delimited command/response interface, as shown in the following examples:

| Client Command | Server Response |
|---|---|
| `startRecording e3af4dd4-b358-431e-9450-e7c82f64b6df` | `OK|9eed71ca-d567-4a16-9d9c-26c6c5ad282e` |
| `shareRecording someuser OWN 9eed71ca-d567-4a16-9d9c-26c6c5ad282e` | `OK` |
| `stopRecording 9eed71ca-d567-4a16-9d9c-26c6c5ad282e` | `OK` |
| `undefinedCommand "Example of \"Error\""` | `ERROR|Command not found` |

Command execution is synchronous:

1. HMP reads commands from the TCP/IP connection one at a time.
2. When a command is received, it is executed immediately.
3. After each command has executed, a response is sent to the client device.

## Character Set and Encoding

HMP supports the Unicode character set using the UTF-8 character encoding. All commands sent to the Command Line API must be valid UTF-8.

> ⚠ **Note**
>
> UTF-8 is a compatible superset of the basic US-ASCII character set. For more information on Unicode visit the **Unicode Consortium.** For more information on UTF-8, refer to *IETF RFC 3629*.

## End-Of-Line Character

The line feed control character (code U+000A) delimits the end of commands and responses.

> ⚠ **Note**
>
> Many programming languages use the escape sequence \n to represent a newline control character in a string. The character codes generated for this newline may differ depending on the programming language and operating system in use.

## Command Syntax

HMP follows the standard command-line conventions of a *command* followed by positional *arguments* delimited by spaces. When the command is executed, HMP assigns the values of these arguments to the corresponding *parameters* of the command's definition.

Arguments that contain whitespace (e.g., spaces or tabs) must be represented with quoted strings. Arguments that contain certain special characters must be represented with quoted strings and the special characters must be encoded using escape sequences.

## Quoted Strings

HMP uses spaces (code U+0020) to separate the name of command and its list of arguments. Arguments that contain spaces must be represented with quoted strings. A quote character (code U+0022) indicates the beginning or end of a quoted string. For example:

```
updateMetadata UUID title Jan01-Room101
updateMetadata UUID title "January 1st - Room 101"
```

## Escape Sequences

Within a quoted string, the backslash character (code U+005C) is used to escape certain special characters. Outside of a quoted string, the backslash character is a normal character. A quote character (code U+0022) indicates the beginning or end of a quoted string.

| Escape Sequence | Name | Code |
|---|---|---|
| \n | Line feed | U+000A |
| \" | Quote | U+0020 |
| \\ | Backslash | U+005C |

# Response Syntax

## Success

When a command that does not return a result value succeeds, HMP responds with `OK` .

## Success with Result

When a command that returns a result value succeeds, HMP responds with OK followed by a | character (code U+007C) followed by the response value.

## Error

When a command fails for any reason, HMP responds with:

- ERROR followed by a | character (code U+007C)
- followed by a six-digit error code followed by another | character
- followed by an error message.

The response has the following format: `"ERROR|Code|Error message"` .

For a general list of error messages, see **General Errors**. Also, refer to command-specific errors listed under each command in **Command Line API Reference**.

## Syntax Conventions

The following syntax conventions are used in this document:

| Convention | Description |
|---|---|
| `monospace` | Command names and options, filenames, and code samples appear in monospace typeface. |
| *italics* | Italic font indicates variables or placeholders that you replace with a user-defined value or name. |
| `[ ]` | Square brackets indicate optional items or parameters. Do not type the brackets. |
| `x | y` | Vertical bars are used to separate items in a list of options from which you must select one. |
| … | An ellipsis (…) indicates that the previous option can be repeated multiple times with different values. It can be used inside or outside of brackets. |

# Enabling Command Line API Access

Access to the Command Line API is based on IP address. By default, the Command Line API is disabled, connections to TCP/IP port 23233 are refused, and no IP address is authorized to access the Command Line API. Any unauthorized attempt to connect to the API results in a "Not Authorized" error response and the connection is closed. The list of IP addresses permitted to access the API is managed from the HMP web interface.

Before you can get started, you must activate the Command Line API on HMP and add the client devices' IP addresses to the list of authorized devices. These changes are applied automatically and take effect within 10 seconds. There is no need to restart any services.

> ⚠ **Important**
>
> No further authentication is required for the Command Line API.

To activate Command Line API access on HMP:

1. Click the ☰ icon and click **Administration**.
2. Click **System Settings** on the toolbar and then click **API Access** on the sidebar.

The API Access pane opens, displaying the list of authorized devices (if any).
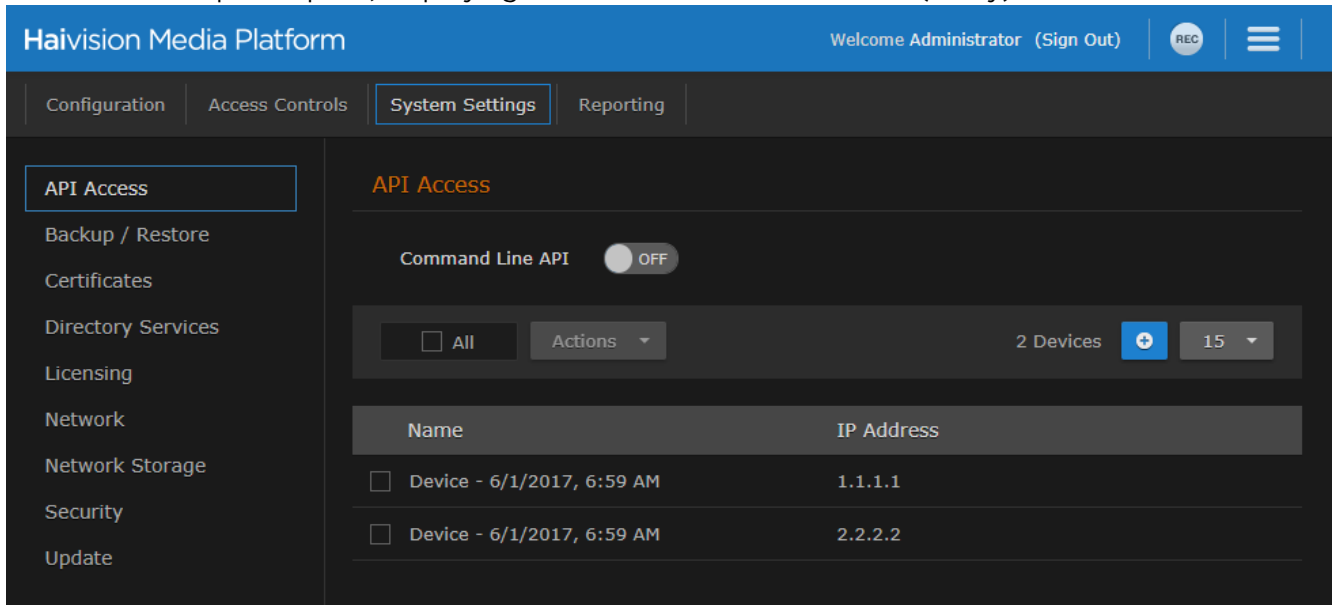


3. Toggle the Command Line API button to **On**.

> ⚠ **Note**
>
> Toggling this button activates *all* devices in the list.

4. To add a device to the list, click the ⊕ icon.
5. In the Add Device dialog, enter a unique name that identifies this device.
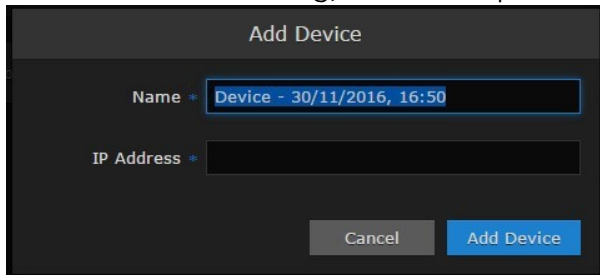


6. Enter the IP address of the device that will be accessing the Command Line API.
7. Click **Add Device**. The new device is added to the list of authorized devices.

> ⚠ **Note**
>
> Further changes to the list of authorized devices is applied automatically. You do not need to restart HMP.

# Command Line API Reference

This command reference lists and describes the available commands for the HMP Command Line API. These commands can be used to create and manage sessions and record and share video.

> ⚠ **Important**
>
> Ensure that you have activated Command Line API access on the HMP system and added your computer to the list of authorized devices. See Enabling Command Line API Access.

> ⚠ **Note**
>
> You must connect to the HMP Command Line network socket on TCP/IP port 23233. During development, the Telnet utility may be used to connect to HMP on this port, allowing you to execute commands interactively.

**Topics Include**

- **Summary of Command Line API Commands**
- **Sessions API**
  - **Create a Session**
  - **Set Live Session**
  - **Delete a Session**
  - **Add a Source to a Session**
  - **Remove a Source from a Session**
  - **Share a Session with a User**
  - **Share a Session with a Group**
  - **Update Session Metadata**
  - **Set Session Category**
- **Recordings API**
  - **Start a Recording**
  - **Pause a Recording**
  - **Resume a Recording**
  - **Stop a Recording**
  - **Get Recording Status**
  - **Share a Recording with a User**
  - **Share a Recording with a Group**
  - **Update Recording Metadata**
  - **Set Recording Category**
  - **Add a HotMark**
  - **Start Re-streaming a Recording**
  - **Stop Streaming a Recording**

## Summary of Command Line API Commands

| Command | Result | Use to... |
|---|---|---|
| `createSession` *sessionTitle* | *sessionId* | **Create a Session** |
| `setLiveSession` *sessionID active* | None | **Set Live Session** |
| `deleteSession` *sessionId* | None | **Delete a Session** |
| `addSourceToSession` *sessionId sourceId* | None | **Add a Source to a Session** |
| `removeSourceFromSession` *sessionId sourceId* | None | **Remove a Source from a Session** |
| `shareSessionWithUser` *username permissionId sessionId* | None | **Share a Session with a User** |
| `shareSessionWithGroup` *groupname permissionId sessionId* | None | **Share a Session with a Group** |
| `updateSessionMetadata` *sessionId* `title` \| `description` \| `createdby` *value* | None | **Update Session Metadata** |

| Command | Result | Use to... |
|---------|--------|-----------|
| `setSessionCategory` *sessionId categoryId value [value…]* | None | **Set Session Category** |
| `startRecording` *sessionId* | *recordingId* | **Start a Recording** |
| `pauseRecording` *recordingId* | None | **Pause a Recording** |
| `resumeRecording` *recordingId* | None | **Resume a Recording** |
| `stopRecording` *recordingId* | None | **Stop a Recording** |
| `getRecordingStatus` *recordingId* | *state* | **Get Recording Status** |
| `shareRecordingWithUser` *username permissionId recordingId* | None | **Share a Recording with a User** |
| `shareRecordingWithGroup` *groupname permissionId recordingId* | None | **Share a Recording with a Group** |
| `updateMetadata` *recordingId* `title \| description \| createdby` *value* | None | **Update Recording Metadata** |
| `setCategory` *recordingId categoryId value [value…]* | None | **Set Recording Category** |
| `addHotmark` *recordingId title* | None | **Add a HotMark** |
| `startRestreamRecording` *recordingId ip port* | *streamId* | **Start Re-streaming a Recording** |
| `stopRestreamRecording` *streamId* | None | **Stop Streaming a Recording** |

# Sessions API

Topics in this section:

- **Create a Session**
- **Set Live Session**
- **Delete a Session**
- **Add a Source to a Session**
- **Remove a Source from a Session**
- **Share a Session with a User**
- **Share a Session with a Group**
- **Update Session Metadata**
- **Set Session Category**

# Create a Session

This command creates a new session and returns a session ID. The session ID is required in later commands.

**Authorizations:** All

## Requests

```
createSession sessionTitle
```

## Parameters

| Parameter | Description/Values |
|-----------|-------------------|
| sessionTitle | Enter a title for the session. ⚠ **Note** The title must be enclosed in quotes if it contains one or more spaces. |

## Result

### Success

```
sessionId
```

### Error

| Error | Description |
|-------|-------------|
| ERROR\|010003\|Expected 1 parameter(s). Got Y parameter(s) | Calling `createSession` requires 1 parameter. |
| ERROR\|010001\|Session name must be at least 1 character | The session title cannot be empty string " ". |
| ERROR\|070019\|Failed to create a session | Creating a new session failed. Please see `calypso.log` for details. |

## Example

The following example creates a session entitled "Test Session1". The system returns OK and the ID of the new session.

```
createSession "Test Session1"

OK|71edd346-2cf0-4c62-b3d3-9d457b287394
```

## Example

The first example toggles the session Live setting/state to Live ( `active = "true"` ). The system returns `OK` .

```
setLiveSession 46713454-8e31-4efc-babc-6bc9b88f2834 true

OK
```

The second example toggles the session Live setting/state to not-Live ( `active = "false"` ). The system returns `OK` .

```
setLiveSession 46713454-8e31-4efc-babc-6bc9b88f2834 false

OK
```

# Delete a Session

This command deletes an existing session.          **Authorizations:** All

## Request

```
deleteSession sessionId
```

## Parameters

| Parameter | Description/Values |
|-----------|--------------------|
| sessionId | The ID of a HMP session |

## Result

**Success**

```
OK
```

**Error**

| Error | Description |
|-------|-------------|
| `ERROR|010003|Expected 1 parameter(s). Got Y parameter(s)` | Calling `deleteSession` requires 1 parameter. |
| `ERROR|010001|'Y' is not a uuid` | The `sessionId` provided must be a UUID. |
| `ERROR|040006|Session not found` | The 'session' with UUID '`sessionId`' was not found. |
| `ERROR|060003|Recording currently in progress` | The session is being recorded. All recordings for this session must be stopped before it can be removed. |
| `ERROR|070000|Internal Server Error` | Database returned an error while removing session. Please see `calypso.log` for details. |

## Example

The following example deletes a session entitled "Test Session1". The system returns OK.

```
deleteSession "Test Session1"

OK
```

# Add a Source to a Session

This command adds a source to a session specified by the session ID. You specify the source using the source ID.

**Authorizations:** All

> ✅ **Tip**
>
> The source ID is generated by HMP when the source is defined. To add a source to a session, go the Web interface (☰ > **Administration** > **Configuration** > **Sources** and click on the desired source) to get the source ID.
>
> 
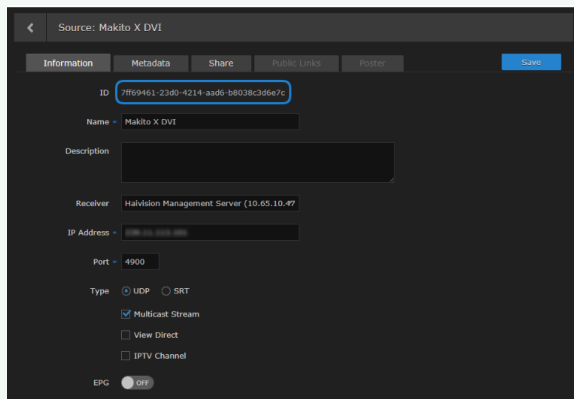
> ⚠️ **Note**
>
> To create a multi-source session, you simply add up to four sources to a session, executing the `addSourceToSession` command once for each source you wish to add.

## Requests

```
addSourceToSession sessionId sourceId
```

## Parameters

| Parameter | Description/Values |
|-----------|-------------------|
| sessionId | The ID of a HMP session. |
| sourceId | The ID of a HMP source. See the Tip above to obtain the desired source ID. |

## Result

**Success**

```
OK
```

**Error**

| Error | Description |
|---|---|
| `ERROR\|010003\|Expected 2 parameter(s). Got Y parameter(s)` | Calling `addSourceToSession` requires 2 parameters. |
| `ERROR\|010001\|'Y' is not a UUID` | The `sessionId` provided was not a valid UUID. OR The `sourceId` provided was not a valid UUID. |
| `ERROR\|060003\|Active recording currently in progress` | The session is being recorded. All recordings for this session must be stopped before a source can be added. |
| `ERROR\|070010\|Could not create a track` | Attempting to create a track for the source failed. |
| `ERROR\|040006\|Session not found` | The 'session' with UUID '`sessionId`' was not found. |
| `ERROR\|040009\|Source not found` | The 'source' with UUID '`sourceId`' was not found. |
| `ERROR\|060008\|Source already added to this session` | The source with UUID '`sourceId`' has already been added to this session. |

**Example**

The following example adds a source to the session specified by ID 71edd346-2cf0- 4c62-b3d3-9d457b287394. The system returns OK.

```
addSourceToSession 71edd346-2cf0-4c62-b3d3-9d457b287394
  b86befc3-7b44-4655-852f-21a77935d110

OK
```

# Remove a Source from a Session

This command removes the specified source from the session.

**Authorizations:** All

## Requests

```
removeSourceFromSession sessionId sourceId
```

## Parameters

| Parameter | Description/Values |
|-----------|--------------------|
| sessionId | The ID of a HMP session. |
| sourceId | The ID of a HMP source. See the Tip in **Add a Source to a Session** to obtain the source ID. |

## Result

### Success

```
OK
```

### Error

| Error | Description |
|-------|-------------|
| ERROR\|010003\|Expected 2 parameter(s). Got Y parameter(s) | Calling `removeSourceFromSession` requires 2 parameters. |
| ERROR\|010001\|'Y' is not a uuid | The `sessionId` provided was not a valid UUID. OR The `sourceId` provided was not a valid UUID. |
| ERROR\|040006\|Session not found | The 'session' with UUID '`sessionId`' was not found. |
| ERROR\|040009\|Source not found | The source with UUID '`sourceId`' was not found. |
| ERROR\|060003\|Active recording currently in progress | The session is being recorded. All recordings for this session must be stopped before a source can be added. |
| ERROR\|070000\|Internal Server Error | Removing source from session failed. Please see `calypso.log` for details. |

## Example

The following example removes the source from session ID 71edd346-2cf0-4c62- b3d3-9d457b287394. The system returns OK.

```
removeSourceFromSession 71edd346-2cf0-4c62-b3d3-9d457b287394
 b86befc3-7b44-4655-852f-21a77935d110

OK
```

# Share a Session with a User

This command shares the specified session with the specified user and sets the access permission (either WATCH, EDIT, RECORD, OWN, or custom permission ID).

**Authorizations:** All

## Request

```
shareSessionWithUser username permissionId sessionId
```

## Parameters

| Parameter | Description/Values |
|---|---|
| username | The unique name specified for the HMP user. `username` is case-insensitive. |
| permissionId | The access permission to assign to this user, either:<br>• WATCH: View this session<br>• EDIT: View and edit this session.<br>• RECORD: View and record this session.<br>• OWN: All capabilities, including deleting this session.<br>  OR<br>• \<uuid\> of admin-defined share permission. |
| sessionId | The ID of a HMP session. |

## Result

### Success

```
OK
```

### Error

| Error | Description |
|---|---|
| ERROR\|010003\|Expected 3 parameter(s).<br>  Got Y parameter(s) | Calling `shareSessionWithUser` requires 3 parameters. |
| ERROR\|010001\|'Y' is not a UUID | The `sessionId` provided must be a UUID. |
| ERROR\|010001\|Permission must be one of:<br>  WATCH, EDIT, RECORD, OWN | The 'permission' passed in was not one of the expected values. |
| ERROR\|070008\|Error searching for a user | An internal error occurred while searching for user 'username'. |

| Error | Description |
|-------|-------------|
| `ERROR\|040004\|User not found` | The user with UUID 'username' was not found. |
| `ERROR\|060001\|User has no roles assigned` | Make sure a role is assigned to the user or to at least one of the user's groups. |

| Error | Description |
|-------|-------------|
| `ERROR\|040006\|Session not found` | The session with UUID '`sessionId`' was not found. |
| `ERROR\|030003\|Cannot share` | Trying to share to a permission that user doesn't have access to. Ex.: User has permission to 'WATCH' and tries to share with 'OWN'. |

## Examples

This example uses a default HMP share permission:

```
shareSessionWithUser viewer EDIT 71edd346-2cf0-4c62-b3d3-9d457b287394

OK
```

This example uses a custom share permission.

```
shareSessionWithUser InfoDev 12b0cc54-9018-42c2-9dfb-3f7707a09829
  71edd346-2cf0-4c62-b3d3-9d457b287394

OK
```

# Share a Session with a Group

This command shares the specified session with the specified group and sets the access permission (either WATCH, EDIT, RECORD, OWN, or custom permission ID).

**Authorizations:** All

> ⚠ **Important**
>
> Groups are only available on HMP if your system is connected to an LDAP or Active Directory server. See "Managing Directory (Authentication) Services" in the *HMP Administrator's Guide*.

## Request

```
shareSessionWithGroup groupname permissionId sessionId
```

## Parameters

| Name | Description |
|------|-------------|
| groupname | The unique name specified for the HMP group. groupname is case-insensitive. |
| permissionId | The access permission to assign to this group, either:<br>• WATCH: View this session.<br>• EDIT: View and edit this session.<br>• RECORD: View and record this session.<br>• OWN: All capabilities, including deleting this session.<br>  OR:<br>• <uuid> of admin-defined permission. |
| sessionId | The ID of a HMP session. |

## Result

### Success

```
OK
```

### Error

| Error | Description |
|-------|-------------|
| ERROR\|010003\|Expected 3 parameter(s).<br> Got Y parameter(s) | Calling `shareSessionWithGroup` requires 3 parameters. |
| ERROR\|010001\|'Y' is not a UUID | The `sessionId` provided must be a UUID. |

| Error | Description |
|-------|-------------|
| `ERROR\|010001\|Permission must be one of:`<br>`  WATCH, EDIT, RECORD, OWN` | The 'permission' passed in was not one of the expected values. |
| `ERROR\|070009\|Error searching for a group` | An internal error occurred while searching for group '`groupname`'. |
| `ERROR\|040005\|Group not found` | The group with UUID '`groupName`' was not found. |
| `ERROR\|060002\|Group has no roles assigned` | Returned when a group has no roles. |
| `ERROR\|040006\|Session not found` | The session with UUID '`sessionId`' was not found. |
| `ERROR\|030003\|Cannot share` | Trying to share to a permission that user doesn't have access to.<br>Ex.: User has permission to 'WATCH' and tries to share with 'OWN'. |

### Example

This example uses a default HMP share permission.

```
shareSessionWithGroup Viewers RECORD 71edd346-2cf0-4c62-b3d3-9d457b287394

OK
```

This example uses a custom share permission.

```
shareSessionWithGroup Admins 12b0cc54-9018-42c2-9dfb-3f7707a09829
  71edd346-2cf0-4c62-b3d3-9d457b287394

OK
```

# Update Session Metadata

This command updates the metadata assigned to the specified session (the contents of either the title, description, or createdby field).

**Authorizations:** All

## Request

```
updateSessionMetadata sessionId title|description|createdby value
```

## Parameters

| Parameter | Description/Values |
|-----------|--------------------|
| sessionId | The ID of a HMP session. |
| fieldName | The metadata field to update, either `title,` `description` or `createdby` |
| value | The contents of the metadata field. |

## Result

### Success

```
OK
```

### Error

| Error | Description |
|-------|-------------|
| `ERROR\|010003\|Expected 3 parameter(s). Got Y parameter(s)` | Calling `updateSessionMetadata` requires 3 parameters. |
| `ERROR\|010001\|'Y' is not a UUID` | The `sessionId` provided must be a UUID. |
| `ERROR\|010001\|'Z' is an invalid field name` | The `fieldName` provided is invalid. |
| `ERROR\|040006\|Session not found` | A session with UUID '`sessionId`' was not found. |
| `ERROR\|070000\|Internal Server Error` | Updating the session failed. Please see `calypso.log` for details. |

## Example

```
updateMetadata 71edd346-2cf0-4c62-b3d3-9d457b287394 createdby Test2

OK
```

# Set Session Category

This command sets the value for a category assigned to the specified session. You specify the category using the category ID.

**Authorizations:** All

> ✅ **Tip**
>
> The category ID is generated by HMP when the category is defined. To assign category to a session or recording, go the Web interface (☰ > **Administration** > **Configuration** > **Metadata** and click on the metadata entry) to get the category ID.
>
> 
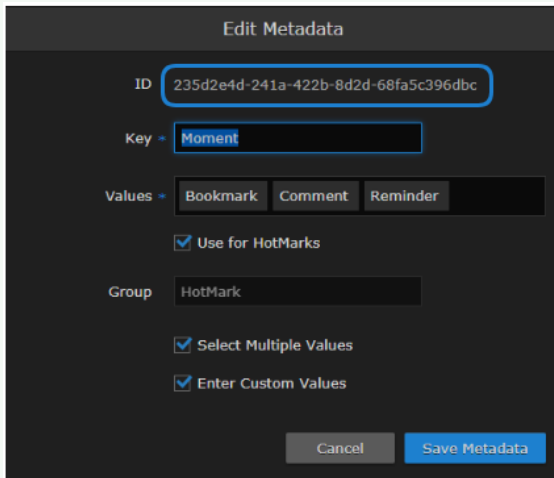
> ⚠️ **Note**
>
> When using this command, the (Web interface's) Multiple Values and Custom Values settings are ignored. The command always allows multiple and custom values regardless of the checkbox settings.

## Request

```
setSessionCategory sessionId categoryId value [value...]
```

## Parameters

| Parameter | Description/Values |
|---|---|
| sessionId | The ID of a HMP session. |
| categoryId | The ID of the category to assign to the session. See the tip above to obtain category ID. |
| value | The value defined for the category. Multiple values are separated by spaces. |

## Result

**Success**

> `OK`

**Error**

| Error | Description |
|-------|-------------|
| `ERROR\|010003\|Expected 3 parameter(s).`<br>` Got Y parameter(s)` | Calling `setSessionCategory` requires at least 3 parameters. |
| `ERROR\|010001\|'Y' is not a UUID` | The `sessionId` provided must be a UUID. The `categoryId` provided must be a UUID. |
| `ERROR\|010001\|Values must be strings` | Categories must be strings. |
| `ERROR\|010001\|Please provide non-empty values` | Categories cannot be empty |
| `ERROR\|010001\|Please remove any leading or`<br>` trailing spaces in the values` | Categories cannot start or end with spaces. |
| `ERROR\|010001\|Please verify that values are unique` | Category list passed in contains duplicate entries. |
| `ERROR\|040007\|Category not found` | A category with UUID '`categoryId`' was not found. |
| `ERROR\|040006\|Session not found` | A session with UUID '`sessionId`' was not found. |

## Example

```
setSessionCategory 71edd346-2cf0-4c62-b3d3-9d457b287394
  692e9ab8-d446-4bfd-b64d-e39e8e57c4c5 Surgery

OK
```

# Recordings API

Topics in this section:

- **Start a Recording**
- **Pause a Recording**
- **Resume a Recording**
- **Stop a Recording**
- **Get Recording Status**
- **Share a Recording with a User**
- **Share a Recording with a Group**
- **Update Recording Metadata**
- **Set Recording Category**
- **Add a HotMark**
- **Start Re-streaming a Recording**
- **Stop Streaming a Recording**

# Start a Recording

This command creates an active recording for specified session and returns its ID. You will use the recording ID in later commands.

**Authorizations:** All

> ⚠ **Note**
>
> If you record a multi-source session, you get a multi-stream recording, whereas if you record a session with one source, you get a one-stream recording. The Command Line API is the same in either case; you are always working with a single recording ID.

> ⚠ **Note**
>
> The Command Line API only records the session intact. For example, you cannot use it to record only two of the sources for a four-source session.

## Request

```
startRecording sessionId
```

## Parameters

| Parameter | Description/Values |
|---|---|
| sessionId | The ID of a HMP session. |

## Result

### Success

```
recordingId
```

### Error

| Error | Description |
|---|---|
| ERROR\|010003\|Expected 1 parameter(s).<br> Got Y parameter(s) | Calling `startRecording` requires 1 parameter. |
| ERROR\|010001\|'Y' is not a UUID | The `sessionId` provided must be a UUID. |
| ERROR\|040006\|Session not found | The session with UUID '`sessionId`' was not found. |

| Error | Description |
|---|---|
| `ERROR|0700014|Error writing to file` | An internal error occurred while attempting to write to the disk. Please see `calypso.log` for details. |
| `ERROR|070000|Internal Server Error` | The recording could not be started due to an internal error. Please see `calypso.log` for details. |
| `ERROR|070027|Insufficient Disk Space` | Not enough disk space on server for creating a recording. |
| `ERROR|060009|Session requires at least one source` | The session does not have a source. |

## Example

```
startRecording 71edd346-2cf0-4c62-b3d3-9d457b287394

OK|f678f415-0665-48cd-b141-1e95283469f1
```

## Pause a Recording

This command pauses the specified active recording.

**Authorizations:** All

### Request

```
pauseRecording recordingId
```

### Parameters

| Parameter | Description/Values |
|---|---|
| recordingId | The ID of a HMP recording. |

### Result

**Success**

```
OK
```

**Error**

| Error | Description |
|---|---|
| ERROR\|010003\|Expected 1 parameter(s). Got Y parameter(s) | Calling `pauseRecording` requires 1 parameter. |
| ERROR\|010001\|'Y' is not a UUID | The `recordingId` provided must be a UUID. |
| ERROR\|040001\|Active recording not found | The recording with UUID '`recordingId`' was not found. |
| ERROR\|070003\|Failed to update state | Pausing recording failed. Please see `calypso.log` for details. |

### Example

```
pauseRecording f678f415-0665-48cd-b141-1e95283469f1

OK
```

## Resume a Recording

This command resumes the specified paused, active recording.

**Authorizations:** All

### Request

```
resumeRecording recordingId
```

### Parameters

| Name | Description |
|------|-------------|
| recordingId | The ID of a HMP recording. |

### Result

**Success**

```
OK
```

**Error**

| Error | Description |
|-------|-------------|
| ERROR\|010003\|Expected 1 parameter(s). Got Y parameter(s) | Calling `resumeRecording` requires 1 parameter. |
| ERROR\|010001\|'Y' is not a UUID | The `recordingId` provided must be a UUID. |
| ERROR\|040001\|Active recording not found | The recording with UUID '`recordingId`' was not found. |
| ERROR\|070003\|Failed to update state | Resuming recording failed. Please see `calypso.log` for details. |

### Example

```
resumeRecording f678f415-0665-48cd-b141-1e95283469f1

OK
```

## Stop a Recording

This command stops the specified active recording.

**Authorizations:** All

### Request

```
stopRecording recordingId
```

### Parameters

| Name | Description |
|------|-------------|
| recordingId | The ID of a HMP recording. |

### Result

#### Success

```
OK
```

#### Error

| Error | Description |
|-------|-------------|
| ERROR\|010003\|Expected 1 parameter(s). Got Y parameter(s) | Calling `stopRecording` requires 1 parameter. |
| ERROR\|010001\|'Y' is not a UUID | The `recordingId` provided must be a UUID. |
| ERROR\|040001\|Active recording not found | The recording with UUID '`recordingId`' was not found. |
| ERROR\|070003\|Internal Server Error | An internal error occurred while stopping the recording. Please see `calypso.log` for details. |

### Example

```
stopRecording f678f415-0665-48cd-b141-1e95283469f1

OK
```

# Get Recording Status

This command gets the status of the specified active recording.

**Authorizations:** All

## Request

```
getRecordingStatus recordingId
```

## Parameters

| Name | Description |
| --- | --- |
| recordingId | The ID of a HMP recording. |

## Result

### Success

```
state
```

where `state` could be one of the following:

- RECORDING: It is currently being recorded (active recording).
- PAUSED: Recording has already started but is currently paused.
- FINISHED: Recording has finished recording.
- ERROR: Any other error on the recording will be set to this state.

### Error

| Error | Description |
| --- | --- |
| ERROR\|010003\|Expected 1 parameter(s). Got Y parameter(s) | Calling `getRecordingStatus` requires 1 parameter. |
| ERROR\|010001\|'Y' is not a UUID | The `recordingId` provided must be a UUID. |
| ERROR\|040001\|Active recording not found | The recording with UUID '`recordingId`' was not found. |
| ERROR\|070003\|Recording not found | A recording with UUID `'recordingId'` was not found. |

**Example**

```
getRecordingStatus f678f415-0665-48cd-b141-1e95283469f1

OK|PAUSED
```

# Share a Recording with a User

This command shares the specified recording with the specified user and sets the access permission (either `WATCH, EDIT, OWN,` or custom permission ID).

**Authorizations:** All

## Request

```
shareRecordingWithUser username permissionId recordingId
```

## Parameters

| Name | Description |
|------|-------------|
| username | The HMP user name specified for the user. username is case-insensitive. |
| permissionId | The access permission to assign to this user, either:<br>• WATCH: View this recording<br>• EDIT: View and edit this recording<br>• OWN: All capabilities, including deleting this recording, or<br>• <uuid> of admin-defined share permission |
| recordingId | The ID of a HMP recording. |

## Result

### Success

```
OK
```

### Error

| Error | Description |
|-------|-------------|
| ERROR\|010003\|Expected 3 parameter(s).<br>  Got Y parameter(s) | Calling `shareRecordingWithUser` requires 3 parameter. |
| ERROR\|010001\|'Y' is not a UUID | The `recordingId` provided must be a UUID. |
| ERROR\|010001\|Permission must be one of:<br>  WATCH, EDIT, OWN | Validation failed on `'permission'`. Verify that it is set to `WATCH`, `EDIT`, or `OWN`. |
| ERROR\|070008\|Error searching for a user | An internal error occurred while searching for user `'username'`. |
| ERROR\|040004\|User not found | A user with UUID `'username'` was not found. |

| Error | Description |
|-------|-------------|
| `ERROR\|060001\|User has no roles assigned` | Make sure a role is assigned to the user or to at least one of the user's groups. |
| `ERROR\|040002\|Recording not found` | A recording with UUID `'recordingId'` was not found. |
| `ERROR\|030003\|Cannot share` | Trying to share to a permission that user doesn't have access to. Ex.: User has permission to 'WATCH' and tries to share with 'OWN'. |

**Example**

This example uses a default HMP share permission.

```
shareRecordingWithUser creator OWN f678f415-0665-48cd-b141- 1e95283469f1
OK
```

This example uses a custom share permission.

```
shareRecordingWithUser InfoDev 12b0cc54-9018-42c2-9dfb-3f7707a09829
 f678f415-0665-48cd-b141-1e95283469f1

OK
```

# Share a Recording with a Group

This command shares the specified recording with the specified group and sets the access permission (either WATCH, EDIT, OWN, or custom permission ID).

**Authorizations:** All

## Request

```
shareRecordingWithGroup groupname permissionId recordingId
```

## Parameters

| Name | Description |
|------|-------------|
| groupname | The name specified for the group. groupname is case-insensitive. |
| permissionId | The access permission to assign to this group, either:<br>• WATCH: View this recording<br>• EDIT: View and edit this recording<br>• OWN: All capabilities, including deleting this recording, or<br>• <uuid> of admin-defined share permission. |
| recordingId | The ID of an HMP recording. |

## Result

### Success

```
OK
```

### Error

Make sure a role is assigned to the group

| Error | Description |
|-------|-------------|
| `ERROR\|010003\|Expected 3 parameter(s).   Got Y parameter(s)` | Calling `shareRecordingWithGroup` requires 3 parameter. |
| `ERROR\|010001\|'Y' is not a UUID` | The `recordingId` provided must be a UUID. |
| `ERROR\|010001\|Permission must be one of:   WATCH, EDIT, OWN` | Validation failed on 'permission'. Verify that it is set to `WATCH,` `EDIT,` or `OWN`. |
| `ERROR\|070009\|Error searching for a group` | An internal error occurred while searching for group `'groupname'`. |

| Error | Description |
|-------|-------------|
| `ERROR\|040005\|Group not found` | A group with UUID `'groupname'` was not found. |
| `ERROR\|060002\|Group has no roles assigned` | Make sure a role is assigned to the group. |
| `ERROR\|040002\|Recording not found` | A recording with UUID `'recordingId'` was not found. |
| `ERROR\|030003\|Cannot share` | Trying to share to a permission that user doesn't have access to. Ex.: User has permission to 'WATCH' and tries to share with 'OWN'. |

## Example

This example uses a default HMP share permission.

```
shareRecordingWithGroup Grp10 OWN f678f415-0665-48cd-b141-1e95283469f1

OK
```

This example uses a custom share permission.

```
shareRecordingWithGroup Grp10 12b0cc54-9018-42c2-9dfb3f7707a09829
  f678f415-0665-48cd-b141-1e95283469f1

OK
```

# Update Recording Metadata

This command updates the metadata assigned to the specified recording (the contents of either the `title`, `description`, or `createdby` field).

**Authorizations:** All

## Requests

```
updateMetadata recordingId title|description|createdby value
```

## Parameters

| Name | Description |
|------|-------------|
| recordingId | The ID of a HMP recording. |
| fieldName | The metadata field to update, either `title`, `description` or `createdby`.<br><br>⚠️ **Note**<br><br>"createdby" is the display name on the Edit User page (☰ > **Administration** > **Access Controls** > **Users**). createdby is also the creator in the "Created M/D/YYYY by [creator]" status displayed in the Videos or Session player window |
| value | The contents of the metadata field. |

## Result

### Success

```
OK
```

### Error

| Error | Description |
|-------|-------------|
| `ERROR\|010003\|Expected 3 parameter(s). Got Y parameter(s)` | Calling `updateMetadata` requires 3 parameters. |
| `ERROR\|010001\|'Y' is not a UUID` | The `recordingId` provided must be a UUID. |
| `ERROR\|040002\|Recording not found` | A recording with UUID `'recordingId'` was not found. |
| `ERROR\|070017\|Failed to update a recording` | An internal error occurred while updating the metadata. Please see `calypso.log` for details. |

| Error | Description |
|---|---|
| `ERROR\|010001\|'Z' is an invalid field name` | The `fieldName` provided is invalid |
| `ERROR\|040012\|No results` | The list of metadata was not returned from the server. |

## Example

```
updateMetadata f678f415-0665-48cd-b141-1e95283469f1 title Test2

OK
```

# Set Recording Category

This command sets the value for a category assigned to the specified recording. You specify the category using the category ID.

**Authorizations:** All

## Request

```
setCategory recordingId categoryId value [value...]
```

## Parameters

| Name | Description |
|---|---|
| recordingId | The ID of a HMP recording. |
| categoryId | The ID of the category to assign to the recording. See the Tip in **Set Session Category** to obtain the category ID. |
| value | The value defined for the category. Multiple values are separated by spaces. |

## Result

### Success

```
OK
```

### Error

| Errors | Description |
|---|---|
| `ERROR|010003|Expected at least 3 parameters.`<br>`  Got Y parameter(s)` | Calling `setCategory` requires at least 3 parameters |
| `ERROR|010001|'Y' is not a UUID` | The `recordingId` and `categoryId` provided must be a UUID. |
| `ERROR|010001|Values must be strings` | Categories must be strings. |
| `ERROR|010001|Please provide non-empty values` | Categories cannot be empty. |
| `ERROR|010001|Please remove any leading or`<br>`  trailing spaces in the values` | Categories cannot start or end with spaces. |
| `ERROR|010001|Please verify that values`<br>` are unique` | Category list passed in contains duplicate entries. |
| `ERROR|040007|Category not found` | A category with UUID `'categoryId'` was not found. |

| Errors | Description |
|---|---|
| `ERROR\|040002\|Recording not found` | A recording with UUID `'recordingId'` was not found. |
| `ERROR\|040012\|No results` | The list of metadata was not returned from the server. |

## Example

```
setCategory f678f415-0665-48cd-b141-1e95283469f1
   692e9ab8-d446-4bfd-b64d-e39e8e57c4c5 HD

OK
```

# Add a HotMark

A HotMark is a metadata "bookmark" that you can add to annotate an active recording.

HotMarks can also be added to completed recordings using the HMP Web Interface. Viewers can then select the HotMarks to jump directly to specific points of interest in the recording.

This command adds a HotMark to an active recording now (i.e., at the recording's current timestamp).

**Authorizations:** All

> ⚠️ **Note**
>
> HMP rounds the "now" timestamp down to the nearest second.

## Request

```
addHotmark recordingId title
```

## Parameters

| Name | Description |
|------|-------------|
| recordingId | The ID of a HMP recording. |
| title | The title to assign to HotMark. |

## Result

**Success**

```
OK
```

**Error**

| Error | Description |
|---|---|
| `ERROR\|010003\|Expected 2 parameters. Got Y parameter(s)` | Calling `addHotmark` requires 2 parameters. |
| `ERROR\|010001\|'Y' is not a UUID` | The `recordingId` provided must be a UUID. |
| `ERROR\|040002\|Recording not found` | A recording with UUID `'recordingId'` was not found. |
| `ERROR\|060004\|Hotmark already exists at this time` | A HotMark at the specified timestamp already exists. |
| `ERROR\|010004\|'Y' is not an active recording` | Returned if the recording is no longer active. Y is the `recordingId`. |
| `ERROR\|070011\|Could not locate recording` | An internal error occurred while adding HotMark. Please see `calypso.log` for details. |

## Example

```
addHotmark f678f415-0665-48cd-b141-1e95283469f1 PhaseI

OK
```

# Start Re-streaming a Recording

This command creates a stream for the specified recording (asset) and returns its ID. You specify the destination IP address and port.

**Authorizations:** All

## Request

```
startRestreamRecording recordingId ip port
```

## Parameters

| Name | Description |
|------|-------------|
| recordingId | The ID of a HMP recording. |
| ip | The destination unicast or multicast IP address. |
| port | The destination Port number. |

## Result

### Success

```
streamId
```

### Error

| Error | Description |
|-------|-------------|
| `ERROR\|010001\|'Y' is not a UUID` | The `recordingId` provided must be a UUID. |
| `ERROR\|010001\|'Y' is not an ip address` | The IP Address must follow the IPv4 dotted decimal notation. |
| `ERROR\|010001\|'Y' is not a port number` | Port must be between 0 and 65535. |
| `ERROR\|040002\|Recording not found` | A recording with UUID `'recordingId'` was not found. |
| `ERROR\|010005\|Only single source recordings can be streamed` | `recordingId` is a UUID of a multi-sourced session. |
| `ERROR\|010006\|Address or port already in use` | The specified IP address or port is currently in use by the session. |
| `ERROR\|010006\|Address or port already in use by a source` | The specified IP address or port is currently in use by the source. |

| Error | Description |
|-------|-------------|
| `ERROR\|070004\|Error while starting stream` | An internal error occurred while starting the `restream`. Please see `calypso.log` for details. |

**Example**

```
startRestreamRecording f678f415-0665-48cd-b141-1e95283469f1
 10.65.10.240 222

OK|bfdac49c-beb7-4271-8acb-de79f3fd2ddf
```

# Stop Streaming a Recording

This command stops streaming the specified recording (asset).

**Authorizations:** All

## Request

```
stopRestreamRecording streamId
```

## Parameters

| Parameter | Description/Values |
|-----------|-------------------|
| streamId | The ID of a HMP stream. |

## Result

### Success

```
OK
```

### Error

| Error | Description |
|-------|-------------|
| `ERROR\|010001\|'Y' is not a UUID` | The `streamId` provided must be a UUID. |
| `ERROR\|070005\|Error while stopping stream` | An internal error occurred when attempting to stop the stream. Please see `calypso.log` for details. |
| `ERROR\|040003\|Stream not found` | A stream with UUID `'streamId'` was not found. |

## Example

```
stopRestreamRecording bfdac49c-beb7-4271-8acb-de79f3fd2ddf

OK
```

# General Errors

This command reference lists and describes the general errors for the HMP Command Line API. If an error occurs, the response has the following format: " `ERROR|Code|Error message` ".

> ⚠️ **Note**
>
> Error codes that begin with "07" are internal system errors. Please contact Haivision Technical Support if you encounter this error code.

The following is the list of general Command Line API error messages:

| Code | Error | Description |
|------|-------|-------------|
| 000000 | `ERROR|000000|Unknown Error` | Unknown error |
| 000001 | `ERROR|000001|Syntax  Error` | For any parameter which was enclosed within quotes, this error will be returned if quotes are not balanced. (Ex.: `createSession "asdf` returns this error.) |
| 000002 | `ERROR|000002|Command not found` | Command entered by user does not exist. |
| 010001 | `ERROR|010001|'Y' is not a UUID` | For any parameter ending with "Id" (i.e., `:recordingId` , `:sessionId` , `:categoryId` ), the parameter provided must be a UUID. ("Y'" displays the wrong parameter text provided by the user.) |
| 010003 | `ERROR|01003|Expected X parameter(s). Got Y parameter(s)` | For a command with a required number of parameters, the user must provide all the parameters. Or for a command with variable number of parameters. (Ex.: `setCategory` and `setSessionCategory` ), the user must provide all required parameters. |
| 020000 | `ERROR|020000|Not Authorized` | User tries to connect to the Command Line API but is not authorized. |
| 070000 | `ERROR|070000|Internal Server Error` | This error occurs when the service is not available. (Ex.: HMP is restarted and command is sent while HMP is not yet ready.) This is a more general error message that could occur if something else goes wrong and requires investigation. Please see `calypso.log` for details. |
| 070006 | `ERROR|070006|Bus client error` | There was an internal bus communication error. |

| Code | Error | Description |
|------|-------|-------------|
| 070007 | `ERROR|070007|Bus response parsing error` | There was an error parsing the response of a bus request. |
| 090001 | `ERROR|090001|No bus connection` | There is no connection to the bus. |

# 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.<br>• Ability to add or modify objects and metadata through the application programming interface ("API")<br>• 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<br>• Ability for Audience Members to authenticate through the Access Service<br>• Ability for Audience Members to play Published Media Objects<br>• Ability for Audience Members to play Haivision VCMS-authenticated or entitled Published Media Objects<br>• 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<br>• Addition of new features/functions<br>• 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)<br>**1 (877) 224-5445**<br><br>International<br>**1 (514) 334-5445**<br><br>*and choose from the following:*<br>Sales - 1, Cloud Services - 3, Support - 4 |
|---|---|
| **Managed Services** | U.S. and International<br>1 (512) 220-3463 |
| **Fax** | 1 (514) 334-0088 |
| **Support Portal** | https://support.haivision.com |
| **Product Information** | info@haivision.com |