

# SwitchEngine API

## Customer Integration Guide

Prepared by: Christopher Deas, *Product Manager*.

30/03/2021

Changes:

# Table of Contents

<b>Introduction</b>	<b>1</b>
<b>Overview</b>	<b>1</b>
Goal	
Constraints	
Requirements	
Architecture	
<b>Getting Started</b>	<b>3</b>
<b>Launch Switch</b>	<b>3</b>
Hyperlink	
Command Line	
Manually	
<b>Connect</b>	<b>6</b>

<b>Video</b>	<b>7</b>
<b>Commands</b>	<b>9</b>
<b>Inspection</b>	<b>9</b>
Inspect	
<b>Audio</b>	<b>13</b>
Send	
<b>Navigation</b>	<b>16</b>
Stop	
Play	
Slow	
Step	
Skip	
Next	
Previous	
Loop	
Jump	
<b>Closed Captions &amp; Subtitles</b>	<b>20</b>
Captions	
<b>Video Images</b>	<b>20</b>
Image	
<b>Editing</b>	<b>22</b>
Video	
Gutters	
<b>Secondary Files</b>	<b>23</b>
Add	
Remove	
<b>Supported Timecode Sources</b>	<b>26</b>

<b>Troubleshooting</b>	<b>28</b>
Mac	28
Windows	28
<b>Debug Logging</b>	<b>29</b>
Enabling on Mac	29
Enabling on Windows	29
Log Levels	29
<b>Sample Code</b>	<b>31</b>

## Overview

### Goal

This API is designed to provide the playback and inspection capabilities of Switch within web-based applications. Web browsers do not support the majority of video containers and codecs that are required for professional video applications, but Switch does. Switch also brings the same professional-level capabilities for captions, subtitles, timecode, inspection, and metadata.

### Constraints

Future versions of Switch may bring additional features and capabilities for this API, but for this, the following constraints have been imposed:

1. All media files must be accessible to Switch via the local file system on the client computer. This includes directattached storage (DAS), storage area networks (SAN), or network-attached storage (NAS), such as via SMB, or AFP. Remote media, such as that available via HTTP, FTP, or Aspera, is not supported. Streaming media, such as that available via RTSP, RTP, MMS, HTTP, etc. is not supported.
2. Switch must be running on the same computer as the web browser running the web application.
3. Each SwitchEngine server will support only a single video, along with multiple secondary audio, and/or secondary subtitle files. However, multiple SwitchEngine servers can be run from a single instance of the Switch application. To change videos, a new SwitchEngine server must be created.
4. Each SwitchEngine server will support only a single client connection at one time.
5. The API can not be used for exporting from Switch. Switch is not intended for batch exports. That should be done via other Telestream applications, such as Vantage.
6. Due to the additional overhead and lossy nature of compressing video as JPEG images, transmitting them over WebSockets, and then decompressing them again, do not expect the playback experience within a web browser to be of the same quality as what is experienced while playing within the Switch application. You may see an increase in dropped frames, as well as degradation in picture clarity and color.

7. The websocket connection between the client application and Switch must be bidirectional. Switch will begin sending messages whenever video changes, and if those messages are not accepted it will eventually cause problems. This does not seem to be an issue for web applications, but when using the Windows WebSocket API, you may have to add additional code to accept these messages.

## Requirements

The SwitchEngine requires the following:

1. Switch 4.5 or later must be installed on the client computer. Refer to [Telestream's website for Switch's minimum system requirements](#) for macOS and Windows.
2. The web application must be run in a supported web browser. The following web browsers are supported:

	Mac	Windows
Safari	Version 11.0.1 or later	None
Chrome	Version 62.0.3202.94 or later	Version 62.0.3202.94 or later
Firefox	Version 55.0.3 or later	Version 57.0 or later

## Architecture

### JavaScript, WebSockets, & JSON

Web applications can use [JavaScript WebSockets](#) to communicate with Switch. Data sent from Switch to the web application will be in a [custom binary format](#) for video and audio, and [JSON](#) for everything else. Video frames are compressed as maximum-quality [JPEG](#) images.

# Getting Started

## Launch Switch

Switch can be launched by hyperlink, command line, or manually.

### Hyperlink

Switch installs a custom URL handler that allows it to respond to hyperlinks that begin with the `switch://` scheme. After the scheme, the URL must contain either an `open` command, which launches Switch, opens the file, and waits for user interaction:

```
switch://open?file=/Users/telestream/Desktop/DV%20with%20Captions.mov
```

...or a `server` command, which launches Switch, opens the file, starts a SwitchEngine server, and waits for WebSocket interaction:

```
switch://server?port=6789&file=/Users/telestream/Desktop/DV%20with%20Captions.mov
```

Both of these commands take a `file` parameter, which must be an absolute path that is appropriate for the current platform (Mac or Windows). The path, and the entire URL, must be [URL encoded](#). For example, note the use of “%20” instead of spaces. If the client computer’s file system is case sensitive, make sure that the path matches what the file system expects in terms of case. On Windows, the path must be a UNC path. It is the web application’s responsibility to ensure that the media file exists and can be found by the client machine at the supplied path.

The `server` command also takes a `port` parameter, allowing you to specify which port the server should listen on. It is the web application’s responsibility to choose an available port.

The `open` command also allows you to set an initial display time, using any of the following parameters: `jumptime`, `jumpframe`, or `jumptimecode`. If you specify more than one of these parameters, one will be chosen and the other ignored. For the `jumptime` parameter, the value should be in fractional seconds.

```
switch://open?file=/Users/telestream/Desktop/DV%20withCaptions.mov&jumptime=21.359
```

For the `jumpframe` parameter, the value should be a zero-based frame index.

```
switch://open?file=/Users/telestream/Desktop/DV%20withCaptions.mov&jumpframe=643
```

For the `jumptotimecode` parameter, the value should be a timecode that matches the format of the video track's timecode, including the scale, modulus, duration, and drop-frame format. The timecode value should also be relative to the timecode of the first video frame. For example, if the first frame's timecode is 01:00:00;00 and you want to jump to the third frame, the value should be 01:00:00;02. Discontiguous timecode is not supported. You can not jump to a timecode in audio-only files. Files may have multiple timecodes for any given frame. Therefore, you must include the `timecodesource` parameter to specify which timecode source to use. See the [table below](#) for a list of supported timecode sources and the corresponding four-character-code values you can use. If you omit the `timecodesource` parameter, Switch will assume you want to use the zero-based timecode made by Switch as the source. The fourcharacter codes are case sensitive.

```
switch://open?file=/Users/telestream/Desktop/  
DV%20withCaptions.mov&jumptimecode=01:00:21;13&timecodesource=movt
```

You can put these hyperlinks in your web application and wait for the user to click on them with the following HTML:

#### HTML

```
<a href="switch://server?port=6789&file=/Users/telestream/Desktop/  
DV%20with%20Captions.mov">Launch</a>
```

...or you can trigger them programmatically using JavaScript:

#### JavaScript

```
location.href = "switch://server?port=6789&file=/Users/telestream/Desktop/  
DV%20with%20Captions.mov";
```

## Command Line

### Mac

To launch Switch via the command line or Terminal application on macOS, use the following command:

```
open /Applications/Switch.app/Contents/MacOS/Switch
```

To launch Switch, open a file, and wait for user interaction on macOS, use the following command:

```
/Applications/Switch.app/Contents/MacOS/Switch "switch://open?file=/Users/telestream/  
Desktop/DV%20with%20Captions.mov"
```

To launch Switch, open a file, jump to a particular time in fractional seconds, and wait for user interaction on macOS, use the following command:

```
/Applications/Switch.app/Contents/MacOS/Switch "switch://open?file=/Users/telestream/  
Desktop/DV%20withCaptions.mov&jumptime=21.359"
```

To launch Switch, open a file, jump to a particular zero-based frame index, and wait for user interaction on macOS, use the following command:

```
/Applications/Switch.app/Contents/MacOS/Switch "switch://open?file=/Users/telestream/Desktop/DV%20withCaptions.mov&jumptoframe=643"
```

To launch Switch, open a file, jump to a particular timecode, and wait for user interaction on macOS, use the following command:

```
/Applications/Switch.app/Contents/MacOS/Switch "switch://open?file=/Users/telestream/Desktop/DV%20withCaptions.mov&jumptotimecode=01:00:21;13&timecodesource=movt"
```

The timecode must match the format of the video track's timecode, including the scale, modulus, duration, and dropframe format. The timecode value should also be relative to the timecode of the first video frame. For example, if the first frame's timecode is 01:00:00;00 and you want to jump to the third frame, the value should be 01:00:00;02. Discontiguous timecode is not supported. You can not jump to a timecode in audio-only files. Files may have multiple timecodes for any given frame. Therefore, you must include the `timecodesource` parameter to specify which timecode source to use. See the [table below](#) for a list of supported timecode sources and the corresponding four-character-code values you can use. If you omit the `timecodesource` parameter, Switch will assume you want to use the zero-based timecode made by Switch as the source. The four-character codes are case sensitive.

To launch Switch, open a file, start a SwitchEngine server, and wait for WebSocket interaction on macOS, use the following command:

```
/Applications/Switch.app/Contents/MacOS/Switch "switch://server?port=6789&file=/Users/telestream/Desktop/DV%20with%20Captions.mov"
```

## Windows

To launch Switch via the command prompt on Windows, use the following command:

```
start /b "" "C:\Program Files\Telestream\Switch 4.5\Switch.exe"
```

To launch Switch, open a file, and wait for user interaction on Windows, use the following command:

```
start /b "" "C:\Program Files\Telestream\Switch 4.5\Switch.exe" "switch://open?file=\Mac\Home\Desktop\DVwithCaptions.mov"
```

To launch Switch, open a file, jump to a particular time in fractional seconds, and wait for user interaction on Windows, use the following command:

```
start /b "" "C:\Program Files\Telestream\Switch 4.5\Switch.exe" "switch://open?file=\Mac\Home\Desktop\DVwithCaptions.mov&jumptotime=21.359"
```

To launch Switch, open a file, jump to a particular zero-based frame index, and wait for user interaction on Windows, use

```
start /b "" "C:\Program Files\Telestream\Switch 4.5\Switch.exe" "switch://open?file=\
\Mac\Home\Desktop\DVwithCaptions.mov&jumptoframe=643"
```

To launch Switch, open a file, jump to a particular timecode, and wait for user interaction on Windows, use the following command:

```
start /b "" "C:\Program Files\Telestream\Switch 4.5\Switch.exe" "switch://open?file=\
\Mac\Home\Desktop\DVwithCaptions.mov&jumptotimecode=01:00:21;13"
```

The timecode must match the format of the video track's timecode, including the scale, modulus, duration, and dropframe format. The timecode value should also be relative to the timecode of the first video frame. For example, if the first frame's timecode is 01:00:00;00 and you want to jump to the third frame, the value should be 01:00:00;02. Discontiguous timecode is not supported. You can not jump to a timecode in audio-only files. Files may have multiple timecodes for any given frame. Therefore, you must include the `timecodesource` parameter to specify which timecode source to use. See the [table below](#) for a list of supported timecode sources and the corresponding four-character-code values you can use. If you omit the `timecodesource` parameter, Switch will assume you want to use the zero-based timecode made by Switch as the source. The four-character codes are case sensitive.

To launch Switch, open a file, start a SwitchEngine server, and wait for WebSocket interaction on Windows, use the following command:

```
start /b "" "C:\Program Files\Telestream\Switch 4.5\Switch.exe" "switch://server?
port=6789&file=\Mac\Home\Desktop\DVwithCaptions.mov"
```

## Manually

On macOS, Switch can be launched by double-clicking it in the Finder, clicking on it in the Dock, typing "Switch" into Spotlight (command + space), or by opening a media file that is associated with Switch.

On Windows, Switch can be launched by double-clicking it in the File Explorer, clicking on it in the Task Bar, typing "Switch" into the Start Menu's search field, typing "Switch" into the Run dialog, or by opening a media file that is associated with Switch.

The SwitchEngine server can NOT be launched manually from within the application. However, once the application is running, you can still start SwitchEngine servers using hyperlinks or the command line.

## Connect

Now that you have a SwitchEngine server running, you must connect your web application to it using a WebSocket.

### JavaScript

```
var websocket = new WebSocket("ws://localhost:6789/");
```

You can define event handlers to be notified when the connection opens, closes, or encounters an error.

### JavaScript

```
websocket.onopen = function() { alert("Connection opened..."); }  
  
websocket.onerror = function (error) { console.log("WebSocket Error: " + error); };  
  
websocket.onclose = function() { alert("Connection closed..."); };
```

Once you have a connection established you can start sending commands to the SwitchEngine, such as:

### JavaScript

```
websocket.send("inspect container");
```

The next section details all of the commands that are available.

## Video

Once you are connected, you will receive video images and related metadata whenever the video changes, such as when playing, stepping, jumping, etc. This could be the result of a command you sent, or as the result of the user interacting with the Switch application. You do not need to explicitly request these messages, and they can not be turned off.

			Sample Video Image Message
	Header		Data
Binary Message Type 8 bits 1 = Video, 2 = Audio	Video Frame Index 32 bits	Padding 24 bits	JPEG Encoded Video Image Variable length
0x01	0x00000001	0x000000	0xFFD8.....0xFF

You can correlate video metadata with video frame images using the **index** property, which is zero-based. The **type** property tells you whether the video frame is an intra-coded picture (I) frame, predicted (P) frame, or bidirectional

### Sample Video Metadata JSON Message

```
{
  "height": 480.0,
  "index": 0,
  "kind": "video",
  "metadata": "\nQuickTime Time Code Track, \nDV/DVCPRO LTC Time Code, \nDV/DVCPRO
CEA-608 Closed Captions, \nCEA-608 Closed Captions (Field 1), \nCEA-608 Closed
Captions (Field 2)",
  "timecodes": {
    "Made by Switch": "00:00:00:00",
    "QuickTime Time Code Track": "01:00:00;00",
    "MPEG-2 GOP Time Code": "00:00:00:00",
  },
  "type": "I",
  "width": 640.0
}
```

The SwitchEngine server supports the following commands.

## Inspection

### Inspect

Use the **inspect** command to get information about the media. You can ask for information about the container, the video track (only one video track is supported per SwitchEngine server instance), each of the audio tracks, or each of the subtitle/caption tracks.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>inspect</b>	container	None	None	JSON	container properties
	video	None	None	JSON	video properties
	audio	<track index>	None	JSON	audio properties
	subtitle	<track index>	None	JSON	subtitle properties

To determine the number of audio or subtitle/caption tracks available, inspect the container. The resulting container properties will include the audio track count and the subtitle track count. The track index parameters are zero-based (the first track is at index 0).

Sample responses for each type of inspection are below.

inspect container

Sample JSON Response

```
{
  "audio track count": 1,
  "creation date": "09/01/15 03:51 PM",
  "directory": "\/Users/telestream/Desktop",
  "estimated duration": "01:01.26",
  "file name": "DVwithCaptions.mov",
  "kind": "container properties",
  "license required": "Switch 4 Player",
  "metadata": {
    "mov.tmc": "Enabled",
    "mpeg2.ftyp.brand": "QuickTime File Format",
    "timecodes": {
      "Made by Switch": "00:00:00:00",
      "QuickTime Time Code Track": "01:00:00;00",
      "MPEG-2 GOP Time Code": "00:00:00:00",
    }
  },
  "modification date": "03/15/06 09:27 AM",
  "path": "\/Users/telestream/Desktop/DVwithCaptions.mov",
  "size": "232.1 MB",
  "subtitle track count": 0,
  "type": "QuickTime Movie",
  "video track count": 1 }
```

```
{
  "bitrate": "28.8 Mbit/s",
  "clean aperture": "0 x 0",
  "color matrix coefficients": "ITU-R BT.601-6",
  "color primaries": "ITU-R BT.601-6 (525)",
  "color transfer characteristics": "ITU-R BT.601-6",
  "colospace": "ITU-R BT.601 NTSC",
  "crop bottom": 0.0,
  "crop height": 512.0,
  "crop left": 0.0,
  "crop right": 0.0,
  "crop top": 0.0,
  "crop width": 720.0,
  "display height": 480.0,
  "display width": 640.0,
  "duration": "01:01.26",
  "duration in scale": 183600,
  "encoded height": 480.0,
  "encoded width": 720.0,
  "format": "DV 25 NTSC",
  "frame count": 1836,
  "frame rate and field order": "29.97 fps, Interlaced (Lower Field First)",
  "kind": "video properties",
  "language": "English",
  "license required": "Switch 4 Player",
  "metadata": {
    "mp4v.compressorname": "DV/DVCPRO - NTSC"
  },
  "pixel aspect ratio": "8:9 Pixels",
  "resize height": 512.0,
  "resize width": 720.0,
  "subsampling": "4:1:1",
  "time scale": 2997
}
```

inspect audio 0

Sample JSON Response

```
{
  "availability": "Available",
  "bitrate": "1.5 Mbit/s",
  "channel coding": "",
  "channel count": 2,
  "channels": [
null,
    {
      "index": 0,
"label": "Unassigned"
    },
    {
      "index": 1,
"label": "Unassigned"
    }
  ],
  "duration": "01:01.26",
  "duration in scale": 2940525,
  "file name": "DVwithCaptions.mov",      "format": "PCM, 16-
Bit Big-Endian Signed Integer - 'twos'",
  "index": 0,
  "kind": "audio properties",
  "label": "A",
  "language": "English",
  "license required": "Switch Demo",
  "program": "Dual Channel",
  "sample rate": "48 kHz",
  "time scale": 48000
}
```

inspect subtitle 0

Sample JSON Response

```

{
  "character set": "Latin",
  "duration": "01:39.64",
  "duration in scale": 2491,      "file name":
"OP47TeletextEmbeddedXDCAMHD.mxf",
  "format": "'ttx't'",
  "index": 0,
  "kind": "subtitle properties",
  "label": "A",
  "language": "Undefined",
  "license required": "Switch 4 Plus",
  "time scale": 25,
  "timecode format": "29nd",
  "type": "Teletext"
}

```

## Audio

### Send

Use the **send** command to request audio data or audio metering data. In both cases, data will continue to be sent to the web application until it asks the server to stop, or the connection is terminated.

Video frame images and metadata are automatically sent whenever a new frame is to be displayed. You do not need to explicitly request them, and they can not be turned off.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>send</b>	audio	None	None	Binary	Audio
	meters	None	None	JSON	audio meters

If you do not request audio data, the SwitchEngine will play audio directly through the client computer's operating system. If you do request audio data, the SwitchEngine will send two channels (mixing up or down to stereo as necessary) of 32-bit, floating-point, little-endian, interleaved Linear PCM at 48kHz. The SwitchEngine will not play audio through the client computer's operating system after the web application requests audio data. The web application can use the **stop audio** command to cease receiving audio data and resume having audio played through the client computer's operating system.

### Sample Video Metadata JSON Message

```

{
  "height": 480.0,
  "index": 0,
  "kind": "video",
  "metadata": "\nQuickTime Time Code Track, \nDV/DVCPRO LTC Time Code, \nDV/DVCPRO
CEA-608 Closed Captions, \nCEA-608 Closed Captions (Field 1), \nCEA-608 Closed
Captions (Field 2)",
  "timecodes": {
    "Made by Switch": "00:00:00:00",
    "QuickTime Time Code Track": "01:00:00:00",
    "MPEG-2 GOP Time Code": "00:00:00:00",
  },
  "type": "I",
  "width": 640.0
}

```

You can correlate video metadata with video frame images using the **index** property, which is zero-based. The **type** property tells you whether the video frame is an intra-coded picture (I) frame, predicted (P) frame, or bidirectional predicted (B) frame.

<b>Sample Video Message</b>			
Header		Data	
Binary Message Type 8 bits 1 = Video, 2 = Audio	Video Frame Index 32 bits	Padding 24 bits	JPEG Encoded Video Image Variable length
0x01	0x00000001	0x000000	0xFFD8.....0xFF
<b>Send audio</b>		<b>Sample Response</b>	
Header		Data	
Binary Message Type 8 bits 1 = Video, 2 = Audio	Audio Chunk Index 32 bits	Padding 24 bits	Audio Chunk Variable length
0x02	0x00000001	0x000000	0xA1.....0xFF

Since JavaScript requires that byte offsets for creating arrays are a multiple of the variable size, the binary message headers are padded out to a total of 8 bytes. This allows you to create Uint8Array or Float32Array using 8 as an offset to skip the header, which avoids a superfluous copy.

Send meters	Sample JSON Response
<pre>{   "channel": [     {       "peak": -21.1202411187721,       "rms": -20.7785511016846     },     {       "peak": -21.4977739577611,       "rms": -20.774845123291     }   ],   "frame index": 87,   "kind": "audio meters",   "track": [     {       "integrated": { "peak": -24.3473339080811, "rms": -72.0       },       "momentary": { "peak": -20.7766990661621, "rms": -72.0       },       "range": {         "peak": 0.0,         "rms": 0.0       },       "short term": { "peak": -24.0495948791504, "rms": -72.0       }     }   ] }</pre>	

The **channel** property contains audio meter values for each channel, in the order that they appear in the file (and I Switch).

The **frame index** allows you to correlate the audio meter values to the corresponding video frame.

The **track** property contains the loudness measurements for each track, in the order that they appear in the file (and in Switch). The loudness measurements are either gated ([ITU-R BS.1770-3](#)) or ungated (ITU-R BS.1770-1) depending on the setting chosen within Switch's Audio->Loudness Type menu.

The **peak** properties contain peak-to-peak measurements in decibels. The **rms** properties contain root mean squared measurements in decibels.

## Navigation

### Stop

Use the **stop** command to cease transmission of audio data or audio metering data, or to stop the SwitchEngine server altogether.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>stop</b>	audio	None	None	No Direct Response	Not Applicable
	meters	None	None	No Direct Response	Not Applicable
	server	None	None	No Direct Response	Not Applicable

The web application can use the **stop audio** command to cease receiving audio data and resume having audio played through the client computer's operating system.

Once you send the stop server command, the connection will be terminated and can not be resumed. You must create a new SwitchEngine server to continue.

### Play

Use the **play** command to initiate forward playback at normal speed. This may cause video frames and related metadata to be sent to the web application.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>play</b>	None	None	None	No Direct Response	Not Applicable

### Pause

Use the **pause** command to stop playback at the current frame.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>pause</b>	None	None	None	No Direct Response	Not Applicable

## Reverse

Use the **reverse**

to be sent to the web application.

peed. This may cause video  
adata

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>reverse</b>	None	None	None	No Direct Response	Not Applicable

## J

Use the **j** command to initiate reverse playback at 1x speed when the video is paused. If the video is already rewinding, sending the **j** command will increase the reverse playback by 1x. If the video is moving forward, sending the **j** command will decrease the playback speed by 1x.

This may cause video frames and related metadata to be sent to the web application.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>j</b>	None	None	None	No Direct Response	Not Applicable

## K

Use the **k** command to stop playback no matter what the current playback speed or direction is. This may cause video frames and related metadata to be sent to the web application.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>k</b>	None	None	None	No Direct Response	Not Applicable

## L

Use the **l** command to initiate forward playback at 1x speed when the video is paused. If the video is already rewinding, sending the **l** command will decrease the reverse playback by 1x. If the video is moving forward, sending the **l** command will increase the playback speed by 1x.

This may cause video frames and related metadata to be sent to the web application.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>I</b>	None	None	None	No Direct Response	Not Applicable

### Slow

Use the **s1ow** command to playback in slow motion (half speed) in either the forward or reverse directions. This may cause video frames and related metadata to be sent to the web application.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>slow</b>	forward	None	None	No Direct Response	Not Applicable
	backward	None	None	No Direct Response	Not Applicable

### Step

Use the **step** command to display the previous or next frame and then pause on that frame. This may cause a video frame and related metadata to be sent to the web application.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>step</b>	forward	None	None	No Direct Response	Not Applicable
	backward	None	None	No Direct Response	Not Applicable

### Skip

Use the **skip** command to jump forward or backward by 1 or 10 seconds. This may cause a video frame and related metadata to be sent to the web application.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>skip</b>	forward	<1 or 10>	None	No Direct Response	Not Applicable
	backward	<1 or 10>	None	No Direct Response	Not Applicable

### Next

Use the **next** command to jump to the start of the next chapter or flag. This may cause a video frame and related metadata to be sent to the web application.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>next</b>	chapter	None	None	No Direct Response	Not Applicable
	flag	None	None	No Direct Response	Not Applicable

### Previous

Use the **previous** command to jump to the start of the previous chapter or flag. This may cause a video frame and related metadata to be sent to the web application.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>previous</b>	chapter	None	None	No Direct Response	Not Applicable
	flag	None	None	No Direct Response	Not Applicable

### Loop

Use the **loop** command to enable or disable looping. When looping is enabled, once the playhead reaches the end of the video, it will jump back to the beginning and continue playing. Use the **loop trim** command to enable or disable looping of just the trimmed segment of the movie.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>loop</b>	on	None	None	No Direct Response	Not Applicable
	off	None	None	No Direct Response	Not Applicable
	trim	on		No Direct Response	Not Applicable
	trim	off		No Direct Response	Not Applicable

### Jump

Use the **jump** command to jump the playhead to another point in the video. This may cause video frames and related metadata to be sent to the web application.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>jump</b>	to frame	<Frame Index>	None	No Direct Response	Not Applicable
	to time	<Time in Seconds>	None	No Direct Response	Not Applicable
	to timecode	<Timecode Source>	<Timecode>	No Direct Response	Not Applicable

When jumping to a frame, the frame index is zero-based.

When jumping to a time, supply a time in fractional seconds, such as "98.23".

When jumping to a timecode, supply a [timecode source four-character code](#) and a timecode that is in the same format (scale, duration, modulus, and drop-frame) as that source's timecode on the first frame. Discontiguous timecode is not supported. You can not jump to a timecode in audio-only files. Files may have multiple timecodes for any given frame. Therefore, you must supply a timecode source four-character code so that Switch know which one to use. If you omit it, the command will fail.

## Closed Captions & Subtitles

### Captions

Use the `captions` command to determine which type of captions are currently selected, if any, or to select a different type of captions to be burned into the video image. You can also turn captions off so that none are burned into the video image. This may cause video frames and related metadata to be sent to the web application.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>captions</b>	cea-608	cc	<1-4>	No Direct Response	Not Applicable
	cea-708	service	<1-6>	No Direct Response	Not Applicable
	teletext	page	<101-999>	No Direct Response	Not Applicable
	settings	None	None	JSON	captions settings
	off	None	None	No Direct Response	Not Applicable

`captions settings`

**Sample JSON Response**

```
{
  "channel": 2,
  "enabled": true,
  "kind": "captions settings",
  "type": "CEA-708"
}
```

## Video Images

### Image

Use the `image` command to request a JPEG encoded video image at a specific index, time, or timecode.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<code>image</code>	at index	<Frame Index>	None	JSON	image
	at time	<Time in Seconds>	None	JSON	image
	at timecode	<Timecode Source>	<Timecode>	JSON	image

The frame index is zero-based.

When using a time, supply one in fractional seconds, such as "98.23".

When using a timecode, supply a [timecode source four-character code](#) and a timecode that is in the same format (scale, duration, modulus, and drop-frame) as that source's timecode on the first frame. Discontiguous timecode is not supported. You can not jump to a timecode in audio-only files. Files may have multiple timecodes for any given frame. Therefore, you must supply a timecode source four-character code so that Switch know which one to use. If you omit it, the command will fail.

The JPEG image is compressed with maximum quality, and then base-64 encoded.

image at index 4.	Sample JSON Response
	<pre>{   "height": 480.0,   "image": "/9j/4AAQSkZJRgABAQAASAB.....AQEBAQEBAQEBAQEBAQ",   "width": 640.0,   "kind": "image",   "index": 4,   "timecode": "00:00:00;04",   "time": 1.33 }</pre>

## Editing

### Video

Use the `video` command to edit the video or revert those changes. This may cause video frames and related metadata to be sent to the web application.

Command	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Response Format	Response Kind
<b>video</b>	crop	<Left>	<Right>	<Top>	<Bottom>	No Direct Response	Not Applicable
	size	<Width>	x	<Height>	None	No Direct Response	Not Applicable
	revert	None	None	None	None	No Direct Response	Not Applicable

Currently, the only edit supported is resizing and/or cropping.

The **size** and **crop** commands correspond to the resize and crop options in the video tab of the Switch inspector. The order the settings appear in the inspector is the order in which they are applied in the video processing pipeline (crop, resize, clean aperture, aspect ratio).

When resizing or cropping is done, a new binary video message will be sent to the web application, even while paused.

## Gutters

Use the **gutters** command to hide or show the area of pixels that have been removed by crop and/or clean aperture. This only affects the display within the Switch app. The video sent to client apps will not be affected by this.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>gutters</b>	on	None	None	No Direct Response	Not Applicable
	off	None	None	No Direct Response	Not Applicable

Turning the gutters on will show the area of pixels that have been removed by crop and/or clean aperture. They will be covered by a semi-transparent white overlay to make them distinct from the pixels that will be kept. Turning the gutters off hides these pixels and the Switch player window will only contain the pixels that will be visible in exports.

## Secondary Files

### Add

Use the **add** command to add audio, subtitles, or flags from secondary files. This may cause video frames and related metadata to be sent to the web application.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>add</b>	audio	<File Path>	None	No Direct Response	Not Applicable
	subtitles	<File Path>	None	No Direct Response	Not Applicable
	flags	<File Path>	None	No Direct Response	Not Applicable

The file path must be an absolute path, not a relative path. It must be in the appropriate format for the current platform (Mac or Windows). On Windows, it must be a UNC path. The path must point to a file that exists and is complete.

## Remove

Use the **remove** command to remove secondary audio, secondary subtitles, or flags. This may cause video frames and related metadata to be sent to the web application.

Command	Parameter 1	Parameter 2	Parameter 3	Response Format	Response Kind
<b>remove</b>	audio	<File Path>	None	No Direct Response	Not Applicable
	subtitles	<File Path>	None	No Direct Response	Not Applicable
	all flags	None	None	No Direct Response	Not Applicable

The file path must be an absolute path, not a relative path. It must be in the appropriate format for the current platform (Mac or Windows). On Windows, it must be a UNC path. The path must point to a file that was previously loaded.

# Supported Timecode Sources

Files may have multiple timecodes for any given frame. Below is a list of the timecode sources supported by Switch. The corresponding four-character codes (a.k.a. Four CC's) can be used to specify the timecode source to use when asking Switch to jump to a timecode, such as via the [timecodesource](#) parameter when launching Switch via command line or hyperlink. Four-character codes are case sensitive.

Timecode Source	Four Character Code
Made by Switch ( <i>Always Available</i> )	swch
TIFO Atom	tfoa
TIFO Timecode	tfot
QuickTime MOV Timecode Track	movt
DV/DVCPRO LTC Timecode	dift
SMPTE ST377 MXF Timecode Track	mxft
SMPTE ST377 MXF Material Package Timecode	mxfm
SMPTE ST377 MXF Source Package Timecode	mxff
SMPTE ST377 MXF System Item Timecode	mxfs
SMPTE ST360 GXF Striped Timecode	gxst
SMPTE ST360 GXF Track Timecode	gxtt
LXF Timecode	lxft
SEI Picture Timing	pic
SEI AVCHD Metadata	MDPM
AVC-Intra SEI Message Type-1	avc1

SMPTE ST328 Elementary Stream Editing Information	esi
---	-----

<b>Timecode Source</b>	<b>Four Character Code</b>
Vertical Blanking Interval	vbi
MPEG-2 GOP Timecode	gopt

# Troubleshooting

## Mac

1. Make sure there is only one copy of Switch on the client computer. If there are multiple copies, you will not be able to predict which one the operating system opens when the user clicks on a `switch://` link.

## Windows

1. If you are having trouble getting the path correct, make sure you are using a UNC path. You can [use drive letters in a UNC path](#) if you are logged in as an administrator and have the [Microsoft Loopback Adaptor](#) installed. Here is an example: `\\localhost\c$\Users\switchqa\Desktop\Tokyo_ID0009.mxf`
2. Do not use the Microsoft Edge browser. It does not work. Use a [supported browser](#) instead.

# Debug Logging

While developing and debugging your web applications, you may find it useful to enable debug logging in Switch. This causes more information to be printed to the console, including errors, warnings, and other information detailing what the application is doing. The SwitchEngine will log each message it receives and each JSON message it sends. It will not log the binary messages it sends, due to their size and frequency.

## Enabling on Mac

1. Quit Switch if it is running.
2. Launch the Terminal application.
3. Enter the following command to enable debug logging: `defaults write net.telestream.switch LogLevel -int 5`
4. Launch the Console application.
5. Launch Switch and watch the new information appear in the Console.

## Enabling on Windows

1. Quit Switch if it is running.
2. Open your preferences file in a text editor. It can be found at: `C:\Users\  
$username\AppData\Roaming\Telestream\Switch\net.telestream.switch.xml`
3. Change the `LogLevel1` value to 5. Close and save.
4. Run Dbgview.exe. You can download from here: <http://technet.microsoft.com/en-us/s.../bb896647.aspx>.
5. Run Switch and view the output in Debugview.

## Log Levels

There are various levels of logging, with zero causing debug logging to be suppressed, and each higher level causing more information to be displayed. Here's a breakdown of each level:

Level	Description
0	Logging disabled. (This is the default.)
1	Only errors are logged.
2	Errors and warnings are logged.
3	Errors, warnings, and additional info are logged.
4	Errors, warnings, additional info, and verbose info are logged.
5	Errors, warnings, additional info, verbose info, and debugging info are logged.

Increasing the debug logging can degrade performance.

To disable debug logging, use the same steps as above, but set the `LogLevel1` to 0.

### Sample Debug Logging Output

```
2017-11-30 10:45:49.715983-0800 Switch[80507:8157350] Engine: Received
WebSocketServer Message: captions settings 2017-11-30 10:45:49.716315-0800
Switch[80507:8147526] Engine: Sending JSON: {
  "channel": 2,
  "enabled": true,      "kind":
"captions settings",  "type":
"CEA-708"
}
```

# Sample Code

You can download a minimal web application (HTML & JavaScript) on request from Telestream.