Osprey®

Video

Osprey® Talon Encoder Configuration and Control API User Guide

Contents

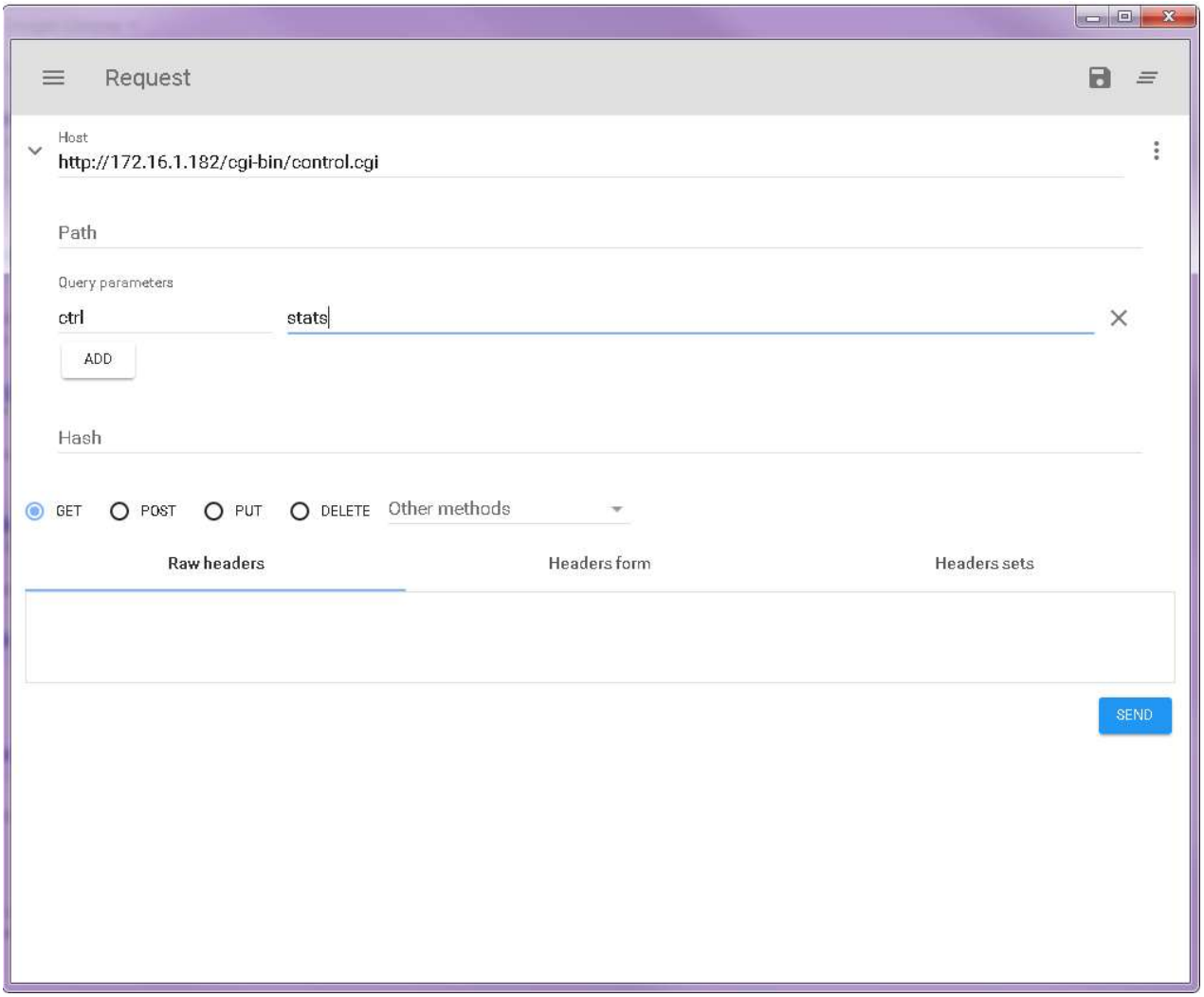
Description

This document describes the Osprey Talon Configuration and Control API (OTCC API). The API uses HTTP PUT/GET requests and values are returned through JSON.

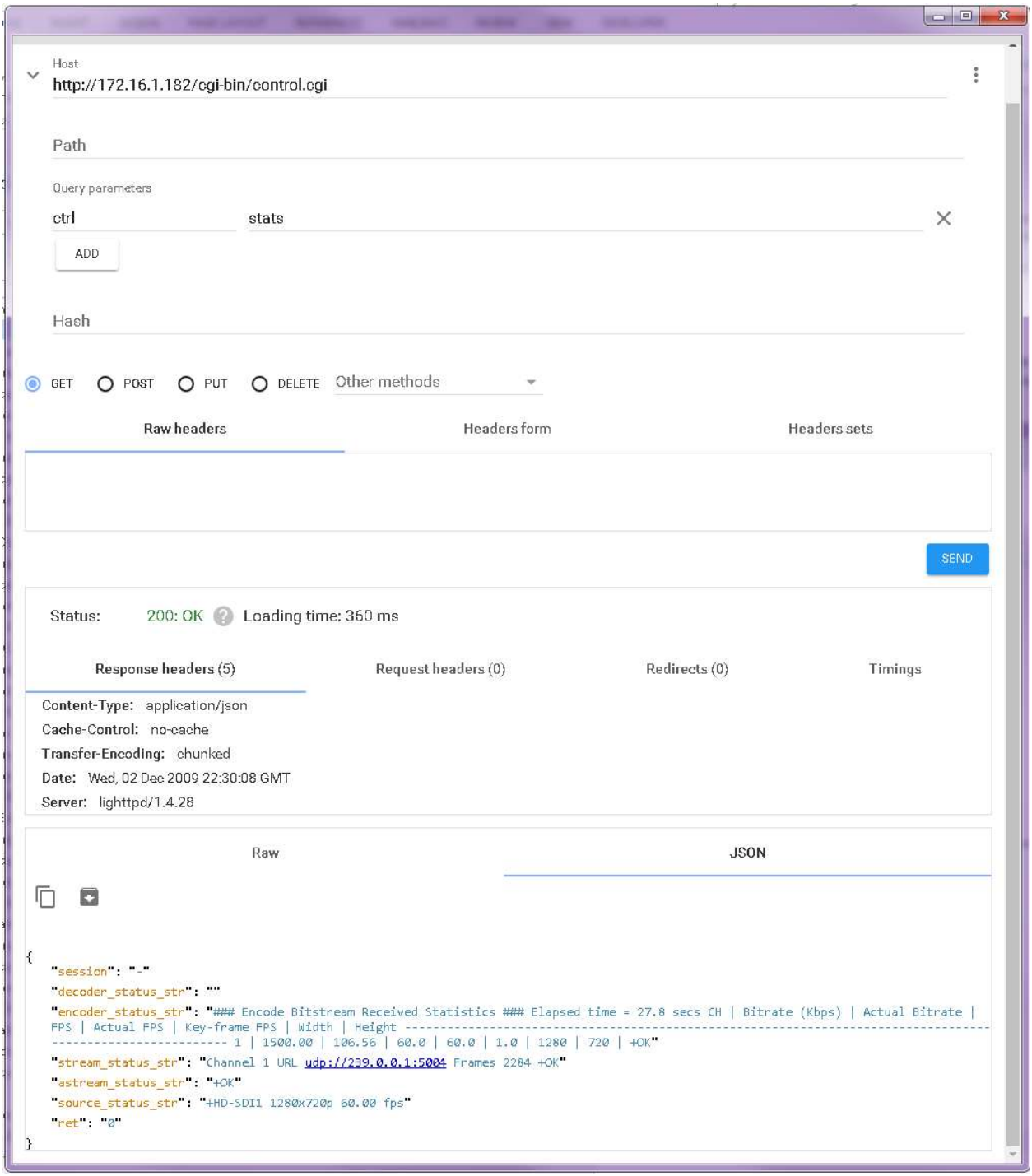
All Osprey Talon Encoders are shipped with the OTCC API pre-installed. This API can be used in C++, C#, Java, IOS, etc.

Testing

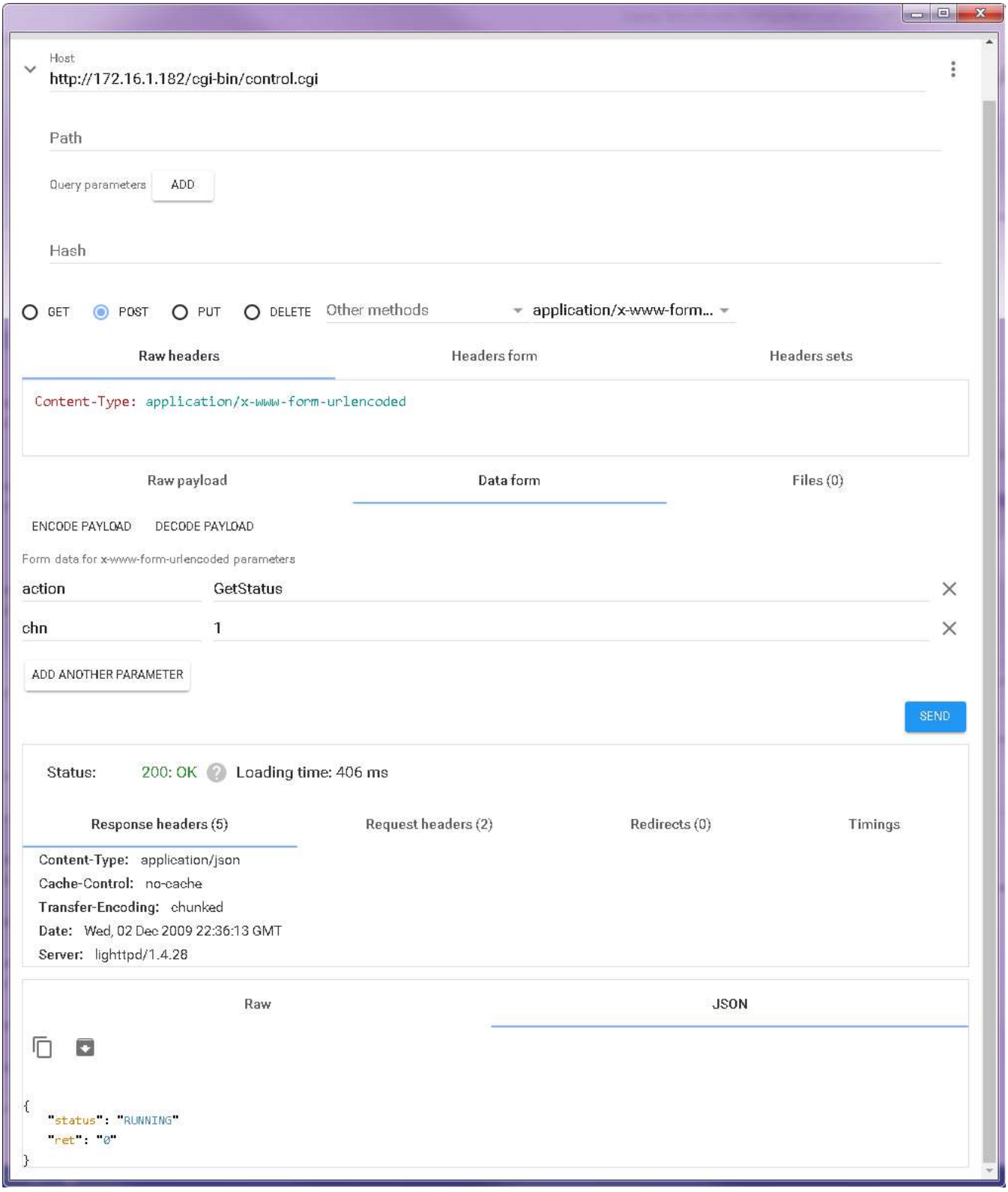
The functions in this document can all be tested outside of a development environment using CURL in Linux or the Advanced REST Client (ARC) that is free for Chrome. An example is provided below for a GET to obtain the stats from the Osprey Talon.



The server response headers and JSON results are all shown in the ARC windows.



Similarly, the ARC can be used to test the POST functions such as the GetStatus function is in this example.



Starting an Encoder

Description

Starts the selected channel on the Talon Encoder.

Method: POST
http://EncoderIP/cgi-bin/control.cgi?action=StartChannel&chn=*{channel}*

Required Parameters

Name	Type	Description	Example
action	*string*	*The start command.*	StartChannel
chn	*string*	*This is a numerical value that represents the channel to start*	1

Response Payload

```
{
"status": "POLLING"
"ret": "0"
}
```

Name	Type	Description	Example
status	*string*	The last status of the encoder. (POLLING, RUNNING, error responses, etc.)	“RUNNING”
ret	*string*	Return Code	“0”

Curl Example

$ curl - -X POST http://10.0.0.1/cgi-bin/control.cgi?action=StartChannel&chn=1

Stopping an Encoder

Description

Stops the selected channel on the Talon Encoder.

Method: POST
http://EncoderIP/cgi-bin/control.cgi?action=StopChannel&chn=*{channel}*

Required Parameters

Name	Type	Description	Example
action	*string*	*The stop command.*	StopChannel
chn	*string*	*This is a numerical value that represents the channel to stop*	1

Response Payload

```
{
"status": "STOPPED"
"ret": "0"
}
```

Name	Type	Description	Example
status	*string*	The last status of the encoder. (POLLING, STOPPED, error responses, etc.)	“STOPPED”
ret	*string*	Return Code	“0”

Curl Example

```
$ curl - -X POST http://10.0.0.1/cgi-bin/control.cgi?action=StopChannel&chn=1
```

Delete Channel

Description

Deletes the selected channel on the Talon Encoder.

Method: POST
http://EncoderIP/cgi-bin/control.cgi?action=DeleteChannel&chn=*{channel}*

Required Parameters

Name	Type	Description	Example
action	*string*	*The delete channel command.*	DeleteChannel
chn	*string*	*This is a numerical value that represents the channel to delete*	3

Response Payload

```
{
"channels": "C1,C2"
"ret": "0"
}
```

Name	Type	Description	Example
channels	*string*	The list of currently configured channels.	“C1,C2”
ret	*string*	Return Code	“0”

Curl Example

$ curl - -X POST http://10.0.0.1/cgi-bin/control.cgi?action=DeleteChannel&chn=3

Add Channel

Description

Creates an additional channel with the channel number specified on the Talon Encoder.

Method: POST
http://EncoderIP/cgi-bin/control.cgi?action=AddChannel&newchn=*{channel}*

Required Parameters

Name	Type	Description	Example
action	*string*	*The add channel command.*	AddChannel
newchn	*string*	*This is a numerical value that represents the channel to add*	1

Response Payload

```
{
"ret": "0"
}
```

Name	Type	Description	Example
ret	*string*	Return Code	“0”

Curl Example

$ curl - -X POST http://10.0.0.1/cgi-bin/control.cgi?action=AddChannel&chn=1

Clone Channel

Description

Clones a selected channel to a specified channel number on the Talon Encoder.

Method: POST
http://EncoderIP/cgi-bin/control.cgi?action=CloneChannel&from=*{channel}*& from=*{channel2}*

Required Parameters

Name	Type	Description	Example
action	*string*	*The clone channel command.*	CloneChannel
from	*string*	*This is a numerical value that represents the channel to clone*	1
to	*string*	*This is a numerical value that represents the channel to create*	2

Response Payload

```
{
"status": "Idle"
"ret": "0"
}
```

Name	Type	Description	Example
status	*string*	The last status of the encoder. (POLLING, STOPPED, error responses, etc.)	"!Error: Bad destination [10.0.0.1]"
ret	*string*	Return Code	"0"

Curl Example

$ curl - -X POST http://10.0.0.1/cgi-bin/control.cgi?action=CloneChannel&chn=1

Get Encoder Status

Description

Returns the encoder status of a specified channel number on the Talon Encoder.

Method: POST
http://EncoderIP/cgi-bin/control.cgi?action=GetStatus&chn=*{channel}*

Required Parameters

Name	Type	Description	Example
action	*string*	*The get status command.*	GetStatus
chn	*string*	*This is a numerical value that represents the channel to stop*	1

Response Payload

```
{
"status": "IDLE"
"ret": "0"
}
```

Name	Type	Description	Example
status	*string*	The last status of the encoder. (POLLING, IDLE, ENCODING)	“ENCODING”
ret	*string*	Return Code	“0”

Curl Example

$ curl - -X POST http://10.0.0.1/cgi-bin/control.cgi?action=GetStatus&chn=1

Get Last Encoder Error

Description

Returns the last error (if there is one) for the specified encoder

Method: POST
http://EncoderIP/cgi-bin/control.cgi?action=GetLastError&chn=*{channel}*

Required Parameters

Name	Type	Description	Example
action	*string*	*The get status command.*	GetStatus
chn	*string*	*This is a numerical value that represents the channel to stop*	1

Response Payload

```
{
"session":"-",
"status": "!Error: Bad destination [192.168.0.6]"
"ret": "0"

}
```

Name	Type	Description	Example
status	*string*	The last error of the encoder. It will be “NONE” or the actual error message.	“"!Error: Bad destination [192.168.0.6]"
ret	*string*	Return Code	“0”

Curl Example

$ curl - -X POST http://10.0.0.1/cgi-bin/control.cgi?action=GetLastError&chn=1

Facebook Live Logout

Description

Logs out of Facebook Live for the specified encoder

Method: POST
http://EncoderIP/cgi-bin/control.cgi?action=FBLogout&chn=*{channel}*

Required Parameters

Name	Type	Description	Example
action	*string*	*The Facebook Live logout command.*	FBLogout
chn	*string*	*This is a numerical value that represents the channel to log out of*	1

Response Payload

```
{
"session":"-",
"ret": "0"

}
```

Name	Type	Description	Example
ret	*string*	Return Code	“0”

Curl Example

$ curl - -X POST http://10.0.0.1/cgi-bin/control.cgi?action=FBLogout&chn=1

Get List of All Configured Streams

Description

Returns the encoder station of a specified channel number on the Talon Encoder.

Method: POST
http://EncoderIP/cgi-bin/control.cgi?action=GetChannels

Required Parameters

Name	Type	Description	Example
action	*string*	*The get channels command.*	GetChannels

Response Payload

```
{
"session": "-"
"channels":"C1,C2",
"ret": "0"
}
```

Name	Type	Description	Example
channels	*string*	A list of configured streams. This could be C1, C2, C3, etc	C1,C2
ret	*string*	Return Code	“0”

Curl Example

$ curl - -X POST http://10.0.0.1/cgi-bin/control.cgi?action=GetChannels

Update Source Status

Description

Updates the internal status of the source connection status. This is to be used in conjunction with the GetStats function on the Talon Encoder

Method: POST
http://EncoderIP/cgi-bin/control.cgi?action=SourceStatus

Required Parameters

Name	Type	Description	Example
action	*string*	*The get update source command.*	SourceStatus

Response Payload

```
{
"ret": "0"
}
```

Name	Type	Description	Example
ret	*string*	Return Code	“0”

Curl Example

$ curl - -X POST http://10.0.0.1/cgi-bin/control.cgi?action=SourceStatus

Update Stream Status

Description

Updates the internal status of the streamstatus. This is to be used in conjunction with the GetStats function on the Talon Encoder

Method: POST
http://EncoderIP/cgi-bin/control.cgi?action=StreamStatus

Required Parameters

Name	Type	Description	Example
action	*string*	*The get update stream status command.*	StreamStatus

Response Payload

```
{
"ret": "0"
}
```

Name	Type	Description	Example
ret	*string*	Return Code	“0”

Curl Example

$ curl - -X POST http://10.0.0.1/cgi-bin/control.cgi?action=StreamStatus

Get Stats

Method: GET
http://EncoderIP/cgi-bin/control.cgi

Required Query Parameters

Name	Type	Description	Example
ctrl	*string*	*The get stats command.*	stats

**This method returns the last known status for each item. To update the status, first call individual methods: EncoderStatus, SourceStatus, AStreamStatus, StreamStatus prior to calling the status function.*

Response Payload


```
{
"session": "-"
"decoder_status_str": ""
"encoder_status_str": "### Encode Bitstream Received Statistics ### Elapsed time = 27.8 secs CH | Bitrate (Kbps) | Actual Bitrate | FPS | Actual FPS | Key-frame FPS | Width | Height ---------------------------------------------------------------------------------------------- 1 | 1500.00 | 106.56 | 60.0 | 60.0 | 1.0 | 1280 | 720 | +OK"
"stream_status_str": "Channel 1 URL udp://239.0.0.1:5004 Frames 2284 +OK"
"astream_status_str": "+OK"
"source_status_str": "+HD-SDI1 1280x720p 60.00 fps"
"ret": "0"
}
```


Name	Type	Description	Example
session	*string*	The last status of the encoder. (POLLING, STOPPED, error responses, etc.)	"-"
decoder_status_str	*string*	Decoder status	N/A
encoder_status_str	*string*	Encoder status for all channels	"### Encode Bitstream Received Statistics ### Elapsed time = 27.8 secs

			CH | Bitrate (Kbps) | Actual Bitrate | FPS | Actual FPS | Key-frame FPS | Width | Height ---------- 1 | 1500.00 | 106.56 | 60.0 | 60.0 | 1.0 | 1280 | 720 | +OK
stream_status_str	*string*	Return Code	"Channel 1 URL udp://239.0.0.1:5004 Frames 2284 +OK"
astream_status_str	*string*	The last status of the encoder. (POLLING, STOPPED, error responses, etc.)	"+OK"
source_status_str	*string*	Return Code	"+HD-SDI1 1280x720p 60.00 fps
ret	*string*	Return Code	“0”

Curl Example

```
$ curl - -X GET http://10.0.0.1/cgi-bin/control.cgi?ctrl=stats
```

Restart Talon

Description

Restarts the Osprey Talon device immediately.

Method: POST
http://EncoderIP/cgi-bin/control.cgi?action=RestartBoard

Required Parameters

Name	Type	Description	Example
action	*string*	*The restart command.*	RestartBoard

Response Payload

```
No response.  System immediately restarts.
```

Curl Example

```
$ curl - -X POST http://10.0.0.1/cgi-bin/control.cgi?action=RestartBoard
```

Change Operational Mode

Description

Allows the operational mode to be changed on the Talon Encoder.

Method: POST
http://EncoderIP/cgi-bin/control.cgi?action=SetOpmode&newopmode=*{mode}*

Required Parameters

Name	Type	Description	Example
action	*string*	*The change operational mode command.*	SetOpmode
newopmode	*string*	*The new operational mode.* *Valid Modes:* 2x2 LC*: enc_int_prog* 1x3 MBR*: enc_1_to_3_ch* 1X1 3G: enc_1ch_noscale	*enc_1_to_3_ch*

Response Payload

```
{
"status": "Idle"
"ret": "0"
}
```

Name	Type	Description	Example
status	*string*	The last status of the encoder. (POLLING, STOPPED, error responses, etc.)	"`!Error: Bad destination [10.0.0.1]`"
ret	*string*	Return Code	"0"

Curl Example

$ curl - -X POST http://10.0.0.1/cgi-bin/control.cgi?action=SetOpmode&newopmode=enc_int_prog

Modifying an Encoder Property

Description

Modifies selected properties on the Osprey Talon Encoder

Method: POST
http://EncoderIP/cgi-bin/control.cgi?*{property}={value}*

Required Parameters

Name	Type	Description	Example
property	*string*	*The property to change formatted as C{#}_PropertyName where "#" is the channel number of the stream to modify (1, 2, 3)* *(Refer to table 1 for property names and values)*	C1_enc_gopsize

Response Payload

```
{
   "ret": "0"
}
```

Name	Type	Description	Example
ret	*string*	Return Code	"0"

```
$ curl - -X POST http://10.0.0.1/cgi-bin/control.cgi?C1_enc_gopsize=25
```

Curl Example

Table 1: Publicly Accessible Encoder Properties

Property Name	Valid Values	Description
enc_autostart	0 1	Used to enable or disable auto start of channel upon system startup.
enc_vidin	HD-SD1 HDMI2 COMPOSITE2	The input source selection for the stream.
enc_vbitrate	Numerical Value	The video bitrate for the stream in kilobits.
enc_frameratediv	1 2 4	Used to encode at less than the input frame rate. For example, an input frame rate of 30 fps with a divider of 2 would produce an output stream of 15 fps.
enc_gopsize	10 12 15 25 30 50 60 90 100 120 200 240 300	The I-frame frequency.
enc_WxH	1920x1080 1280x1024 1280x720 720x480 720x576 1024x576 704x576 640x480 640x360 352x576 352x288 352x240 320x240 176x144	The encode resolution. If this resolution is different from the capture resolution resizing will occur.

enc_vprofile	main high baseline	The H264 profile of the bitstream. Generally High is used for 1080p, Main for D1, and Baseline for less than D1 sizes.
enc_DeiModeId	nodei toponly dei	Deinterlacing modes: *nodei=Off toponly=Deinterlace dei=DeinterlaceHQ*
enc_maxdelayms	0 ms to 5000ms	The number of milliseconds to buffer. The higher the value, the smoother the encode, but also the higher the latency.
enc_interframeinterval	0 1 2 3	The number of B-frames to be used. A value of '0' means B-frames are disabled
enc_audonoff	on off	Enables or disables audio for the encode.
enc_transport	rtmp rtp udp tsfile	Select the output format (container format) for the stream.
enc_ip	rtmp://rtmp.ustream.tv/live/stream1	The output URL for the encoder. If using multicast, this will be the multicast address. This is also the full path to a save when saving to archiving.
enc_port	Numerical Value	The output port to be used for the selected output format.
enc_vpid	0-8191	PID assigned to video in the TS stream.
enc_pcrpid	0-8191	PID assigned to PCR (Program Clock Reference) in the TS stream.
enc_pmtpid	Numerical Value	The PMT (Program Map Table) for the output TS stream.
enc_sharepath	Network share path.	This is the network share path (ie //server/share) This is case sensitive.
enc_domain	Network share domain name	This is the domain name for the network share. This is case sensitive. (optional)

enc_username	Network share username	This is the username for the network share. This is case sensitive.
enc_password	Network share password	This is the password for the network share. This is case sensitive.
enc_rcmode	vbr cbr	Selection of encoding of variable bitrate (VBR) or constant bitrate (CBR)
enc_tsrate	Numerical Value	The rate of the transport stream. Normally this value should be set to 1.25 x (video bitrate + audio bitrate). This can be set to higher values to allow for extra headroom when many bits are needed to encode.
enc_lowlate	on off	Enables or disables the low latency mode for encoding.
enc_auxonoff	on off	Enables or disables the TS File output for UDP streams.
enc_auxfname	String Value	The fully qualified path for the aux TS file to be created without the ".TS" extension.
enc_framealign	on off	Enables or disables frame align for the stream.
enc_force169	on off	For SD inputs only. Forces aspect ratio to 16:9 in player when on.
enc_authonoff	enable disable	Enables or disables RTMP authentication for the stream.
enc_authuser	String Value	The username that will be used during RTMP authentication for the stream.
enc_AuthPasswd	String Value	The password that will be used during RTMP authentication for the stream.
enc_framealign	on off	Enables or disables frame align for the stream.
enc_fbtitle	String Value	The title Facebook Live stream. (Only valid when Facebook Live is being

		used and a user is authenticated)
enc_fbdescription	String Value	The description Facebook Live stream. (Only valid when Facebook Live is being used and a user is authenticated)
enc_fbpictureurl	String Value	Returns the URL for the Facebook Live picture of the logged in user. (Only valid when Facebook Live is being used and a user is authenticated)
enc_fbname	String Value	Returns the name of the authenticated user for the Facebook Live connection
enc_fbstreamtype	REGULAR AMBIENT	REGULAR streams can only stream for the maximum time of 4 hours and are automatically archived. AMBIENT streams can stream 24/7 but are not archived.
enc_audchan{ChannelID}_enable	yes no	Enable or disable the audio for the specified channel.
enc_audchan{ChannelID}_source	none sdi hdmi analog	Selection of what source to use for the selected subchannel of audio.
enc_audchan{ChannelID}_codec	fdk_aaclc fdk_aaclc_mp2 fdk_aache arm_mp12	The codec to use for the encoding of the audio source. *fdk_aaclc=AACLC* *fdk_aaclc_mp2=AACLC MPEG2* *fdk_aache=AACHE* *arm_mp12=MPEG1L2*
enc_audchan{ChannelID}_brate	12000 24000 32000 56000 64000 96000 128000 192000 256000 320000 384000	The bitrate of the audio in the encoding in bits.

enc_audchan{ChannelID}_srate	48000 44100 32000 16000	The sample rate of the audio in the encoding.
enc_audchan{ChannelID}_mode	stereo mono	Allows selection between stereo and mono input.
enc_audchan{ChannelID}_pid	Numerical Value	The PID to use for the audio subchannel. (Valid for UDP only)
enc_audchan{ChannelID}_rtpport	Numerical Value	The port number associated with that audio channel. (Valid for RTP only)
enc_audchan{ChannelID}_ptspcr	Numerical Value	PTS is the presentation timestamp. This is used to achieve synchronization of the separate elementary streams. PCR is the Program Clock Reference.
enc_audchan{ChannelID}_format	adts latm	Select Audio Data Transport Stream (ADTS) or Low Overheard Audio Transport Multiplex formats (LATM).
enc_audchan{ChannelID}_jumbo_pes	0 1 2 3 4	Select the number of AUs for Packetized Elementary Stream (PES). *0=none* *1=1 AUs* *2=2 AUs* *3=3 AUs* *4=4 AUs*

www.ospreyvideo.com