Eos Integration via OSC (as of Eos 2.3 build 192):

This document describes how Eos software exposes integration methods via the Open Sound Control protocol (OSC).

The ***preferred*** method for transmitting and receiving OSC packets is over a ***TCP*** connection. Eos will listen for incoming TCP connections on Port 3032. TCP communication still requires String RX and String TX to be enabled in Show Control Setup. When using TCP, each OSC packet begins with a 4-byte uint32 specifying the size of the OSC content that directly follows. This is part of the OSC specification. (ex: <4-bytes: osc packet size=65 bytes><65-bytes, osc packet data>)

Alternatively, you may transmit and receive packets over UDP, but this method is not preferred because packets may be dropped or delivered out of order. When using a UDP, the appropriate IP address and Ports must be configured in Show Control Setup.

To support both protocols, Eos will keep OSC packets to a maximum size of 512 bytes. When an OSC command requires more than 512 bytes of data, the OSC List convention is used as specific below.

All examples in this document will use the following format:
<OSC Path> = <OSC Argument 1>
<OSC Argument 2>
<OSC Argument 3>
(etc...)

OSC Arguments will be specified one of the following formats:
<OSC Argument Type: Description>
<OSC Argument Type: Example>

Troubleshooting:

In Eos, you may open the Diagnostics tab *(Hold [Tab] and press [9][9])* and then click the appropriate buttons to enable logging of incoming and/or outgoing OSC commands.

To verify basic OSC communication is working, you may send the command **/eos/ping** and Eos will reply with **/eos/out/ping**. You may also add any number of arguments to the command, for example, if you want to measure latency.

Additional OSC Functionality:

Additional OSC functionality not described in this document can be found in the OSC section of the Eos manual. It includes a lot of functionality that may be useful for 3rd party apps to use in conjunction with integration techniques listed here, such as:

- Current command line text — **/eos/out/cmd** = <string: command line text
- Current user id — **/eos/out/user** = <int32: user id>
- Current show title — **/eos/out/show/name** = <string: title>
- Current/Pending cue information — *(see doc)*
- Show Control Events:
  - Cue Fired/Stopped — **/eos/out/event/cue/<cue list number>/<cue number>/fire**
  - Sub Bump On/Off — **/eos/out/event/sub/<sub number>** = <uint32: 0=off, 1=on>
  - Macro Fired — **/eos/out/event/macro/<macro number>**

    - Relay On/Off — **/eos/out/event/relay/<relay number>/<group number>** = <uint32: 0=off, 1=on>
- Console Events:
    - Show Saved — **/eos/out/event/show/saved** = <string: file path>
    - Show Loaded — **/eos/out/event/show/loaded** = <string: file path>
    - Show Cleared — **/eos/out/event/show/cleared**
    - Live/Blind State — **/eos/out/event/state** = <uint32: 0=blind, 1=live>

OSC List Convention:

The OSC List convention is used to send OSC commands that *may* exceed 512 bytes of data.

To add an OSC List of items to an OSC Command, append the OSC Command Path with **/list/<index>/<count>**, where **<index>** is the zero-based index offset into the entire list and **<total>** is the total number of elements in the entire list.

Examples:

OSC List that fits in a single packet:
/eos/out/get/curve/901/list/0/3 = <uint32: 0> <string: 0DF9082C-4A39-40FC-9532-6C3AC01BC6B5> <string: IES Square>

OSC List that spans 2 packets:
/eos/out/get/curve/901/list/0/3 = <uint32: 0> <string: 0DF9082C-4A39-40FC-9532-6C3AC01BC6B5>
/eos/out/get/curve/901/list/2/3 = <string: IES Square>

OSC UID:

UID's uniquely identify each show data target, and are preserved in the show file. This allows you to synchronize with a show file once and then again at a later time, even if changes were made in between.

UID's will be specified as strings in the following format:
XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

Ex:
"B0BAE0A0-3BBE-4004-888B-F61CA125D0B0"

OSC Numbers and Number Ranges:

OSC Arguments that contains numbers or number ranges will be sent as follows:

Eos target numbers will be sent as 32-bit integers when possible. If they are not whole numbers *(ex: Cue 1.23)* then they will be sent as strings.

Ex:
10
"1.23"

When a range numbers contains 2 or more consecutive whole numbers, they will be represented as strings in the following format:
X-Y

Ex:

“1-100”

OSC Gel:

Gels will be represented as strings in the following format:
<Gel Manufacturer Abbreviation><Gel Number>

Ex:

“AP1150” *(Apollo 1150)*
“G101” *(Gam 101)*
“L2” *(Lee 2)*
“R80” *(Rosco 80)*
“SG1” *(Rosco Super Gel 1)*
“E194” *(Rosco E Color 194)*
“T12” *(TokyoBS Poly Color 12)*

Integrating Your App with Eos: Step 1 – Request Eos Software Version

Request the version number of the Eos by sending the following command:
/eos/get/version

Eos will reply with:
/eos/out/get/version = <string: X.X.X.X.X.X>

Ex:
<string: 2.3.0.1.0.111>

(This may be useful if future version of Eos software change the way OSC integration commands are handled)

Integrating Your App with Eos: Step 2 - Synchronize

Request the number of items of a specific type of data you are interested with one of the following commands:
/eos/get/patch/count
/eos/get/cuelist/count
/eos/get/cue/<cue list number>/count
/eos/get/group/count
/eos/get/macro/count
/eos/get/sub/count
/eos/get/preset/count
/eos/get/ip/count *(ip = Intensity Palette)*
/eos/get/fp/count *(fp = Focus Palette)*
/eos/get/cp/count *(cp = Color Palette)*
/eos/get/bp/count *(bp = Beam Palette)*
/eos/get/curve/count
/eos/get/fx/count *(fx = Effect)*
/eos/get/snap/count *(snap = Snapshot)*
/eos/get/pixmap/count
/eos/get/ms/count *(ms = Magic Sheet)*

Eos will reply with the matching command:
/eos/out/get/patch/count = <uint32: count>
/eos/out/get/cuelist/count = <uint32: count>
/eos/out/get/cue/<cue list number>/count = <uint32: count>
/eos/out/get/group/count = <uint32: count>
/eos/out/get/macro/count = <uint32: count>
/eos/out/get/sub/count = <uint32: count>
/eos/out/get/preset /count = <uint32: count>
/eos/out/get/ip/count = <uint32: count>
/eos/out/get/fp/count = <uint32: count>
/eos/out/get/cp/count = <uint32: count>
/eos/out/get/bp/count = <uint32: count>
/eos/out/get/curve/count = <uint32: count>
/eos/out/get/fx/count = <uint32: count>
/eos/out/get/snap/count = <uint32: count>
/eos/out/get/pixmap/count = <uint32: count>
/eos/out/get/ms/count = <uint32: count>

Now you can request detailed information for each item form index 0 to count as follows:
/eos/get/patch/index/<index number>
/eos/get/cuelist/index/<index number>
/eos/get/cue/<cue list number>/index/<index number>
/eos/get/group/index/<index number>
/eos/get/macro/index/<index number>
/eos/get/sub/index/<index number>
/eos/get/preset/index/<index number>
/eos/get/ip/index/<index number>
/eos/get/fp/index/<index number>
/eos/get/cp/index/<index number>
/eos/get/bp/index/<index number>
/eos/get/curve/index/<index number>
/eos/get/fx/index/<index number>
/eos/get/snap/index/<index number>
/eos/get/pixmap/index/<index number>
/eos/get/ms/index/<index number>

Eos will reply with the matching command: *(detailed OSC arguments for each data type listed below)*
/eos/out/get/patch/<channel number>/<part number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...
/eos/out /get/cuelist/<cue list number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...
/eos/out /get/cue/<cue list number>/<cue number>/<cue part number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...
/eos/get/group/<group number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...
/eos/get/macro/<macro number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...
/eos/get/sub/<sub number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...
/eos/get/preset/<preset number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...
/eos/get/ip/<ip number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...
/eos/get/fp/<fp number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...
/eos/get/cp/<cp number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...
/eos/get/bp/<bp number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...
/eos/get/curve/<curve number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...
/eos/get/fx/<fx number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...
/eos/get/snap/<snap number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...

/eos/get/pixmap/<pixmap number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...
/eos/get/ms/<ms number>/list/<list index>/<list count> = <uint32: list index> <string: UID> ...

Integrating Your App with Eos: Step 3 – Staying in Sync

Now your app can request all of the show data from Eos, but if a user is editing show data, your app would become out of sync. The solution to this is to subscribe to Eos show data changes with the following command:
/eos/subscribe = <uint32: X> *(where 0=unsubscribe, 1=subscribe)*

While subscribed, Eos will send the following commands when Eos show data changes:

In the reply, the first argument will be a sequence number, followed by a list of the targets that changed. The targets are specified OSC Numbers and/or OSC Number Ranges *(specified above)*

/eos/out/notify/patch/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/cuelist/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/cue/<cue list number>/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/group/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/macro/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/sub/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/preset/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/ip/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/fp/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/cp/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/bp/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/curve/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/fx/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/snap/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/pixmap/list/<list index>/<list count> = <uint32: sequence number>, ...
/eos/out/notify/ms/list/<list index>/<list count> = <uint32: sequence number>, ...

When your app receives a notification that Eos show data has changed, you should then request detailed information about the modified show data. You may request detailed show data via target number or UID. *(From the initial sync, you should be able to build a mapping of each type of show data to correlate target number with UID)*

Request detailed show data information via target number:
/eos/get/patch/<channel number> *(Eos returns **ALL** parts)*
/eos/get/patch/<channel number>/<part number> *(specific channel part)*
/eos/get/cuelist/<cue list number>
/eos/get/cue/<cue list number>/<cue number> *(Eos returns base cue and **ALL** parts)*
/eos/get/cue/<cue list number>/<cue number>/0 *(base cue)*
/eos/get/cue/<cue list number>/<cue number>/<cue part number> *(specific cue part)*
/eos/get/group/<group number>
/eos/get/macro/<macro number>
/eos/get/sub/<sub number>
/eos/get/preset/<preset number>
/eos/get/ip/<ip number>
/eos/get/fp/<fp number>
/eos/get/cp/<cp number>
/eos/get/bp/<bp number>
/eos/get/curve/<curve number>

/eos/get/fx/<fx number>
/eos/get/snap/<snap number>
/eos/get/pixmap/<pixmap number>
/eos/get/ms/<ms number>

Request detailed show data information via UID:
/eos/get/patch/uid/<UID>
/eos/get/cuelist/uid/<UID>
/eos/get/cue/uid/<UID>
/eos/get/group/uid/<UID>
/eos/get/macro/uid/<UID>
/eos/get/sub/uid/<UID>r>
/eos/get/preset/uid/<UID>
/eos/get/ip/uid/<UID>
/eos/get/fp/uid/<UID>
/eos/get/cp/uid/<UID>
/eos/get/bp/uid/<UID>
/eos/get/curve/uid/<UID>
/eos/get/fx/uid/<UID>
/eos/get/snap/uid/<UID>
/eos/get/pixmap/uid/<UID>
/eos/get/ms/uid/<UID>

Eos will reply with the same command as if the detailed information were requested via index as shown in Step 2.

Integrating Your App with Eos: Step 4 – Modifying Eos Show Data

Your may also modify Eos show data. Typically you should build Eos command lines and send them with the command **/eos/cmd** or **/eos/newcmd**. However, you can also use the following convenience commands for editing the most common show data attributes:

/eos/set/patch/<channel number>/label = <string: text> *(include part number in the path when necessary)*
/eos/set/patch/<channel number>/text1 = <string: text>
/eos/set/patch/<channel number>/text2 = <string: text>
/eos/set/patch/<channel number>/text3 = <string: text>
/eos/set/patch/<channel number>/text4 = <string: text>
/eos/set/patch/<channel number>/text5 = <string: text>
/eos/set/patch/<channel number>/text6 = <string: text>
/eos/set/patch/<channel number>/text7 = <string: text>
/eos/set/patch/<channel number>/text8 = <string: text>
/eos/set/patch/<channel number>/text9 = <string: text>
/eos/set/patch/<channel number>/text10 = <string: text>
/eos/set/patch/<channel number>/notes = <string: text>
/eos/set/patch/<channel number>/gel = <string: text>
/eos/set/cuelist/<cue list number>/label = <string: text>
/eos/set/cue/<cue list number>/<cue number>/label = <string: text> *(base data)*
/eos/set/cue/<cue list number>/<cue number>/<cue part number>/label = <string: text> *(part data)*
/eos/set/group/<group number>/label = <string: text>
/eos/set/macro/<macro number>/label = <string: text>
/eos/set/sub/<sub number>/label = <string: text>
/eos/set/preset/<preset number>/label = <string: text>
/eos/set/ip/<ip number>/label = <string: text>
/eos/set/fp/<fp number>/label = <string: text>

/eos/set/cp/<cp number>/label = <string: text>
/eos/set/bp/<bp number>/label = <string: text>
/eos/set/curve/<curve number>/label = <string: text>
/eos/set/fx/<fx number>/label = <string: text>
/eos/set/snap/<snap number>/label = <string: text>
/eos/set/pixmap/<pixmap number>/label = <string: text>
/eos/set/ms/<ms number>/label = <string: text>

Detailed Information Packet Contents:

NOTE: <uint32: index> is only valid when detailed information is requested via /index (for performance reasons)

PATCH (1 OF 2):
/eos/out/get/patch/<channel number>/<part number>/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<string: label>
<string: fixture manufacturer>
<string: fixture model>
<uint32: address>
<uint32: address of intensity parameter> *(useful for monitoring streaming output to see live levels)*
<uint32: current level>
<string: OSC Gel>
<string: text 1>
<string: text 2>
<string: text 3>
<string: text 4>
<string: text 5>
<string: text 6>
<string: text 7>
<string: text 8>
<string: text 9>
<string: text 10>
<uint32: part count>

Ex:
/eos/out/get/patch/1/1/list/0/20 = 0, “00000000-0000-0000-0000-000000000000”, “My Fixture Label”, “ETC_Fixtures”, “S4_LED_S2_Lustr_Direct”, 1, 1, 0, “R80”, “My_Text_One”, “My_Text_Two”, “My_Text_Three”, “My_Text_Four“, “My_Text_Five”, “My_Text_Six”, “My_Text_Seven”, “My_Text_Eight”, “My_Text_Nine”, “My_Text_Ten”, 1

PATCH (2 OF 2):
/eos/out/get/patch/<channel number>/<part number>/notes =
<uint32: index>
<string: OSC UID>
<string: notes>

Ex:
/eos/out/get/patch/1/1/notes = 0, “00000000-0000-0000-0000-000000000000”, “My Notes”

CUELIST (1 OF 2):
/eos/out /get/cuelist/<cue list number>/list/<list index>/<list count> =
<uint32: index>

<string: OSC UID>
<string: label>
<string: playback mode>
<string: fader mode>
<bool: independent>
<bool: HTP>
<bool: assert>
<bool: block>
<bool: background>
<bool: solo mode>
<uint32: timecode list>
<bool: OOS sync>

Ex:
/eos/out/get/cuelist/1/list/0/13 = 0, "00000000-0000-0000-0000-000000000000", "My Cue List One Label", "Master", "Proportional", True, False, True, False, False, False, 1, False

CUELIST (2 OF 2):
/eos/out /get/cuelist/<cue list number>/links/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<OSC Number Range: linked cue lists list>

Ex:
/eos/out/get/cuelist/1/links/list/0/3 = 0, "00000000-0000-0000-0000-000000000000", 2

CUE (1 OF 4):
/eos/out /get/cue/<cue list number>/<cue number>/<cue part number>/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<string: label>
<uint32: up time duration (ms)>
<uint32: up time delay (ms)>
<uint32: down time duration (ms)>
<uint32: down time delay (ms)>
<uint32: focus time duration (ms)>
<uint32: focus time delay (ms)>
<uint32: color time duration (ms)>
<uint32: color time delay (ms)>
<uint32: beam time duration (ms)>
<uint32: beam time delay (ms)>
<bool: preheat>
<OSC Number: curve>
<uint32: rate>
<string: mark>
<string: block>
<string: assert>
<OSC Number: link> or <string: link> *(string if links to a separate cue list)*
<uint32: follow time (ms)>
<uint32: hang time (ms)>
<bool: all fade>
<uint32: loop>
<bool: solo>
<string: timecode>

<uint32: part count> *(not including base cue, so zero for cues with no parts)*

Ex:
/eos/out/get/cue/1/1/0/list/0/27 = 0, “00000000-0000-0000-0000-000000000000”, “My Cue One Label”, 0, 0, 0, 0, 0, 0, 6000, 0, 0, 0, True, 901, 90, “”, “B”, “A”, 3, 0, 0, True, 1, False, “00:00:00:02”, 0

CUE (2 OF 4):
/eos/out /get/cue/<cue list number>/<cue number>/<cue part number>/fx/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<OSC Number Range: effect list>

Ex:
/eos/out/get/cue/1/1/0/fx/list/0/3 = 0, “00000000-0000-0000-0000-000000000000”, “1-3”

CUE (3 OF 4):
/eos/out /get/cue/<cue list number>/<cue number>/<cue part number>/links/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<OSC Number Range: linked cue lists list>

Ex:
/eos/out/get/cue/1/1/0/links/list/0/3 = 0, “00000000-0000-0000-0000-000000000000”, 2

CUE (4 OF 4):
/eos/out /get/cue/<cue list number>/<cue number>/<cue part number>/actions/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<string: ext link action>

Ex:
/eos/out/get/cue/1/1/0/actions/list/0/3 = 0, “00000000-0000-0000-0000-000000000000”, “Chan 90 At Full”

GROUP (1 OF 2):
/eos/get/group/<group number>/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<string: label>

Ex:
/eos/out/get/group/1.2/list/0/3 = 0, “00000000-0000-0000-0000-000000000000”, “My Group One Point Two Label”

GROUP (2 OF 2):
/eos/get/group/<group number>/channels/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<string: label>

Ex:
/eos/out/get/group/1.2/channels/list/0/5 = 0, “00000000-0000-0000-0000-000000000000”, “1-100”, 200, 300

MACRO (1 OF 2):
/eos/get/macro/<macro number>/list/<list index>/<list count> =

<uint32: index>
<string: OSC UID>
<string: label>
<string: mode>

Ex:
/eos/out/get/macro/1/list/0/4 = 0, “00000000-0000-0000-0000-000000000000”, “My Macro One Label”, “”

MACRO (2 OF 2):
/eos/get/macro/<macro number>/text/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<string: command text> *(split into multiple packets via OSC List convention if necessary)*

Ex:
/eos/out/get/macro/1/text/list/0/3 = 0, “00000000-0000-0000-0000-000000000000”, “Go_To_Cue Out Time 0”

SUB (1 OF 2):
/eos/get/sub/<sub number>/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<string: label>
<string: mode>
<string: fader mode>
<bool: HTP>
<bool: exclusive>
<bool: background>
<bool: restore>
<string: priority>
<string: up time>
<string: dwell time>
<string: down time>

Ex:
/eos/out/get/sub/3/list/0/13 = 0, “00000000-0000-0000-0000-000000000000”, “My Sub Three Label”, “Additive”, “Proportional”, True, False, True, False, “”, “0”, “Man”, “0”

SUB (2 OF 2):
/eos/get/sub/<sub number>/fx/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<OSC Number Range: effect list>

Ex:
/eos/out/get/sub/3/fx/list/0/3 = 0, “00000000-0000-0000-0000-000000000000”, 10

PRESET (1 OF 4):
/eos/get/preset/<preset number>/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<string: label>
<bool: absolute>
<bool: locked>

Ex:
/eos/out/get/preset/10/list/0/5 = 0, “00000000-0000-0000-0000-000000000000”, “My Preset Ten Label”, True, True

PRESET (2 OF 4):
/eos/get/preset/<preset number>/channels/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<OSC Number Range: channel list>

Ex:
/eos/out/get/preset/10/channels/list/0/3 = 0, “00000000-0000-0000-0000-000000000000”, “1-5”

PRESET (3 OF 4):
/eos/get/preset/<preset number>/byType/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<OSC Number Range: by type channel list>

Ex:
/eos/out/get/preset/10/byType/list/0/2 = 0, “00000000-0000-0000-0000-000000000000”

PRESET (4 OF 4):
/eos/get/preset/<preset number>/fx/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<OSC Number Range: effect list>

Ex:
/eos/out/get/preset/10/fx/list/0/0 = 0, “00000000-0000-0000-0000-000000000000”

PALETTE (1 OF 3):
/eos/get/<palette type>/<palette number>/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<string: label>
<bool: absolute>
<bool: locked>

Ex:
/eos/out/get/ip/1/list/0/5 = 0, “00000000-0000-0000-0000-000000000000”, “My IP One Label”, False, False

PALETTE (2 OF 3):
/eos/get/<palette type>/<palette number>/channels/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<OSC Number Range: channel list>

Ex:
/eos/out/get/ip/1/channels/list/0/3 = 0, “00000000-0000-0000-0000-000000000000”, 1-5(s)

PALETTE (3 OF 3):
/eos/get/<palette type>/<palette number>/byType/list/<list index>/<list count> =

<uint32: index>
<string: OSC UID>
<OSC Number Range: by type channel list>

Ex:
/eos/out/get/ip/1/byType/list/0/2 = 0, “00000000-0000-0000-0000-000000000000”

CURVE (1 OF 1):
/eos/get/curve/<curve number>/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<string: label>

Ex:
/eos/out/get/curve/901/list/0/2 = 0, “00000000-0000-0000-0000-000000000000”, “IES Square”

EFFECT (1 OF 1):
/eos/get/fx/<fx number>/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<string: label>
<string: effect type>
<string: entry>
<string: exit>
<string: duration>
<uint32: scale>

Ex:
/eos/out/get/fx/901/list/0/8 = 0, “00000000-0000-0000-0000-000000000000”, “Circle”, “Focus”, “Immediate”, “Immediate”, “Infinite”, 25

SNAPSHOT (1 OF 1):
/eos/get/snap/<snap number>/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<string: label>

Ex:
/eos/out/get/snap/1/list/0/2 = 0, “00000000-0000-0000-0000-000000000000”, “My Snap One Label”

PIXEL MAP (1 OF 2):
/eos/get/pixmap/<pixmap number>/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<string: label>
<uint32: server channel>
<string: interface>
<uint32: width>
<uint32: height>
<uint32: pixel count>
<uint32: fixture count>

Ex:

/eos/out/get/pixmap/1/list/0/9 = 0, “00000000-0000-0000-0000-000000000000”, “My Pixmap One Label”, 100, “sACN”, 32, 32, 1024, 1024

PIXEL MAP (2 OF 2):
/eos/get/pixmap/<pixmap number>/channels/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<OSC Number Range: layer channel list>

Ex:
/eos/out/get/pixmap/1/channels/list/0/3 = 0, “00000000-0000-0000-0000-000000000000”, “101-105”

MAGIC SHEET (1 OF 1):
/eos/get/ms/<ms number>/list/<list index>/<list count> =
<uint32: index>
<string: OSC UID>
<string: label>

Ex:
/eos/out/get/ms/1/list/0/2 = 0, “00000000-0000-0000-0000-000000000000”, “My MS One Label”