December 2024

Developer Information

Blackmagicdesign

HyperDeck Ethernet Protocol

HyperDeck Extreme
HyperDeck Shuttle
HyperDeck Studio

Developer Information

Blackmagic HyperDeck Ethernet Protocol

The Blackmagic HyperDeck Ethernet Protocol is a text based protocol accessed by connecting to TCP port 9993 on HyperDeck models that have a built in Ethernet connection. If you are a software developer, you can use the protocol to construct devices that integrate with our products. Here at Blackmagic Design our approach is to open up our protocols and we eagerly look forward to seeing what you come up with!

You can connect to your HyperDeck recorder using the HyperDeck Ethernet Protocol using a command line program on your computer, such as Terminal on a Mac and putty on a Windows computer.

The HyperDeck Ethernet Protocol lets you schedule playlists and recordings. The following is an example of how to play 7 clips from clip number 5 onwards via the HyperDeck Ethernet Protocol.

On a Mac

1. Open the Terminal application which is located with the applications > utilities folder.
2. Type in “nc” and a space followed by the IP address of your HyperDeck disk recorder, another space and “9993” which is the HyperDeck Ethernet Protocol port number. For example type: nc 192.168.1.154 9993. The Protocol preamble will appear.
3. Type “playrange set: clip id: 5 count: 7” and press ‘return’.

   On HyperDeck disk recorders with a timeline view, you will see in and out points marked around clips 5 through the end of clip 11.
4. Type “play”. Clips 5 through 11 will now play back.
5. To clear the playrange, type “playrange clear”
6. To exit from the protocol, type ‘quit’.

Protocol Commands

Command	Command Description
help or ?	Provides help text on all commands and parameters
commands	return commands in XML format
device info	return device information
disk list	query clip list on active disk
disk list: slot id: {n}	query clip list on disk in slot {n}
disk list: device: {device}	query clip list on disk device USB/network devices can be queried without being active “device” and “slot id” parameters are mutually exclusive in all commands
quit	disconnect ethernet control
ping	check device is responding
preview: enable: {true/false}	switch to preview or output
play	play from current timecode

Command	Command Description
play: speed: {-5000 to 5000}	play at specific speed
play: loop: {true/false}	play in loops or stop-at-end
play: single clip: {true/false}	play current clip or all clips
play: {clip id/clip/timecode/timeline/...}	play from the specified position see “goto” command for description of parameters parameters can be combined with {speed/loop/single clip}
playrange	query playrange setting
playrange set: clip id: {n}	set play range to play clip {n} only
playrange set: clip id: {n} count: {m}	set play range to {m} clips starting from clip {n}
playrange set: in: {inT} out: {outT}	set play range to play between: - timecode {inT} and timecode {outT}
playrange set: timeline in: {in} timeline out: {out}	set play range in units of frames between: - timeline position {in} and position {out}
playrange clear	clear/reset play range setting
play on startup	query unit play on startup state
play on startup: enable: {true/false}	enable or disable play on startup
play on startup: single clip: {true/false}	play single clip or all clips on startup
play option	query play options
play option: stop mode: {lastframe/nextframe/black}	set output frame when playback stops
record	record from current input
record: name: {name}	record named clip
record spill	spill current recording to next slot
record: spill: slot id: {n}	spill current recording to specified slot use current id to spill to same slot
spill order	query the device order used for record spill
stop	stop playback or recording
clips count	query number of clips on timeline
clips get	query all timeline clips
clips get: clip id: {n}	query a timeline clip info
clips get: clip id: {n} count: {m}	query m clips starting from n
clips get: version: {1/2/3}	query clip info using specified output version: version 1: id: name startT duration version 1: id: name startT duration startT depends on “configuration: timecode output: {clip/timeline}” version 2: id: clipInT clipDuration inT outT filename version 3: id: clipInT clipDuration inT outT folder/filename
clips add: name: {name}	append a clip to timeline, name can include subfolders e.g. folder1/HyperDeck_0001.mp4
clips add: clip id: {n} name: {name}	insert clip before existing clip {n}

Command	Command Description
clips add: in: {inT} out: {outT} name: {name}	append the clip portion between clip timecodes {inT} to {outT}
clips add: frame in: {in} frame out: {out} name: {name}	append the clip portion between clip frame numbers {in} to {out}
clips remove: clip id: {n}	remove clip {n} from the timeline (invalidates clip ids following clip {n})
clips clear	empty timeline clip list
clips rebuild	rebuild timeline with default rules
clip info	query clip info for the current playing/recording clip
clip info: clip id: {n}	query clip info for timeline clip id {n}
clip info: name: {name}	query clip info for the clip named {name} on active disk
transport info	query current activity
slot info	query active slot
slot info: slot id: {n}	query slot {n}
slot info: device: {device}	query slot containing device USB/network devices can be queried without being active “device” and “slot id” parameters are mutually exclusive in all commands
slot select: slot id: {n}	switch to specified slot
slot select: device: {device}	switch to slot containing device
slot select: video format: {video format}	load clips of specified video format
slot unblock	unblock active slot
slot unblock: slot id: {n}	unblock slot {n}
slot unblock: device: {device}	unblock disk device
external drive list	list all available USB/network drives for use in external slot
external drive select: device: {device}	switch external slot to specified external drive
external drive selected	query the currently selected external drive
cache info	query cache status
dynamic range	query dynamic range settings
dynamic range: playback override: {off/Rec709/Rec2020_SDR/HLG/ ST2084_300/ST2084_500/ ST2084_800/ST2084_1000/ ST2084_2000/ST2084_4000/ST2084	set playback dynamic range override
dynamic range: record override: {off/Rec709/Rec2020_SDR/HLG/ ST2084_300/ST2084_500/ ST2084_800/ST2084_1000/ ST2084_2000/ST2084_4000/ST2048}	set record dynamic range override
notify	query notification status
notify: remote: {true/false}	set remote notifications
notify: transport: {true/false}	set transport notifications

Command	Command Description
notify: slot: {true/false}	set slot notifications
notify: configuration: {true/false}	set configuration notifications
notify: dropped frames: {true/false}	set dropped frames notifications (reported dropped frame count is approximate)
notify: display timecode: {true/false}	set display timecode notifications
notify: timeline position: {true/false}	set playback timeline position notifications
notify: playrange: {true/false}	set playrange notifications
notify: cache: {true/false}	set cache notifications
notify: dynamic range: {true/false}	set dynamic range settings notifications
notify: slate: {true/false}	set digital slate notifications
notify: clips: {true/false}	set timeline clips notifications where two types of changes can occur: add: partial update with list of clips and insert positions snapshot: complete update of all clips on timeline
notify: disk: {true/false}	set disk clips notifications where two types of changes can occur: add: partial update with list of clips and insert positions snapshot: complete update of all clips on timeline
notify: device info: {true/false}	set device info notifications
notify: nas: {true/false}	set nas notifications triggered by commands such as “nas add” or “nas remove”
goto: clip id: {start/end}	goto first clip or last clip
goto: clip id: {n}	goto clip id {n}
goto: clip id: +{n}	go forward {n} clips
goto: clip id: -{n}	go backward {n} clips
goto: clip: {start/end}	goto start or end of clip
goto: clip: {n}	goto frame position {n} within current clip
goto: clip: +{n}	go forward {n} frames within current clip
goto: clip: -{n}	go backward {n} frames within current clip
goto: timeline: {start/end}	goto start or end of timeline
goto: timeline: {n}	goto frame position {n} within timeline
goto: timeline: +{n}	go forward {n} frames within timeline
goto: timeline: -{n}	go backward {n} frames within timeline
goto: timecode: {timecode}	goto absolute timecode position in timeline
goto: timecode: +{timecode}	go forward {timecode} duration
goto: timecode: -{timecode}	go backward {timecode} duration
goto: slot id: {n}	goto slot id {n} equivalent to “slot select: slot id: {n}”
goto: clip id: {n} clip: {m}	goto clip id {n} and offset to frame position {m} within that clip

Command	Command Description
goto: clip id: {n} timeline: {m}	goto clip id {n} and offset to frame position {m} within the timeline
goto: clip id: {n} timecode: {timecode}	goto clip id {n} and offset {timecode} duration {clip id/clip/timeline/timecode} support absolute and relative offsets use “play” instead of “goto” to play from seeked position
jog: timecode: {timecode}	jog to timecode
jog: timecode: +{timecode}	jog forward {timecode} duration
jog: timecode: -{timecode}	jog backward {timecode} duration
shuttle: speed: {-5000 to 5000}	shuttle with speed
remote	query unit remote control state
remote: enable: {true/false}	enable or disable remote control
remote: override: {true/false}	session override remote control
configuration	query configuration settings
configuration: video input: {SDI/4xSDI/HDMI/component/composite}	change the video input source
configuration: audio input: {embedded/XLR/RCA}	change the audio input source
configuration: file format: {format}	switch to one of the supported formats: H.265High_422, H.264High, H.264Medium, H.264Low, H.264High10_422, H.265High, H.265Medium, H.265Low, QuickTimeProResHQ, QuickTimeProRes, QuickTimeProResLT, QuickTimeProResProxy DNxHR_HQX, QuickTimeDNxHR_HQX, DNxHR_SQ, QuickTimeDNxHR_SQ, DNxHR_LB, QuickTimeDNxHR_LB, DNxHD220x, QuickTimeDNxHD220x, DNxHD145, QuickTimeDNxHD145, DNxHD45, QuickTimeDNxHD45
configuration: audio codec: {PCM/AAC}	switch to specific audio codec
configuration: timecode input: {external/embedded/internal/preset/clip}	change the timecode input
configuration: timecode output: {clip/timeline}	change the timecode output
configuration: timecode preference: {default/dropframe/nondropframe}	whether or not to use drop frame timecodes when not otherwise specified
configuration: timecode preset: {timecode}	set the timecode preset
configuration: audio input channels: {n}	set the number of audio channels recorded to {n}
configuration: record trigger: {none/recordbit/timecoderun}	change the record trigger
configuration: record prefix: {name}	set the record prefix name (supports UTF-8 name)
configuration: record cache: {true/false}	enable or disable record cache, has no effect if cache is not supported/installed/formatted
configuration: append timestamp: {true/false}	append timestamp to recorded filename

Command	Command Description
configuration: usb spill: {true/false}	enable or disable spilling between usb disks
configuration: reference source: {auto/input/external}	set source for the reference signal
configuration: genlock input resync: {true/false}	enable or disable genlock input resync when enabled set reference source to auto/external
configuration: default standard: {video format}	change the default playback video format, see “slot select” for available video formats
configuration: xlr input id: {n} xlr type: {line/mic}	configure xlr input type multiple xlr inputs can be configured in a single command
configuration: xlr mapping: {none/n}	map input XLRs to channels starting from {n} or unmap when {none}
configuration: rca mapping: {none/n}	map input RCAs to channels starting from {n} or unmap when {none} e.g. “xlr mapping: none rca mapping: 9” maps RCA 1 and 2 to channels 9 and 10, and unmaps input XLRs
uptime	return time since last boot
format: slot id: {n} prepare: {exFAT/HFS+} name: {name}	prepare formatting operation filesystem type with volume name {name} “slot id” can be omitted for the current mounted slot “name” defaults to current volume name if mounted (supports UTF-8)
format: device: {device} prepare: {exFAT/HFS+} name: {name}	prepare formatting operation for {device}
format: confirm: {token}	perform a pre-prepared formatting operation using token
identify: enable: {true/false}	identify the device
watchdog: period: {period in seconds}	client connection timeout
reboot	reboot device
slate clips	slate clips information
slate project	slate project information
slate lens	slate lens information
nas list	list all NAS share bookmarks
nas discovered	list all NAS servers that have been discovered via mDNS
nas selected	currently selected NAS share
nas deselect	unmount the currently selected NAS share

Command	Command Description
connection protocol: response version: {version}	changes which do not affect other client connections change the output of “clips get”, “disk list” and related responses version 1 205 clips get id: filename startT duration startT depends on “configuration: timecode output: {clip/timeline}” 519 clips info id: clipInT clipDuration inT outT filename 206 disk list id: filename codec format duration 520 disk list info id: filename codec format duration version 2 205 clips get id: clipInT clipDuration inT outT folder/filename 519 clips info id: clipInT clipDuration inT outT folder/filename 206 disk list id: codec format duration folder/filename 520 disk list info id: codec format duration folder/filename

Multiline only commands:	Command Description
authenticate:↵	authenticate user for secure access
username: {username}	case sensitive username
password: {password}	case sensitive password
slate clips↵	set slate clips information:
reel: {n}	slate reel number, where {n} is in [1, 999]
scene id: {id}	slate scene id value, where {id} is a string
shot type: {WS/MS/CU/BCU/MCU/ECU/ none}	slate shot type
take: {n}	slate take number, where {n} is in [1, 99]
take scenario: {PU/VFX/SER/none}	slate take scenario
take auto inc: {true/false}	slate take auto increment
good take: {true/false}	slate good take
environment: {interior/exterior}	slate environment
day night: {day/night}	slate day or night
slate project:↵	set slate project information:
project name: {name}	project name (can be empty, supports UTF-8)
camera: {index}	set camera index e.g. A
director: {name}	director (can be empty, supports UTF-8)
camera operator: {name}	camera operator (can be empty, supports UTF-8)

Multiline only commands:	Command Description
slate lens:↵	set lens information:
lens type: {type}	lens type (can be empty, supports UTF-8)
iris: {type}	camera iris (can be empty, supports UTF-8)
focal length: {length}	focal length (can be empty, supports UTF-8)
distance: {distance}	lens distance (can be empty, supports UTF-8)
filter: {filter}	lens filter (can be empty, supports UTF-8)
nas add:↵	add a NAS share to the list of bookmarks
url: {url}	URL of the NAS share e.g. smb://server.local/path/to/share
username: {username}	username to connect as (optional, defaults to guest)
password: {password}	password to connect with (optional)
nas remove:↵	remove NAS share bookmark, does not unmount share if mounted
url: {url}	URL of the NAS share e.g. smb://server.local/path/to/share
nas select:↵	mount NAS share asynchronously. Uses credentials provided in matching bookmark, otherwise uses guest credentials
url: {url}	URL of the NAS share e.g. smb://server.local/path/to/share “slot info: device: network” or “notify: slot: true” to determine when share is mounted.

Command Combinations

You can combine the parameters into a single command, for example:

```
play: clip id: 3 speed: 200 loop: true single clip: true
```

Or for configuration:

```
configuration: video input: SDI audio input: XLR
```

Or to switch to the second disk, but only play NTSC clips:

```
slot select: slot id: 2 video format: NTSC
```

Using XML

While you can use the Terminal to talk to HyperDeck, if you are writing software, you can use XML to confirm the existence of a specific command based on the firmware of the HyperDeck you are communicating with. This helps your software user interface adjust to the capabilities of the specific HyperDeck model and software version.

Protocol Details

Connection

The HyperDeck Ethernet server listens on TCP port 9993.

Basic syntax

The HyperDeck protocol is a line oriented text protocol. Lines from the server will be separated by an ascii CR LF sequence. Messages from the client may be separated by LF or CR LF.

New lines are represented in this document as a "↵" symbol.

Single line command syntax

Command parameters are usually optional. A command with no parameters is terminated with a new line:

```
{Command name}↵
```

If parameters are specified, the command name is followed by a colon, then pairs of parameter names and values. Each parameter name is terminated with a colon character:

```
{Command name}: {Parameter}: {Value} {Parameter}: {Value} ...↵
```

Multiline command syntax

The HyperDeck protocol also supports an equivalent multiline syntax where each parameter-value pair is entered on a new line. E.g.

```
{Command name}:↵
{Parameter}: {Value}↵
{Parameter}: {Value}↵
↵
```

Response syntax

Simple responses from the server consist of a three digit response code and descriptive text terminated by a new line:

```
{Response code} {Response text}↵
```

If a response carries parameters, the response text is terminated with a colon, and parameter name and value pairs follow on subsequent lines until a blank line is returned:

```
{Response code} {Response text}:↵
{Parameter}: {Value}↵
{Parameter}: {Value}↵
...
↵
```

Successful response codes

A simple acknowledgement of a command is indicated with a response code of 200:

```
200 ok↵
```

Other successful responses carry parameters and are indicated with response codes in the range of 201 to 299.

Failure response codes

Failure responses to commands are indicated with response codes in the range of 100 to 199:

```
100 syntax error
101 unsupported parameter
102 invalid value
103 unsupported
104 disk full
105 no disk
106 disk error
107 timeline empty
108 internal error
109 out of range
110 no input
111 remote control disabled
112 clip not found
120 connection failed
121 authentication failed
122 authentication required
150 invalid state
151 invalid codec
160 invalid format
161 invalid token
162 format not prepared
163 parameterized single line command not supported
```

Asynchronous response codes

The server may return asynchronous messages at any time. These responses are indicated with response codes in the range of 500 to 599:

```
5xx {Response Text}:↵
{Parameter}: {Value}↵
{Parameter}: {Value}↵
↵
```

Connection response

On connection, an asynchronous message will be delivered:

```
500 connection info:↵
protocol version: {Version}↵
model: {Model Name}↵
↵
```

Connection rejection

A limited number of clients may connect at a time. If too many clients attempt to connect concurrently, they will receive an error and be disconnected:

```
120 connection failed↵
```

Timecode syntax

Timecodes are expressed as non-drop-frame timecode in the format:

```
HH:MM:SS:FF
```

Handling of deck "remote" state

The “remote” command may be used to enable or disable the remote control of the deck. Any attempt to change the deck state over ethernet while remote access is disabled will generate an error:

```
111 remote control disabled↵
```

To enable or disable remote control:

```
remote: enable: {“true”, “false”} ↵
```

The current remote control state may be overridden allowing remote access over ethernet irrespective of the current remote control state:

```
remote: override: {“true”, “false”} ↵
```

The override state is only valid for the currently connected ethernet client and only while the connection remains open.

The “remote” command may be used to query the remote control state of the deck by specifying no parameters:

```
remote↵
```

The deck will return the current remote control state:

```
210 remote info:↵
enabled: {“true”, “false”}↵
override: {“true”, “false”}↵
↵
```

Asynchronous remote control information change notification is disabled by default and may be configured with the “notify” command. When enabled, changes in remote state will generate a “510 remote info:”asynchronous message with the same parameters as the “210 remote info:” message.

Closing connection

The "quit" command instructs the server to cleanly shut down the connection:

```
quit↵
```

Checking connection status

The "ping" command has no function other than to determine if the server is responding:

```
ping↵
```

Getting help

The "help" or "?" commands return human readable help text describing all available commands and parameters:

```
help↵
```

Or:

```
?↵
```

The server will respond with a list of all supported commands:

```
201 help:↵
{Help Text}↵
{Help Text}↵
↵
```

Switching to preview mode

The "preview" command instructs the deck to switch between preview mode and output mode:

```
preview: enable: {"true", "false"}↵
```

Playback will be stopped when the deck is switched to preview mode. Switching to playback is not permitted during record. Use the stop command to stop recording before switching to playback.

Controlling device playback

The “play” command instructs the deck to start playing:

```
play↵
```

The play command accepts a number of parameters which may be used together in most combinations.

By default, the deck will play all remaining clips on the timeline then stop.
The “single clip” parameter may be used to override this behavior:

```
play: single clip: {“true”, “false”}↵
```

By default, the deck will play at normal (100%) speed. An alternate speed may be specified in percentage between -5000 to 5000:

```
play: speed: {% normal speed}↵
```

By default, the deck will stop playing when it reaches to the end of the timeline. The “loop” parameter may be used to override this behavior:

```
play: loop: {“true”, “false”}↵
```

To play from the start of a particular clip:

```
play: clip id: {Clip Id}↵
```

To play from a position offset from the start of particular clip:

```
play: clip id: {Clid Id} timecode: +{timecode}↵
```

The “playrange” command returns the current playrange setting if any:

```
playrange↵
```

To override this behaviour and select a particular clip:

```
playrange set: clip id: {Clip ID}↵
```

To only play a certain number of clips starting at a particular clip:

```
playrange set: clip id: {n} count: {m}↵
```

To only play a certain timecode range:

```
playrange set: in: {in timecode} out: {out timecode}↵
```

To play a certain timeline range:

```
playrange set: timeline in: {in} timeline out: {out}↵
```

To clear a set playrange and return to the default value:

```
playrange clear↵
```

The “play on startup command” instructs the deck on what action to take on startup. By default, the deck will not play. Use the “enable” command to start playback after each power up.

```
play on startup: enable {“true”, “false”}↵
```

By default, the unit will play back all clips on startup. Use the “single clip” command to override.

```
play on startup: single clip: {“true”, “false”}↵
```

The “play option” command queries the output frame for when playback stops:

```
play option↵
```

By default, the deck will display the last frame when playback stops. To override this behaviour, the “stop mode” parameter can be used:

```
play option: stop mode: {“lastframe”, “nextframe”, “black”}↵
```

Stopping deck operation

The "stop" command instructs the deck to stop the current playback or capture:

```
stop↵
```

Changing timeline position

The "goto" command instructs the deck to switch to playback mode and change its position within the timeline.

To go to the start of a specific clip:

```
goto: clip id: {Clip ID}↵
```

To move forward/back {count} clips from the current clip on the current timeline:

```
goto: clip id: +/-{count}↵
```

Note that if the resultant clip id goes beyond the first or last clip on timeline, it will be clamp at the first or last clip.

To go to the start or end of the current clip:

```
goto: clip: {“start”, “end”}↵
```

To go to the start of the first clip or the end of the last clip:

```
goto: timeline: {“start”, “end”}↵
```

To go to a specified timecode:

```
goto: timecode: {timecode}↵
```

To move forward or back a specified duration in timecode:

```
goto: timecode: {“+”, “-”}{duration in timecode}↵
```

To specify between slot 1 and slot 2:

```
goto: slot id: {Slot ID}↵
```

Note that only one parameter/value pair is allowed for each goto command.

Enumerating supported commands and parameters

The "commands" command returns the supported commands:

```
commands↵
```

The command list is returned in a computer readable XML format:

```
212 commands:
<commands>↵
     <command name="…"><parameter name="…"/>…</command>↵
     <command name="…"><parameter name="…"/>…</command>↵
     …
</commands>↵
↵
```

More XML tokens and parameters may be added in later releases.

Controlling asynchronous notifications

The “notify” command may be used to enable or disable asynchronous notifications from the server.
To enable or disable transport notifications:

```
notify: transport: {“true”, “false”}↵
```

To enable or disable slot notifications:

```
notify: slot: {“true”, “false”}↵
```

To enable or disable remote notifications:

```
notify: remote: {“true”, “false”}↵
```

To enable or disable configuration notifications:

```
notify: configuration: {“true”, “false”}↵
```

Multiple parameters may be specified. If no parameters are specified, the server returns the current state of all notifications:

```
209 notify:↵
transport: {“true”, “false”}↵
slot: {“true”, “false”}↵
remote: {“true”, “false”}↵
configuration: {“true”, “false”}↵
dropped frames: {“true”, “false”}↵
display timecode: {“true”, “false”}↵
timeline position: {“true”, “false”}↵
playrange: {“true”, “false”}↵
cache: {“true”, “false”}↵
dynamic range: {“true”, “false”}↵
slate: {“true”, “false”}↵
clips: {“true”, “false”}↵
disk: {“true”, “false”}↵
device info: {“true”, “false”}↵
nas: {“true”, “false”}↵
↵
```

Retrieving device information

The "device info" command returns information about the connected deck device:

```
device info↵
```

The server will respond with:

```
204 device info:↵
protocol version: {Version}↵
model: {Model Name}↵
unique id: {unique alphanumeric identifier}↵
slot count: {number of storage slots}↵
software version: {software version}↵
name: {device name}↵
↵
```

Retrieving slot information

The "slot info" command returns information about a slot. Without parameters, the command returns information for the currently selected slot:

```
slot info↵
```

If a slot id is specified, that slot will be queried:

```
slot info: slot id: {Slot ID}↵
disk list: device: {device}↵
```

The server will respond with slot specific information:

```
202 slot info:↵
slot id: {Slot ID}↵
slot name: {“slot name”}↵
device name: {identifying name for disk device}↵
status: {"empty", "mounting", "error", "mounted"}↵
volume name: {Volume name}↵
recording time: {recording time available in seconds}↵
video format: {disk's default video format}↵
blocked: {“true”, “false”}↵
remaining size: {remaining size in bytes}↵
total size: {total size in bytes}↵
↵
```

A slot can also be specificed by its device. This is particularly useful when there are multiple drives connected via USB. First list the available external drives:

```
external drive list↵
226 external drive info:↵
device: {device}↵
```

Then use slot info with device to query the drive:

```
slot info: device: {device}↵
```

Asynchronous slot information change notification is disabled by default and may be configured with the "notify" command. When enabled, changes in slot state will generate a "502 slot info:" asynchronous message with the same parameters as the "202 slot info:" message.

Retrieving clip information

The “disk list” command returns the information for each playable clip on a given disk. Without parameters, the command returns information for the current active disk:

```
disk list↵
```

If a slot id is specified, the disk in that slot will be queried:

```
disk list: slot id: {Slot ID}↵
```

The server responds with the list of all playable clips on the disk in the format of: Index, name, formats, and duration in timecode:

```
206 disk list:↵
slot id: {Slot ID}↵
{clip index}: {name} {file format} {video format} {Duration timecode}↵
{clip index}: {name} {file format} {video format} {Duration timecode}↵
…
↵
```

Note that the *clip index* starts from 1.

Retrieving clip count

The "clips count" command returns the number of clips on the current timeline:

```
clips count ↵
```

The server responds with the number of clips:

```
214 clips count: ↵
clip count: {Count}↵
```

Retrieving timeline information

The "clips get" command returns information for each available clip on the current timeline. Without parameters, the command returns information for all clips on timeline:

```
clips get↵
```

In version 1, the start timecode reported is either a clip timecode or a timeline timecode depending on the configured output timecode.

The server responds with a list of clip IDs, names and timecodes:

```
205 clips info:↵
clip count: {Count}↵
{Clip ID}: {Name} {Start timecode} {Duration timecode}↵
{Clip ID}: {Name} {Start timecode} {Duration timecode}↵
…
↵
```

The “clips get” command provides a more detailed response when using the “version: 2” parameter:

```
clips get: version: 2↵
```

The server responds with a list of clip IDs, timecodes, in points, out points and names. Clip name is the last field making it simpler to parse when names have embedded spaces.

```
{Clip ID}: {Clip start timecode} {Duration timecode} {inTimecode}
{outTimecode}
{name}↵
{Clip ID}: {Clip start timecode} {Duration timecode}
{inTimecode}
{outTimecode}
{name}↵
…
↵
```

For models that support recursive timelines “clips get: version: 3” replaces the {name} field with {path to clip name} where the {path to clip name} can include directories and subdirectories.

Retrieving transport information

The “transport info” command returns the state of the transport:

```
transport info ↵
```

The server responds with transport specific information:

```
208 transport info:

status: {“preview”, “stopped”, “play”, “forward”, “rewind”,
“jog”, “shuttle”,”record”}↵

speed: {Play speed between -5000 and 5000 %}↵

slot id: {Slot ID or “none”}↵

slot name: {“slot name”}↵

device name: {identifying name for disk device}↵

clip id: {Clip ID or “none”}↵

single clip: {“true”, “false”}↵

display timecode: {timecode}↵

timecode: {timecode}↵

video format: {Video format}↵

loop: {“true”, “false”}↵

timeline: {n}↵

input video format: {Video format”}↵

dynamic range: {“off”, “Rec709”, “Rec2020_SDR”, “HLG”,
“ST2084_300”, “ST2084_500”, “ST2084_800”, “ST2084_1000”,
“ST2084_2000”, “ST2084_4000”, “ST2048” or “none”}↵

reference locked: {“false”, “true”}

↵
```

The "timecode" value is the timecode within the current timeline for playback or the clip for record. The "display timecode" is the timecode displayed on the front of the deck. The two timecodes will differ in some deck modes.

Asynchronous transport information change notification is disabled by default and may be configured with the "notify" command. When enabled, changes in transport state will generate a "508 transport info:" asynchronous message with the same parameters as the "208 transport info:" message.

Video Formats

The following video formats are currently supported on HyperDeck Extreme, HyperDeck Studio and HyperDeck Shuttle:

720p50, 720p5994, 720p60

1080p23976, 1080p24, 1080p25, 1080p2997, 1080p30, 1080p60

1080i50, 1080i5994, 1080i60

HyperDeck Extreme HDR models also support the following formats:

NTSC, PAL, NTSCp, PALp

2160p23.98, 2160p24, 2160p25, 2160p29.97, 2160p30, 2160p50, 2160p59.94, 2160p60

4Kp23976, 4Kp24, 4Kp25, 4Kp2997, 4Kp30

4Kp50, 4Kp5994, 4Kp60

HyperDeckExtreme 8K HDR adds support for the following 8K formats:

4320p23.98, 4320p24, 4320p25, 4320p29.97, 4320p30, 4320p50, 4320p59.94, 4320p60

8Kp23976, 8Kp24, 8Kp25

HyperDeck Studio Pro and Plus models support these 4k formats:

4Kp23976, 4Kp24, 4Kp25, 4Kp2997, 4Kp30

HyperDeck Studio 4K Pro adds support for the following 4k formats:

4Kp50, 4Kp5994, 4Kp60

Video format support may depend on the file format selected and may vary between models and software releases.

File Formats

All HyperDeck models currently support the following file formats:

H.264High_SDI

H.264High

H.264Medium

H.264Low

QuickTimeProResHQ

QuickTimeProRes

QuickTimeProResLT

QuickTimeProResProxy

QuickTimeDNxHD220x

DNxHD220x

QuickTimeDNxHD145

DNxHD145

QuickTimeDNxHD45

DNxHD45

HyperDeck Studio 4K Pro and HyperDeck Extreme HDR models also support the following file formats:

H.265High_SDI

H.265High

H.265Medium

H.265Low

QuickTimeDNxHR_HQX

DNxHR_HQX

QuickTimeDNxHR_SQ

DNxHR_SQ

QuickTimeDNxHR_LB

DNxHR_LB

Supported file formats may vary between models and software releases.

Querying and updating configuration information

The "configuration" command may be used to query the current configuration of the deck:

```
configuration↵
```

The server returns the configuration of the deck:

```
211 configuration:↵
audio input: {“embedded”, “XLR”, “RCA”}↵
audio mapping: {n}↵
video input: {SDI/4xSDI/HDMI/component/composite}↵
file format: {format}↵
audio codec: {“PCM”, “AAC”}↵
timecode input: {“external”, “embedded”, “preset”, “clip”}↵
timecode output: {“clip”, “timeline”}↵
timecode preference: {“default”, “dropframe”, “nondropframe”}↵
timecode preset: {timecode}↵
audio input channels: {n}↵
record trigger: {“none”, “recordbit”, “timecoderun”}↵
record prefix: {name}↵
record cache: {“true”, “false”}↵
append timestamp: {“true”, “false”}↵
reference source: {“auto”, “input”, “external”}↵
genlock input resync: {“true”, “false”}↵
usb spill: {“true”, “false”}↵
default standard: {video format}
xlr mapping: {“none”, “n”}↵
rca mapping: {“none”, “n”}↵
xlr input id: {“n”}↵
xlr type: {“line”, ”mic”}↵
↵
```

Querying and updating configuration information

One or more configuration parameters may be specified to change the configuration of the deck.

To change the current video input:

```
configuration: video input: {“SDI”, “HDMI”, “component”}↵
```

Valid video inputs may vary between models. To configure audio inputs by mapping XLR and RCA audio inputs to output / recording channels, use the “xlr mapping” and “rca mapping” parameters, which supersede the “audio input” and “audio mapping” options. For example, to map RCAs 1 and 2 to channels 9 and 10, and unmap the input XLRs:

```
configuration: xlr mapping: none rca mapping: 9↵
```

To map input XLRs 1, 2, 3, 4 to channels 1, 2, 3, 4 respectively:

```
configuration: xlr mapping: 1↵
```

Note that the supported options here match the available options shown on the deck UI. This setting has no effect on models that lack XLR or RCA inputs.

Valid audio inputs may vary between models.

To configure the current file format:

```
configuration: file format: {File format}↵
```

Note that changes to the file format may require the deck to reset, which will cause the client connection to be closed. In such case, response code 213 will be returned (instead of 200) before the client connection is closed:

```
“213 deck rebooting”
```

Asynchronous configuration information change notification is disabled by default and may be configured with the “notify” command. When enabled, changes in configuration will generate a “511 configuration:” asynchronous message with the same parameters as the “211 configuration:” message.

Selecting active slot and video format

The "slot select" command instructs the deck to switch to a specified slot, or/and to select a specified output video format.

To switch to a specified slot:

```
slot select: slot id: {slot ID}↵
```

To switch to a disk device, including USB drives that are not yet made active:

```
slot select: device: {identifying name for disk device}↵
```

To select the output video format:

```
slot select: video format: {video format}↵
```

Either or all slot select parameters may be specified. Note that selecting video format will result in a rescan of the disk to reconstruct the timeline with all clips of the specified video format.

Clearing the current timeline

The "clips clear" command instructs the deck to empty the current timeline:

```
clips clear↵
```

The server responds with

```
200 ok↵
```

Adding a clip to the current timeline

The "clips add:" command instructs the deck to add a clip to the current timeline:

```
clips add: name: {clip name}↵
```

The server responds with

```
200 ok↵
```

or in case of error

```
1xx {error description}↵
```

Configuring the watchdog

The “watchdog” command instructs the deck to monitor the connected client and terminate the connection if the client is inactive for at least a specified period of time.

To configure the watchdog:

```
watchdog: period: {period in seconds}↵
```

To avoid disconnection, the client must send a command to the server at least every {period} seconds. Note that if the period is set to 0 or less than 0, connection monitoring will be disabled.

Network Area Storage

On networks using multicast DNS the “nas discovered” command will list network servers the HyperDeck has discovered:

```
nas discovered↵
225 nas host info:
CloudStoreMini.local. CloudStoreMini
CloudStore80.local. CloudStore80
CloudStore320.local. CloudStore320
```

A network share can be added as a bookmark to the HyperDeck using ‘nas add’

```
nas add:
url: smb://CloudStore80.local/Studio1
```

For shares that require a username and password consider using the secure mode of the HyperDeck Ethernet protocol to avoid passwords being sent as plaintext.

```
nas add:
url: smb://192.168.1.1/Main
username: user1234
password: Password1234
```

A share can be made available for recording and playback using ‘nas select’. If a bookmark exists for that share, ‘nas select’ will use the credentials stored in the bookmark. Otherwise ‘nas select’ will connect using Guest credentials.

```
nas select:
url: smb://192.168.1.1/Main
```

Only one share can be mounted at a time using ‘nas select’.

After using “nas select” you can determine when the share is available for recording and playback with “slot info: device: network” and inspecting the “url” field. If ‘notify: slot: true’ was used, an asynchronous notification will be sent when the share is mounted.

You can query the currently selected nas share using the ‘nas selected’ command. If the remote server of the selected nas share is not available or has not yet responded, the share will not be mounted and “slot info” will indicate it is not available for recording and playback.