

1. What this is

The BEYOND HTTP Server is a REST gateway built into Pangolin BEYOND. It is available only in BEYOND Ultimate. Other editions do not include it and cannot add it as an option. It exposes the live show engine — workspace, cue grid, playlist, live-control surfaces, projection-zone flags, and PangoScript — to any HTTP client on the LAN. There is no cloud service: the API is the BEYOND instance on that machine.

- Base URL: `http://<beyond-host>:8088/apiv1/` (use the port configured in Settings).
- Default HTTP port 8088. Video preview is a separate process on 8087.
- Plain HTTP only. Put a reverse proxy in front if you need TLS.
- REST calls do not require credentials in current builds.
- Live Try-it-out is the API Playground on the show machine:
`http://<beyond-host>:8088/beyondswagger/`
- This wiki page is documentation. It does not send live requests.

2. Enable the server and open the examples

The BEYOND Ultimate 5.8 installer places the example pages and API Playground in `<BEYOND.exe folder>\www`. Leave Root blank to serve that folder.

1. Open BEYOND Ultimate. Settings → HTTP Server settings.
2. Enable HTTP server, set the port (default 8088), click Apply.
3. Leave Static file server → Root blank. Blank Root is `<BEYOND.exe folder>\www` (stored as `HttpStatDir` in `BEYOND.ini` if you later pick a folder).

Layout	What to do	URL
Default (www)	Leave Root blank. Open the pages the installer already placed in <code><exe>\www</code> .	<code>http://<host>:<port>/index.html</code>
Custom folder	Optional: click ... and choose a folder that contains <code>index.html</code> .	<code>http://<host>:<port>/index.html</code>

API Playground: `http://<host>:<port>/beyondswagger/`

Smoke-test:

```
GET http://127.0.0.1:8088/apiv1/ws/info
```

A 200 response is JSON describing the current workspace.

3. Request and response format

- REST paths live under `/apiv1/`. Future breaking changes will use `/apiv2/`.
- JSON endpoints return `application/json` unless documented otherwise (thumbnails return images).
- PangoScript to `POST /apiv1/ps/execscript` is raw UTF-8 text (Content-Type: `text/plain`). JSON

wrappers such as {"code": "..."} are not supported.

- Treat any non-2xx status as failure.
- execscript does not return WriteLn / echo text.

First script:

```
POST http://127.0.0.1:8088/apiv1/ps/execscript
Content-Type: text/plain
```

```
Blackout
Sleep 500
EnableLaserOutput
```

```
curl -X POST http://127.0.0.1:8088/apiv1/ps/execscript \
-H "Content-Type: text/plain" \
--data-binary $'Blackout\nSleep 500\nEnableLaserOutput'
```

4. Core concepts

PangoScript objects

Most automation flows through PangoScript. Use these names with GET /apiv1/psobject?name=...

Name	Purpose
master	Master Live Control + transition / metro / BPM state
playlist	Playback state mirror
cue.<page>.<index>	One cue by page and index
zone	Zone list (names, preview, uGC, flags). projectorIndex is 1-based.
zone.<n>	Live Control subset for one zone — not the full list-item record

Resource groups

- Workspace — page/cue grid (/apiv1/ws/info, /apiv1/ws/page)
- Thumbnails — cue and FX raster previews
- PangoScript — run commands and scripts; read/write documented objects
- Live Control — master, cue, and zone surfaces
- Zones — flag document on /apiv1/zone. UGC and preview writes use PangoScript.
- Playlist — transport control

Typical integration flow

- Discover — read workspace endpoints.
- Command — trigger cues or run scripts over REST.

- Poll — read live-control and playlist state (about once a second is enough).
- Host UI — serve custom HTML from the static Root, or run an external app on the LAN.

5. Guides

Build a cue grid UI

- GET /apiv1/ws/info — categories and page list (or /apiv1/ws/info2 for stable property names).
- GET /apiv1/ws/page?index=0 — cue names and empty slots.
- GET /apiv1/cuethumb?page=0&cue=0&size=150 — thumbnail for each slot. Add &time=2000 to render at 2 s.
- POST /apiv1/ps/execscript with Content-Type: text/plain and body CueDown 0,3 (or StartCue). One-line form: GET /apiv1/ps/cmd/cuedown%200,3 (%20 is a space; + is literal).

Control playlist playback

- GET /apiv1/playlist/list — available shows.
- GET /apiv1/playlist/play — start the current track. Use /apiv1/playlist/settrack?index=0 to jump first.
- GET /apiv1/playlist/state — position and transport. Poll this for a now-playing UI.
- GET /apiv1/playlist/setposition?time=120000 — seek in milliseconds.

Adjust live parameters

Read the master surface:

```
GET /apiv1/lc/master
```

Partial JSON write:

```
POST /apiv1/lc/master
Content-Type: application/json

{"scanrate": 80, "hue": 30}
```

The same pattern applies to /apiv1/lc/cue (per cue) and /apiv1/lc/zone?index=0 (per zone).

Read or adjust a projection zone

- GET /apiv1/psobject?name=zone — list (names, preview, uGC, 1-based projectorIndex).
- GET /apiv1/zone?index=0 — flags only (mute, blockzone, 0-based projectorindex, visualizationid, testframe, ...). zone.0 is Live Control, not the full list item.

- POST /apiv1/zone?index=0 with {"visualizationid":0,"testframe":0}. ugc, mesh, name, and preview on this route return 400.
- Write UGC or preview with POST /apiv1/ps/execscript, for example: Zone.0.UGC.SizeX = 99
- Live parameters: GET/POST /apiv1/lc/zone?index=0.

6. Errors and script behavior

Code	Meaning
200	Success. Script calls return JSON with totalLines / executedLines / partial. No WriteLn text.
400	Validation or script error. Treat as failure.
404	Path is not under /apiv1/ and not in the static file root.
5xx	Internal BEYOND error. Check Quick Log on the show machine.

Script error body (1-based line):

```
{
  "error": "Line: 2, Error: Unknown identifier",
  "code": 400,
  "line": 2,
  "totalLines": 2,
  "executedLines": 1,
  "partial": true
}
```

Field validation example:

```
{
  "error": "Unknown or unsupported field",
  "field": "ugc"
}
```

- The whole execscript body is executed. There is no silent line cap.
- Execution stops at the first error. Earlier lines stay applied (partial: true).
- Only the first assignment on a semicolon-joined line runs.
- Out-of-range preview values may be stored and read back unclamped.
- GET /apiv1/ps/cmd/{command} is one statement. Query strings are stripped. %20 is a space. A literal + is not a space.

7. How the API behaves

- Read playlist or Live Control over HTTP whenever you need a fresh value.
- execscript returns a JSON result (status, line counts, partial). It does not return WriteLn / console text.
- /apiv1/zone is the zone flag document (mute, blockzone, projectorindex, and the other flag fields). Change UGC or preview geometry with execscript, for example Zone.0.UGC.SizeX = 99.

- psubject name=zone is the zone list. zone.<n> is that zone's Live Control properties.
- projectorindex on /apiv1/zone is 0-based. projectorIndex on the zone list is 1-based.

8. FAQ

Can I try API requests on this wiki page?

No. Use the API Playground on the BEYOND machine at <http://<beyond-host>:8088/beyondswagger/>.

Do REST calls require authentication?

Not in current builds.

Is HTTPS supported?

Not natively. Place a reverse proxy in front of BEYOND if you need TLS.

Why does my tablet get connection refused?

Use the BEYOND PC LAN IP, not 127.0.0.1. Allow inbound TCP on the HTTP port in Windows Firewall.

What Content-Type for scripts?

text/plain to POST /apiv1/ps/execscript. Do not send {"code":"..."}

Where is the FX thumbnail endpoint?

GET /apiv1/fxthumb.

Are there rate limits?

None documented. Poll about once a second for UI feedback.

Which license includes the HTTP server?

BEYOND Ultimate only. See pangolin.com/collections/beyond-software.

How do I list or edit zones?

List: GET /apiv1/psobject?name=zone. Flags: GET/POST /apiv1/zone?index=0. UGC/preview: execscript.

Where is the machine-readable spec?

openapi.json in <exe>\www\beyonddswagger, also attached beside this manual.

9. Endpoint reference

Every public REST path in OpenAPI 1.0. Replace <beyond-host> and the port with your show machine.

Method	Path	Summary
GET	/apiv1/ws/info	Workspace info - categories as keys
GET	/apiv1/ws/info2	Workspace info - fixed property names
GET	/apiv1/ws/page	Get one workspace page (info + cue array)
GET	/apiv1/cuethumb	Cue thumbnail (raster image)
GET	/apiv1/ps/cmd/{command}	Run a single PangoScript command (URI form)
POST	/apiv1/ps/execscript	Run a multi-line PangoScript
GET	/apiv1/psobject	Read a PangoScript object as JSON
POST	/apiv1/psobject	Write properties of a PangoScript object
GET	/apiv1/lc/master	Read Master Live Control
POST	/apiv1/lc/master	Write Master Live Control
GET	/apiv1/lc/cue	Read Cue Live Control
POST	/apiv1/lc/cue	Write Cue Live Control
GET	/apiv1/lc/zone	Read Projection Zone Live Control
POST	/apiv1/lc/zone	Write Projection Zone Live Control
GET	/apiv1/zone	Read projection zone flags
POST	/apiv1/zone	Write projection zone flags
GET	/apiv1/playlist/state	Get current playback state
GET	/apiv1/playlist/list	Get the show list and per-show metadata
GET	/apiv1/playlist/thumb	Get a playlist item thumbnail (raster image)
GET	/apiv1/playlist/play	Play
GET	/apiv1/playlist/stop	Stop
GET	/apiv1/playlist/first	First
GET	/apiv1/playlist/last	Last
GET	/apiv1/playlist/next	Next
GET	/apiv1/playlist/prev	Prev
GET	/apiv1/playlist/settrack	Jump to a specific track
GET	/apiv1/playlist/setposition	Seek within the current show
GET	/apiv1/fxthumb	FX-table thumbnail (raster image)

Workspace

GET /apiv1/ws/info

Workspace info – categories as keys

Returns the workspace structure: a grid block describing the page matrix, plus a category map whose keys are the user-defined category names in BEYOND (e.g. Animals, Boxes). The All key is always present and lists every page regardless of category.

Use this when you want a category-driven UI like the Cue Grid demo. If you need stable property names instead of category-name keys, use /apiv1/ws/info2.

Responses: 200

GET /apiv1/ws/info2

Workspace info – fixed property names

Same payload as /apiv1/ws/info but with stable property names instead of category names as keys. Easier to type-check; harder to drive a category-named UI.

Responses: 200

GET /apiv1/ws/page

Get one workspace page (info + cue array)

Returns the metadata for a single page (pageName, pageCategory) plus an array of all cues on that page. Cues are 0-indexed up to the workspace's colCount * rowCount (typically 64 = 8×8). The empty flag distinguishes blank slots from real cues so you can render the grid without flicker.

The page index is the linear page number across the whole workspace, not a per-category index. Get it from /apiv1/ws/info.

Parameters: index (query, required)

Responses: 200

Thumbnails

GET /apiv1/cuethumb

Cue thumbnail (raster image)

Returns a raster image preview of one workspace cue. Use this to populate cue grids without instantiating the cue. The server picks the closest pre-rendered size, then resamples; very small size values render fastest.

Tip: put `&time=2000` to render the cue at 2 s of its timeline rather than frame zero – useful for animated cues.

Parameters: page (query, required); cue (query, required); format (query, optional); size (query, optional); time (query, optional); backcolor (query, optional)

Responses: 200, 400

GET /apiv1/playlist/thumb

Get a playlist item thumbnail (raster image)

Returns a thumbnail for one show in the playlist. Unlike `/cuethumb` and `/fxthumb`, playlist thumbnails are user-supplied raster images baked into the show file at fixed sizes (see `thumb[ ]` in `/apiv1/playlist/list`). The server can't synthesize an exact size on demand – it picks the nearest pre-baked thumbnail to your requested size.

Provide either size or image:

- size – server returns the closest pre-baked size.
- image – server returns the Nth image stored in the show.

Parameters: index (query, required); size (query, optional); name (query, optional); image (query, optional)

Responses: 200

GET /apiv1/fxthumb

FX-table thumbnail (raster image)

Returns a raster preview of one effect from the FX table. Same parameter contract as `/apiv1/cuethumb` but indexed by `line` (0..7) and `index` (0..100) instead of `page/cue`.

On current BEYOND HTTP this path is `/apiv1/fxthumb`. The older root path `/fxthumb` returns 404.

Parameters: line (query, required); index (query, required); format (query, optional); size (query, optional); time (query, optional); backcolor (query, optional)

Responses: 200

PangoScript

GET /apiv1/ps/cmd/{command}

Run a single PangoScript command (URI form)

Everything after /cmd/ is one PangoScript statement.

Query parameters are stripped and never become script. The path is percent-decoded once (%20 is a space). A literal + is not a space.

For multi-line scripts, or any statement with spaces, use POST /apiv1/ps/execscript with Content-Type: text/plain.

Examples

- GET /apiv1/ps/cmd/blackout
- GET /apiv1/ps/cmd/master.brightness=75
- GET /apiv1/ps/cmd/cuedown%200,3
- GET /apiv1/ps/cmd/Zone.0.Preview.PositionX%20=%200
- GET /apiv1/ps/cmd/cuedown+0,3 — works because +0 is the number zero, not a space

Parameters: command (path, required)

Responses: 200, 400

POST /apiv1/ps/execscript

Run a multi-line PangoScript

Body is UTF-8 PangoScript source (Content-Type: text/plain). Newlines separate statements. JSON wrappers such as {"code": "..."} are not supported.

The whole body is executed. There is no silent line cap. Execution stops at the first error; earlier lines stay applied (partial: true). Only the first assignment on a ;-joined line runs. Out-of-range preview values may be stored unclamped.

WriteLn / echo text is not returned. Treat any non-2xx status as failure.

Request body: text/plain script

Responses: 200, 400

```
Master.Brightness=75
Master.Hue=20
```

GET /apiv1/psobject

Read a PangoScript object as JSON

Returns the live JSON state of a documented object: master, playlist, cue.<page>.<index>, zone (list), or zone.<n> (Live Control subset).

zone list items include name, preview, uGC, and flags. projectorIndex on the list is 1-based. Top-level count appears once. zone.<n> is not the same shape as a list item.

Parameters: name (query, required)

Responses: 200, 400, 404

POST /apiv1/psobject

Write properties of a PangoScript object

Partial JSON update of a documented writable object. zone.<n> accepts Live Control fields (for example posX). name and preview on zone.<n> return 400. Unknown properties return 400.

Parameters: name (query, required)

Request body: application/json

Responses: 200, 400, 404

```
{
  "BPM": 240
}
```

Live Control

GET /apiv1/lc/master

Read Master Live Control

Returns the current state of the Master Live Control surface – the global brightness/hue/scanrate/etc. layer that overlays everything. Equivalent to reading master via /apiv1/psobject but returns only the Live Control subset of properties.

For a per-cue Live Control, see /apiv1/lc/cue. For a per-zone Live Control, see /apiv1/lc/zone.

Responses: 200

POST /apiv1/lc/master

Write Master Live Control

Send a partial Live Control document; only the keys you include are mutated. To reset a property to default, omit it (POSTing null does not reset).

Request body: application/json

Responses: 200, 400

```
{
  "scanrate": 80,
  "hue": 30
}
```

GET /apiv1/lc/cue

Read Cue Live Control

Per-cue Live Control. The Live Control values on a cue are the *current* values, not the design-time defaults – they reflect any tweaks made while the cue is playing.

Parameters: page (query, required); index (query, required)

Responses: 200

POST /apiv1/lc/cue

Write Cue Live Control

Parameters: page (query, required); index (query, required)

Request body: application/json

Responses: 200

GET /apiv1/lc/zone

Read Projection Zone Live Control

Per-zone Live Control surface. Same property set as /apiv1/lc/master but applied to one projection zone.

Parameters: index (query, required)

Responses: 200

POST /apiv1/lc/zone

Write Projection Zone Live Control

Parameters: index (query, required)

Request body: application/json

Responses: 200

Zones

GET /apiv1/zone

Read projection zone flags

Returns the flag document for one zone: mute, blockzone, projectorindex, active, visualizationid, ratiox, ratioy, testframe, visible.

projectorindex is 0-based. This route does not return ugc, mesh, name, or preview. For the full list record (including ugc and preview), use GET /apiv1/pobject?name=zone.

Parameters: index (query, required)

Responses: 200

POST /apiv1/zone

Write projection zone flags

Send a partial flag document. Supported writes include visualizationid, testframe, and projectorindex (0-based integer in engine range).

name, preview, ugc, and mesh are not writable here — they return 400 (Unknown or unsupported field) and the zone is left unchanged. Non-JSON bodies and unknown fields also return 400 before any write.

To write UGC or preview geometry, use POST /apiv1/ps/execscript.

Parameters: index (query, required)

Request body: application/json

Responses: 200, 400

```
{  
  "visualizationid": 0,  
  "testframe": 0  
}
```

Playlist

GET /apiv1/playlist/state

Get current playback state

Cheap snapshot of the playlist transport: whether it's playing, the position in milliseconds, the duration in milliseconds, and the current track index. Most other Playlist endpoints also return this object so you can update your UI from a single response shape.

Responses: 200

GET /apiv1/playlist/list

Get the show list and per-show metadata

Returns the complete playlist plus per-show metadata: name, comment, file paths, audio file, duration, timecode, and a list of pre-rendered thumbnail sizes. Use it to populate a playlist UI.

Responses: 200

GET /apiv1/playlist/play

Play

Start playback of the current track. Returns the updated Playback Status.

Responses: 200

GET /apiv1/playlist/stop

Stop

Stop playback. Returns the updated Playback Status.

Responses: 200

GET /apiv1/playlist/first

First

Jump to the first track. Returns the updated Playback Status.

Responses: 200

GET /apiv1/playlist/last

Last

Jump to the last track. Returns the updated Playback Status.

Responses: 200

GET /apiv1/playlist/next

Next

Advance to the next track. Returns the updated Playback Status.

Responses: 200

GET /apiv1/playlist/prev

Prev

Step back to the previous track. Returns the updated Playback Status.

Responses: 200

GET /apiv1/playlist/settrack

Jump to a specific track

Selects a track by zero-based index. Equivalent to clicking a row in the playlist UI. Returns the updated Playback Status.

Parameters: index (query, required)

Responses: 200

GET /apiv1/playlist/setposition

Seek within the current show

Sets the playback head, in milliseconds from the start of the current show. Use the `infoDurationMS` field from `/apiv1/playlist/list` for the upper bound. Returns the updated Playback Status.

Parameters: time (query, required)

Responses: 200

10. Zone flag fields

GET/POST `/apiv1/zone` returns and accepts only these fields:

Field	Type	Notes
mute	integer	
blockzone	integer	
projectorindex	integer	0-based projector slot. Out of range → 400, zone unchanged.
active	integer	
visualizationid	integer	Writable
ratiox	number	
ratioy	number	
testframe	integer	Writable
visible	integer	

11. PangoScript command catalog

511 commands from the shipped catalog. Run them with POST `/apiv1/ps/execscript` (text/plain) or GET `/apiv1/ps/cmd/{{command}}` for a single statement.

HTML Output

Command	Signature	Description	Example
HtmlClearHead		Check HtmlBody description first. HtmlClearHead clear the Head string.	HtmlClearHead
HtmlHead	htmlhead(string)	Check HtmlBody description first. Set Header string.	HtmlHead "<html><head><title></title></head>"
HtmlClearBody		Check HtmlBody description first. HtmlClearBody clear the Body string.	HtmlClearBody

Command	Signature	Description	Example
HtmlBody	htmlbody(string)	The Guide window contains HTML Viewer component. Primary purpose - style control info window for Universe or similar applications. This group contains commands for window and html parts control (head and body). Default initialization: HeadText:='<html><head><title></title></head>'; BodyText:='<body> Welcome to BEYOND Guide window </body></html>';	var a a='BEYOND' HtmlBody "<body> Welcome to "+a+" Guide window </body></html>"
HtmlUpdate		Check HtmlBody description first. Update Guide window by correct state of Body and Header string	HtmlUpdate
HtmlHide		Check HtmlBody description first. Close Guide window.	HtmlHide

Live Control

Command	Signature	Description	Example
Zoom	zoom(float)	Command sets Zoom parameter of the current Live Control.	Zoom 100
ZoomDelta	zoomdelta(float)	Command sets Zoom parameter of the current Live Control.	ZoomDelta 5 // add 5%
Position	position(float, float, float)	Set Position of the current Live Control. This is absolute value command.	Position 0,0,0
PositionDelta	positiondelta(float, float, float)	Shift Position of the current Live Control. This is relative value command.	PositionDelta 0,10,0 // X,Y,Z
AngleX	anglex(float), Notes:	Set the Angle X parameter into specified value. Angles in degrees. Same can be done using AngleX property of Live Control enabled objects.	AngleX 0; AngleX 45
AngleY	angley(float)	Set the Angle Y parameter into specified value. Angles in degrees. Same can be done using AngleY property of Live Control enabled objects.	AngleY 0
AngleZ	anglez(float)	Set the Angle Z parameter into specified value. Angles in degrees. Same can be done using AngleZ property of Live Control enabled objects.	AngleZ 0; AngleZ 45

Command	Signature	Description	Example
Angle	angle(float, float, float)	Live Control section of the geometric operations includes both a static rotation angle and a rotation speed. The Angle command defines the static rotation angle for the X, Y, and Z axes. All angles are measured in degrees.	Angle 0,0,0; Angle 0,0,90
AngleDelta	angledelta(float, float, float)	AngleDelta increase Angle X, Y and Z of Live Control by specified value. This is relative command. Suggested use - MIDI encoders or	AngleDelta 0,0,0
RotoSpeedX	rotospeedx(float)	Set rotation speed by X only of the current Live Control.	RotoSpeedX 0
RotoSpeedY	rotospeedy(float)	Set rotation speed by Y only of the current Live Control.	RotoSpeedY 0
RotoSpeedZ	rotospeedz(float)	Set rotation speed by Z only of the current Live Control.	RotoSpeedZ 0
RotoSpeed	rotospeed(float, float, float)	Set rotation speed by X, Y and Z of the current Live Control. If you need static angle - use Angle... commands.	RotoSpeed 0,0,0 // rotation speed by X,Y,Z axis. degree.
RotoSpeedDelta	rotospeeddelta(float, float, float)	Increase rotation speed on the current Live Control.	RotoSpeedDelta 0,0,5 // increment of rotation speed. X,Y,Z axis.
Brightness	brightness(float)	Set Brightness parameter of associated Live Control object.	Brightness 100; Brightness 50
VisiblePoints	visiblepoints(float), Notes:	Sets the Visible Point parameters of the Live Control. This is absolute version of the command. Visible Points control how the end of the frame is handled visually.	VisiblePoints 100 // 0..100 (percents); VisiblePoints 50 // 0..100 (percents)

Command	Signature	Description	Example
ColorSlider	colorslider(float)	Set the Color slider position. The slider has neutral range (0..31), then color gradient and white are at the end. Old FB3 DMX profile style. Use Red, Green, Blue and alpha if you need exact color control	ColorSlider 0 // default pos; ColorSlider 255 // default pos
AnimationSpeed	animationspeed(float), Warnings:	Set animation speed of the Live Control. Animation speed parameter works on Cue and Master Live Control level. The Live Control also embedded into Projection zones, but you cannot set the playback speed of Zone. BEYOND plays content, BEYOND does not play Zones.	AnimationSpeed 100; AnimationSpeed 200
ScanRate	scanrate(float)	Sets the Scan rate of the current Live Control. Note, the scan rate is a multiple of the same rate. Final sample rate controlled by the limit of the exact projector.	ScanRate 100
ResetLiveControl		Reset current LiveControl to default state.	ResetLiveControl
ResetRotation		Reset Rotation X,Y and Z of the current Live Control object. Command resets speed and current value to zero.	ResetRotation
Size	size(float, float, float)	Set Size X, Y and Z of the current Live Control.	Size 100,100,100 // Size X,Y,Z axis; Size 200,200,200
SizeDelta	sizedelta(float, float, float), sizedelta(float, float), sizedelta(float)	Changes the size relative to the current value along the X, Y, and Z axes.	SizeDelta 10,0,0; SizeDelta 0,-10,0
BrightnessDelta	brightnessdelta(float)	Relative version of Brightness command. Increase or decrease current Brightness value by provided delta.	BrightnessDelta 5

Command	Signature	Description	Example
VisiblePointsDelta	visiblepointsdelta(float)	Command increases or decreases the Visible Points parameter of currently active Live Control.	VisiblePointsDelta -2
ColorSliderDelta	colorsliderdelta(float)	Shifts the Color slider position. The slider has neutral range (0..31), then color gradient and white are at the end. Old FB3 DMX profile style. Use Red, Green, Blue and alpha if you need exact color control	ColorSliderDelta 0
AnimationSpeedDelta	animationspeeddelta(float)	Increase or decrease animation speed. This is relative version of AnimationSpeed command.	AnimationSpeedDelta 0
ScanRateDelta	scanratedelta(float)	Modifies the scan rate of the current Live Control.	ScanRateDelta -10
PositionIndex	positionindex(integer, float)	One axis position change. Sometimes it helps to have variable axis.	PositionIndex 1,0 // Axis, Value
SizeIndex	sizeindex(integer, float)	Set Size X, Y or Z, using axis index.	SizeDelta 1,50
RGBADelta	rgbdelta(float, float, float, float), rgbdelta(float, float), rgbdelta(float, float, float)	Command modify the RGBA recoloring values in various combinations. The command has three overloads: 2 parameters — channel index and value change. 3 parameters — R, G, and B change, alpha stay as it is 4 parameters — R, G, B, and Alpha change	RGBADelta 16,0,0,0

Command	Signature	Description	Example
RGBA	rgba(float, float, float, float), rgba(float, float), rgba(float, float, float)	Sets the RGBA recoloring values in various combinations. Remember: if Alpha is 0, the RGB values have no effect. If Alpha is 255, Live Control applies full RGB recoloring (except on blank points). The command has three overloads: 2 parameters — channel index and value. 3 parameters — R, G, and B (Alpha remains unchanged). 4 parameters — R, G, B, and Alpha	RGBA 255,255,255,255 //R,G,B,Alpha. 0..255; RGBA 255,0,0,128 //R,G,B,Alpha. 0..255
ResetPosition		Reset Position X,Y and Z of the current Live Control object.	ResetPosition
FX		Set Effect index for FX layers. 0 means off. 1 to 100 is effect index. First argument is index for 1st layer, second argument for second layer, etc. Up to 8 arguments.	FX 0,0,0,0
SetFX		The SetFX command assigns an FX (effect) from the QuickFX table to the current Live Control. Live Control includes eight layers, with one effect per layer. The command is overloade:d -The first argument specifies the layer index (1-8). -The second argument specifies the FX index for that layer. -Additional arguments define FX indexes for subsequent layers. The command is flexible - it can be used to set a single FX layer or multiple layers at once. Layer index value range is 1 to 8 FX index range is 1 to 100	SetFX 1, 10 // Layer number: 1..8, Effect index: 1..100; SetFX 1, 10,10,10,10 // set all 4 first layers to effect number 10; SetFX 5, 0,0,0,0, // just an example of addressing
ShiftFX	shiftfx(integer, integer)	This command moves the focus within the specified layer of the QuickFX table.	ShiftFX 1, 1

Command	Signature	Description	Example
DropFX	dropfx(integer, integer, float)	Runs an effect from the FX table as a "drop effect" for a specified duration. Drop effects are an additional list of effects: they are defined in the main FX table, but triggered temporarily on top of the fixed FX chain. They are intended for short actions such as flashes, triggers, or brief accents.	DropFX 1,1, 1000
SetFXAction	setfxaction(integer, float)	Sets FX effect action. Action is a morph between laser frame before and after effect. This is sort of CPU heavy operation.	SetFXAction 1, 100
FXAction		Set FX Action for FX layers. 0 - effect does not work at all 100 - effect applied fully, without any modification. 1..99 - morph between input and out frame (after effect calculation). Command accept up to 8 arguments, one per layer.	FXAction 100,100,100,100
ZoneFXTimeScale	zonefxtimescale(float, float, float)	This command is deprecated. Use FXTimeScale instead.	ZoneFXTimeScale 1,2,1 // Layer, Clock multiplier, Metronome multiplier
ZoneFXTimeShift	zonefxtimeshift(float, float, float)	This command is deprecated. Use FXTimeShift instead.	ZoneFXTimeShift 1,2,1 // Layer, Clock multiplier, Metronome multiplier
ToggleFX	togglefx(integer)	Toggle FX playback. If layer not specified than toggle all 8 layers. Command work in context of scripter (same as all Live Control commands)	ToggleFX 1 //
PositionX	positionx(float)	Set Position X coordinate of the current Live Control.	PositionX 100 // -100 to 100
PositionY	positiony(float)	Set Position Y coordinate of the current Live Control.	PositionY 100 // -100 to 100
PositionZ	positionz(float)	Set Position Z coordinate of the current Live Control.	PositionZ 0 // -100 to 100
SizeX	sizeX(float)	Sets the X size value of the current Live Control.	SizeX 100
SizeY	sizeY(float)	Sets the Y size value of the current Live Control.	SizeY 100

Command	Signature	Description	Example
SizeZ	sizez(float)	Sets the Z size value of the current Live Control.	SizeZ 100
ColorOn	coloron(float)	Enables laser frame recoloring using the Live Control RGB section. This command specifically sets the Alpha channel to either 0 or 255 - a simple on/off control for alpha.	ColorOn 1 // 1-enabled, 0-disabled
ColorRGB	colorrgb(integer)	Sets the color used for live frame recoloring. The color value is a 24-bit unsigned integer.	ColorRGB 0x0000FF // red
ColorBGR	colorbgr(integer)	Sets the color used for live frame recoloring. The color value is a 24-bit unsigned integer.	ColorBGR 0x0000FF // blue
ResetRotationX		Reset Rotation X of the current Live Control object. Command resets speed and current value to zero.	ResetRotationX
ResetRotationY		Reset Rotation Y of the current Live Control object. Command resets speed and current value to zero.	ResetRotationY
ResetRotationZ		Reset Rotation Z of the current Live Control object. Command resets speed and current value to zero.	ResetRotationZ
InvertRotationX		Invert rotation X direction of the applicable live control.	InvertRotationX
InvertRotationY		Invert rotation Y direction of the applicable live control.	InvertRotationY
InvertRotationZ		Invert rotation Z direction of the applicable live control.	InvertRotationZ
FXTimeScaleAxReset	fxtimescaleaxreset(float, float)	Reset FX time scale of the current Live Control for specified layer and time component (clock, beat)	FXTimeScaleAxReset 1,3 // Layer, Mask (1,2,3)

Command	Signature	Description	Example
FXTimeScaleAx	fxtimescaleax(float, float, float)	Defines the scale (multiplier) for incoming time. In practice, this is a speed control for the clock and beat time. If you set the speed to anything other than 1, BEYOND automatically enables independent time calculation for this Live Control. You can reset it back to the default using FXTimeSync.	FXTimeScaleAx 1,2,1 // Layer, Mask (1,2,3), multiplier; FXTimeScaleAx 1,3,1 // Layer, Mask (1,2,3), multiplier
SetFXMute	setfxmute(integer, float)	Command open ability to mute (disable) FX layers of the Live Control 0-Off, 1-On, 2-Toggle	SetFXMute 1, 2
SetFXMul	setfxmul(integer, float, float)	This is tricky area. The LC object has time accumulator per FX line. Each line has own Clock and Metronome time (optional). Each of time component has speed control. Overall, you can control the speed of each FX line, for clock and metronome components of time.	SetFXMul 2, 4,1 // Layer, Clock, Metronome multiplier
ZoneFXTimeScaleDelta	zonefxtimescaledelta(float, float, float, float)	This command is deprecated. Use FXTimeScaleDelta instead.	ZoneFXTimeScaleDelta 0,0,0,0
ZoneFXTimeShiftDelta	zonefxtimeshiftdelta(float, float, float, float)	This command is deprecated. Use FXTimeShiftDelta instead.	ZoneFXTimeShiftDelta 0,0,0,0
StopFxCell	stopfxcell(float), stopfxcell(float, float)	Stop effect at FX layer of current. The FX line may play up to 4 effects. If you need to stop exact - specify effect index. Otherwise - all effects in the layer.	StopFXCell
ResetFxTiming		This command resets all time-related effects of the current Live Control, including DJ disk, time scale, time multiplier, time shift, and time value.	ResetFxTiming
StopFX		Stop/reset FX of current Live Control	StopFX

Command	Signature	Description	Example
SetFXAction1	setfxaction1(float)	Set action of FX 1 layer. Command act same as clicking on QuickFX grid.	SetFXAction1 100
SetFXAction2	setfxaction2(float)	Set action of FX 2 layer. Command act same as clicking on QuickFX grid.	SetFXAction2 100
SetFXAction3	setfxaction3(float)	Set action of FX 3 layer. Command act same as clicking on QuickFX grid.	SetFXAction3 100
SetFXAction4	setfxaction4(float)	Set action of FX 4 layer. Command act same as clicking on QuickFX grid.	SetFXAction4 100
SetFXMulAx	setfxmulax(integer, integer, float)	Similar to SetFXMul, this command allows to control the speed of FX, but you can define which Layer number: 1..8, Mask (1 is clock, 2 is beat, 3 is clock and beat), Value of multiplier	SetFXMulAx 1, 3, 2 // layer, mask, multiplier
FXTimeSync	fxtimesync(float, float)	FXTimeSync resets the time calculation for an FX layer to its default state, where the FX no longer uses its own time accumulation or speed. The first parameter specifies the FX layer, and the second parameter is the mode; using value 1 resets the layer to the default (parent time). As with all Live Control commands, this instruction applies to the currently active Live Control.	FXTimeSync 1,1 // Set to master/parent time.
FXTimeScaleDeltaAx	fxtimescaledeltaax(float, float, float)	This is relative version of the FxTimeScaleAx command. It changes the scale (multiplier) for incoming time.	FXTimeScaleDeltaAx 1,2, 0.05

Command	Signature	Description	Example
ResetLCTab		Reset Live Control currently used in UI. Most of commands work with exact Live Control containers. The UI (Live Control Tab) can be connected to various Live Controls, such as Cues, ProTracks, etc. This command made for UI mapping.	ResetLCTab
SetFXAction5	setfxaction5(float)	Set action of FX 5 layer. Command act same as clicking on QuickFX grid.	SetFXAction5 100
SetFXAction6	setfxaction6(float)	Set action of FX 6 layer. Command act same as clicking on QuickFX grid.	SetFXAction6 100
SetFXAction7	setfxaction7(float)	Set action of FX 7 layer. Command act same as clicking on QuickFX grid.	SetFXAction7 100
SetFXAction8	setfxaction8(float)	Set action of FX 8 layer. Command act same as clicking on QuickFX grid.	SetFXAction8 100
StrobeSpeed	strobespeed(float)	Command sets the strobe of the current Live Control.	StrobeSpeed 0.2
StrobeSpeedDelta	strobespeeddelta(float)	Command changes the strobe of the current Live Control.	StrobeSpeedDelta 0.1 // period, flicker per second
MulFXMulAx	mulfxmulax(integer, integer, float)	Command controls the speed/multiplier for FX playback	MulFXMulAx 1, 3, 1 // Multiply multiplier...  Layer number: 1..4, Mask (1 is clock, 2 is beat, 3 is clock and beat), Value of multiplier
Hue	hue(float)	Set the Hue component. Negative values treated as not active, and hue has not impact on recolor	Hue -1
HueDelta	huedelta(float)	Increment or decrement the Hue property by specified value. This is relative command	HueDelta 5

Command	Signature	Description	Example
HueShift	hueshift(float)	Hue shift is an additional offset for the hue value. Made for additional oscillation around current hue. If you need to recolor then use Hue, which replaces the input color Hue.	HueShift 90
HueShiftDelta	hueshiftdelta(float)	Command increment/decrement current HueShift value by provided delta. This is relative version of the Sh	HueShiftDelta 5
Saturation	saturation(float)	Sets saturation level of the current Live Control. Saturation is not direct value. In Live Control it works as an offset. Incoming color transformed to HSV, modified and transformed back.	Saturation 0
SaturationDelta	saturationdelta(float)	Shift the saturation level of the current Live Control.	SaturationDelta 0
BeamBrush	beambrush(float), Notes:	Set BeamBrush value (laser line width).	BeamBrush 25; BeamBrush 0; BeamBrush 100
BeamBrushDelta	beambrushdelta(float)	Relative version of BeamBrush command. Increase or decrease BeamBrush value by specified delta	BeamBrushDelta 1

Local & Animation

Command	Signature	Description	Example
SetPage	setpage(integer)	This command sets the index of the currently visible page in the main Grid.	SetPage 1
GetPage		This command is deprecated. Please do not use it. It remains enabled only to avoid syntax check failures in very old scripts. 1027	
Restart	Notes:	This is a script instruction, not a command. It restarts script execution from the beginning and also resets the execution start time, which is used by the WaitForTimePos instruction.	StartCue 1, randomin(1,100) // action Sleep 1000 // delay restart // go to beginning
Exit		EXIT is an instruction to stop script execution. Typically used at the end of script section, such as universe components, or a triggers. Same logic as in SWITCH () in C.	Exit
GetTimeControl		Deprecated. Same as ControlFromTcTab.	GetPage
GetLiveControl		Deprecated. Same as ControlFromLcTab.	GetLiveControl
GetFxControl		Deprecated. Same as ControlFromFxTab.	GetFxControl
CodeName	codename(string)	This command defines the name of the scripter object. The assigned name appears in the PangoScript tab and can also be used as an identifier in the StartCode and StopCode commands.	CodeName "MyCode"
CodeShortcut	codeshortcut(string), Notes:	Defines the shortcut for the script. The only parameters is shortcut name. Use this link for shortcut names: https://docwiki.embarcadero.com/Libraries/Sydney/en/Vcl.Menus.TextToShortCut	CodeShortcut "Ctrl+D"; CodeShortcut "F2"
Autostart	Warnings:	Enable auto start of the script. This script option applicable for scripts located in PangoScript tab. This command inform BEYOND that this script must be executed at start up.	AutoStart

Command	Signature	Description	Example
StopOnBlackout	stoponblackout(integer)	Enables script termination on Blackout. When this option is enabled, the script will stop automatically if Blackout is activated. This is useful for scripts that start or manage multiple Cues, allowing BEYOND to halt the script immediately when the user clicks Blackout.	StopOnBlackout 1
LinePerCycle	linepercycle(integer)	LinePerCycle defines how many script lines are executed during a single iteration of the scripting core. BEYOND can run hundreds of scripts in parallel—some may be stopped, others waiting, and some running in endless loops. To prevent any single script from monopolizing processing time, a limit is imposed on the number of lines executed per cycle. For large or time-critical scripts that need to be processed more quickly, you can increase this value. The default setting is 30 lines per cycle.	LinePerCycle 100 // put at beginning of script.
CodeColorMarker	codecolormarker(integer)	Defines color marker for the script in PangoScript tab. Color in standard GDI RGB encoding, where 0x0000FF is red, 0x00FF00 is green, and 0xFF0000 is blue. Yes, this is not HTML encoding.	CodeColorMarker 0x00FF00
SelectMidi	selectmidi(integer)	Each scripiter has a property that defines which MIDI device it operates with. Command defined index of the MIDI IN and MIDI OUT.	SelectMidi 1 // Select MIDI device pair, 1..4.
MidiOut	midout(integer, integer, integer), midout(integer, integer, integer, integer)	Sends a MIDI message to the MIDI OUT device. By default, the target device is device 1\.. However, if the command is called from MIDI mapping, BEYOND automatically sets the device index based on the current context. The MIDI device index can be manually defined using the SelectMIDI command, which assigns the MIDI device index for this scripiter. The parameters msg, data1, and data2 follow standard MIDI specifications. Make sure to review MIDI fundamentals before using these commands.	MidiOut 0x90, 0x40, 0x7F
DmxOut	dmxout(float, float), dmxout(float, float, float), Notes:	DmxOut command defines state of the DMX channel. BEYOND has multiple DMX buffers, which merged during calculation. All scripts write the data into shared buffer. The buffer channel has 16 bit signed resolution. Value -1 means undefined. Use it if you want to make this channel “transparent”. Values between 0 and 255 are normal values. This command made for output of 1 or 2 channels. If you need more than two then use DmxOutRange	DmxOut 1, 255 // set first DMX OUT channel to 255 (max value)
ChannelOut	channelout(integer, integer)	BEYOND provides 255 general-purpose channels. A channel is an element in an array of 255 normalized floating-point variables. The main concept is that external inputs modify channel values, and these values are then used in effects as multipliers or in other forms of modulation.	ChannelOut 1, 1000; ChannelOut 1, 500
OscOut		Send OSC message out. The Type Tag String composed by type of the arguments. The first argument must be string with OSC address. Command used OSC Client socket.	OscOut “/beam1/xy”, -5, 100
OscOutTTS		Send OSC message out. First argument - OSC address, string. Second argument - OSC TTS, string. Third, 4th, etc - OSC message values. Command used OSC Client socket.	OscOutTTS “/main/volume”, “i”, 100 // Address, Type Tag String, Arguments (optional)
DmxOutRange		DmxOutRange allows to write array of DMX values. First parameter is channel index, 1 to 2048 Second, third, 4th and so on - DMX values. -1 is transparent DMX value. 0 to 255 are standard DMX values	DmxOutRange 1, 100,101,102,103,104 // send 5 DMX values, starting channel 1
SetMidiLayer	setmidilayer(integer)	The command sets the layer index for the MIDI device currently selected in the active Scripiter (range 1..4). Use SelectMidi to select the MIDI device. The default layer index is 1\.. After selecting the device, you can control which layer is active using SetMidiLayer.	SetMidiLayer 1 // default is 1
SetGrid1MSL	setgrid1msl(float)	Sets the Surface layer index for Grid1 in the MIDI Surface Mapping.	SetGrid1MSL 1
SetGrid2MSL	setgrid2msl(float)	Sets the Surface layer index for Grid2 in the MIDI Surface Mapping.	SetGrid2MSL 1
SetSliderMSL	setslidermsl(float)	Sets the Surface layer index for Slider in the MIDI Surface Mapping.	SetSliderMSL 1
SetButtonMSL	setbuttonmsl(float)	Sets the Surface layer index for Buttons in the MIDI Surface Mapping.	SetButtonMSL 1
SetZoneSelMSL	setzoneselmsl(float)	Sets the Surface layer index for Zone Selection in the MIDI Surface Mapping.	SetZoneSelMSL 1
SetZoneMuteMSL	setzonemutemsl(float)	Sets the Surface layer index for Zone Mute in the MIDI Surface Mapping.	SetZoneMuteMSL 1
SetFX1MSL	setfx1msl(float)	Sets the Surface layer index for FX1 in the MIDI Surface Mapping.	SetFX1MSL 1
SetFX2MSL	setfx2msl(float)	Sets the Surface layer index for FX2 in the MIDI Surface Mapping.	SetFX2MSL 1
SetFX3MSL	setfx3msl(float)	Sets the Surface layer index for FX3 in the MIDI Surface Mapping.	SetFX3MSL 1
SetFX4MSL	setfx4msl(float)	Sets the Surface layer index for FX4 in the MIDI Surface Mapping.	SetFX4MSL 1
SetFX5MSL	setfx5msl(float)	Sets the Surface layer index for FX5 in the MIDI Surface Mapping.	SetFX5MSL 1
SetFX6MSL	setfx6msl(float)	Sets the Surface layer index for FX6 in the MIDI Surface Mapping.	SetFX6MSL 1
MidiSysexStart		Creates the buffer for long MIDI Sysex message. Internally buffer is TMemoryStream, you can add as many bytes as you need. Use command MidiSysexAdd to write bytes into the stream. Use command MidiSysexSend to actually send complete MIDI message to output device. Check description of MidiOut command for device indexing.	MidiSysexStart
MidiSysexAdd		Write parameters as bytes into local sysex stream. Check description of MidiSysexStart	MidiSysexAdd “F0477F”; MidiSysexAdd 1,2,3 // add 3 bytes.
MidiSysexSend		Send prepared sysex stream to MIDI device. Use MidiSysexStart to initialize the buffer, and MidiSysexAdd to add bytes to the buffer.	MidiSysexSend
LoadCue	loadcue(string), Warnings:	Load file into current cue of primary grid.	LoadCue “c:\MyAnimationFile.bani”
LoadWorkspace	loadworkspace(string), Warnings:	Load workspace from file. This is HTTP oriented command, do not use in script directly.	LoadWorkspace “c:\MyWorkspace.BeyondWorkspace”
ControlMaster		ControlMaster set the LiveControl destination of this script Master Live Control. Subsequent LiveControl commands, such as Size, will apply to Master LC only.	ControlMaster
ControlCue	controlcue(integer, integer)	ControlCue set the LiveControl destination of this script to one Cue, addressed by page and cue index.	ControlCue 1,1
ControlZone	controlzone(integer), controlzone(string)	ControlZone set the LiveControl destination of this script to one zone, addressed by index or by name.	ControlZone 1 // name as “Main Graphics”, or zone number 1...N; ControlZone “Main Graphics”

Command	Signature	Description	Example
ControlProjector	controlprojector(integer)	Command depreciated. At some point of time BEYOND had Live Control in Projectors.	ControlProjector 1
ControlSelZones		The ControlSelZones command sets the LiveControl destination of the script to the selected projection zones. Subsequent LiveControl commands, such as Size, will apply to all currently selected zones.	ControlSelZones
ControlSelCues		The ControlSelCues command sets the LiveControl destination of the script to the selected Cues. Subsequent LiveControl commands, such as Size, will apply to all currently selected Cues.	ControlSelCues
StartCode	startcode(string), Notes:	Start execution of script with specified name. In addition to that, this is overloaded command. You can add more arguments to this command and they will form parameter list. To read parameter in the script use param(index) function, where index is an index of parameter.	StartCode "MyCode"; StartCode "MyCode", "hello, this is param", 12345 // other script, place in PangoScript tab: Codename "MyCode" // define code name, required to work displaypopup param(1), param(2) //display two parameters
StopCode	stopcode(string)	Stop execution of scripts with specified name.	StopCode "MyCode"
ControlProTrack	controlprotrack(integer)	The ControlProTrack command sets the LiveControl destination of the script to ProTrack addressed by Index. Subsequent LiveControl commands, such as Size, will apply to this ProTrack only.	ControlProTrack 1 // bind to protrack 1
ControlSelProTracks		The ControlSelProTracks command sets the LiveControl destination of the script to the selected ProTracks. Subsequent LiveControl commands, such as Size, will apply to all currently ProTracks.	ControlSelProTracks
SelectGrid	selectgrid(integer)	Each scripiter has a property that defines which Grid it operates on. The Grid index—similar to the MIDI Device index—determines the context in which commands are executed.	SelectGrid 1
SelectTabName	selecttabname(string)	Select Tab by name. Operation by name are better to exact/safer scripting.	SelectTabName "Graphics" //define Page name!
SelectTab	selecttab(integer), Notes:	Selects a Tab by its index. Index-based addressing works for general-purpose scripting, such as MIDI mapping, but it may have safety implications if the tab order changes.	SelectTab 1 //
SelectNextTab	Notes:	Go to the next tab in the workspace. Grid defined by command SelectGrid.	SelectNextTab
SelectPrevTab	Notes:	Select previous Tab of the Grid. Grid defined by command SelectGrid.	SelectPrevTab
SelectPageName	selectpagename(string), Notes:	Select page by the name. String case sensitive.	SelectPageName "Graphics" // define Page name!
SelectPage	selectpage(integer), Notes:	Page is absolute index, used as cue address. The Tab is UI side. The Grid may show subset of Pages because of Category selected.	SelectPage 1 // tab index, 1..N.
SelectNextPage	Notes:	Go to the next page in the workspace. Grid defined by command SelectGrid.	SelectNextPage
SelectPrevPage	Notes:	Select previous Page of the Grid. Grid defined by command SelectGrid.	SelectPrevPage
ControlFromUI		Depreciated. New version ControlFromLcTab	ControlFromUI
SelectAllCat		Show Grid pages of all categories.	SelectAllCat
SelectCatName	selectcatname(string)	Shows only the Grid pages belonging to the specified category.	SelectCatName "Graphics"
SelectCat	selectcat(integer)	Shows only the Grid pages belonging to the specified category.	SelectCat 1 // category index, 1..N.
SelectNextCat		Go to the next category in the workspace. Grid defined by command SelectGrid.	SelectNextCat
SelectPrevCat		Go to the previous category in the workspace. Grid defined by command SelectGrid.	SelectPrevCat
FocusCell	focuscell(integer, integer)	Focuses the specified Grid cell using the provided column and row values. The row and column refer to UI coordinates; internally, BEYOND uses a flat 0-255 cue index. The command operates in the context of the scripiter and the currently active Grid.	FocusCell 1,1 //Column (1..N), Row (1..N)
FocusCellIndex	focuscellindex(integer)	Focuses the specified Grid cell using the flat index, equal to cue index. The command operates in the context of the scripiter and the currently active Grid.	FocusCellIndex 1 // cell index, 1..255
StartCell		Starts playback of the cue located in the current cell of the current Grid. Arguments are optional and specify the names of the projection zones where the cue should output.	StartCell // start current cell
StopCell		Stops playback of the cue located in the current cell of the current Grid.	StopCell
ShiftFocus	shiftfocus(integer)	This command shifts the focus by N cells, forward or backward.	ShiftFocus -1
MoveFocus	movefocus(integer, integer)	Move current (focused) cue of the current Grid	MoveFocus -1,0 //number of cells to go horizontally and vertically
ReStartCell		Restarts playback of the cue in the current cell of the current Grid.	RestartCell
ToggleCell		Toggles (plays or stops) the current cue in the active Grid. This is a UI-oriented command that simply starts or stops the currently selected cue. The command can also accept multiple projection zone names. These define the destination zones where the cue's output will be sent.	ToggleCell // current grid
Echo	echo(integer)	The Echo command sets the echo mode for the TCP Talk Server. 0 - No echo, no confirmation 1 - Confirmation with "OK" or "ERROR" 2 - Echoes back the command along with the status "OK" or "ERROR"	Echo 2 //0-none, 1-ok/error, 2-echo of input plus ok/error
Password	password(string)	Password protection can be enabled for the TCP Talk Server. The password must be defined in the BEYOND.INI file under the "General" section. If the provided password matches, commands from the client will be accepted. No password by default. \[General\] TcpTalkServerPassword=12345	password "enter pasword here"
Version		Command Version designed for TCP Talk Server. BEYOND will respond with exe version information.	version

Command	Signature	Description	Example
Hello		Command Hello is connection test command. Suggested use case - "TCP Talk Server". Essentially, this is plain text communication, one command per line. Your application connects to BEYOND and send Hello command. Response will be "Hello!".	hello
ControlFromLcTab		ControlFromLcTab set the LiveControl destination of this script to whatever selected now in Live Control tab. Command made to read current state of UI.	ControlFromLcTab
ControlFromFxTab		ControlFromFxTab set the LiveControl destination of this script to whatever selected now in QuickFX tab. Command made to read current state of UI.	ControlFromFxTab
ControlFromTcTab		ControlFromTcTab set the LiveControl destination of this script to whatever selected now in Time Control tab. Command made to read current state of UI.	ControlFromTcTab
StoreZoneSelection	Notes:	The StoreZoneSelection command saves the list of selected projection zones into the local storage of the scripter executing the script. You can restore the selection later using ReStoreZoneSelection. The main idea is: store the current selection, modify it, perform the necessary operations, and then restore the original selection.	StoreZoneSelection // use ReStoreZoneSelection to revert back
ReStoreZoneSelection		ReStoreZoneSelection is the counterpart to StoreZoneSelection. It restores the projection zone selection that was previously saved.	ReStoreZoneSelection
SetDmxEditorChannel	setdmxeditorchannel(float, float)	Set the DMX slider value in QuickDMX editor. Good for MIDI or other external control.	SetDmxEditorChannel 1, 128 // firsh slider/channel, and 128 as value
ProjectionZonesDialog		Command call Projection Zone Settings dialog. Suggested use - from UMAX, sometimes you may need to give access to zone adjustments.	ProjectionZonesDialog
MakeSecuredFile	makesecuredfile(string, string, float)	Command designed to help to prepare secured Timeline files.	MakeSecuredFile "Input file name", "Output file name", ClientHardwareSerial
RestorePlayer	restoreplayer(float, float, float, float, string, float, float)	This command is deprecated. There was an idea to generate a script from the current player state (a kind of "capture to script"), but this approach did not work well in practice.	RestorePlayer
RunApp	runapp(string, string), Warnings:	Command call Windows ShellExec() which can start any applications, or other command line. Command disabled by default. Min purpose - automation in installations projects.	RunApp "C:\Windows\notepad.exe", ""
ExecCmd		ExecCmd is a bit fancy. Idea to execute script located in the string. ExecCmd accent any number of arguments, only strings executed as commands. One command per string.	ExecCmd "Blackout"; var s = "Blackout" ExecCmd s
Write	Notes:	Write command add a string to last string of the Script component of the Universe. See WriteLn for more details.	write "Add/ " // run it a few times, it add to end of the current line.
WriteLn	Notes:	WriteLn provides terminal-style output for scripts. Both Write and WriteLn send text to the script's own terminal output buffer. To test the command, place a Script component on the Universe page, then click the Script Edit button. Inside the OnMouseDown section, add: writeln "Hello World!"; Run the page and click the component—your message will appear. WriteLn appends a new line to the terminal's string list, which can be viewed in the Script component on the Universe page.	writeln "Hello world!"
ResetMidiFeedback	resetmidifeedback(float)	Use to force refresh MIDI feedback for the surface. Command reset stored cache of the state, which cause update the MIDI feedback.	ResetMidiFeedback
ResetOscFeedback		Force sending the feedback of all OSC messages. BEYOND OSC feedback based on checking specified variables and sending the OSC message when the value changes. This command to send the state of all variables no matter of state.	ResetOscFeedback
DefineDmxTrigger	definedmxtrigger(integer), definedmxtrigger(integer, string)		DefineDmxTrigger 1
DefineMidiTrigger	definemiditrigger(integer, integer), definemiditrigger(integer, integer, string), definemiditrigger(integer, integer, string, integer)		DefineMidiTrigger 0xC0, 0x00
DefineTrigger	definetrigger(string), definetrigger(string, string)		DefineTrigger "2+2"
DefineTcTrigger	definetctrigger(string)		DefineTcTrigger "TC Test"
InRangeTrigger	inrangetrigger(float, float, string)		InRangeTrigger 0,100,"MyLabel"
OutOfRangeTrigger	outofrangetrigger(float, float, string)		OutOfRangeTrigger 0,100,"MyLabel"
IncreaseTrigger	increasetrigger(float, float, string)		IncreaseTrigger 0,100,"MyLabel"
DecreaseTrigger	decreasetrigger(float, float, string)		DecreaseTrigger 0,100,"MyLabel"
MoreThanTrigger	morethantrigger(float, string), morethantrigger(float, float, float, string)		MoreThanTrigger 0,"MyLabel"
LessThanTrigger	lessthantrigger(float, string), lessthantrigger(float, float, float, string)		LessThanTrigger 0,"MyLabel"
InRangeTriggerCmd	inrangetriggercmd(float, float, string)		InRangeTriggerCmd 0,100,"Blackout"
OutOfRangeTriggerCmd	outofrangetriggercmd(float, float, string)		OutOfRangeTriggerCmd 0,100,"Blackout"

Command	Signature	Description	Example
AnimateProp	animateprop(string, float, float, float), animateprop(string, float, float, float, string)	Probably unusual idea - instead of writing script with loop, tell BEYOND which variable must be "animated" and let application to the work in background. To make it work you need to supply variable/property name as string, beginning and ending values, and duration. Optionally you can specify Event name which will be triggered at the end of animation process.	AnimateProp "master.brightness", 0,100,1000 // look at LiveControl tab, master lc, brightness
AnimatePropDelta	animateproppdelta(string, float, float), animateproppdelta(string, float, float, float, string)	Similar to the AnimateProp command, but instead of using a from-to value pair, it operates in a relative mode. This command changes the specified property value by a given delta.	AnimatePropDelta "master.brightness", -10,1000
DeletePropAni		Deletes property animators associated with the specified property names. The AnimateProp command creates an animator, and DeletePropAni removes it. This command accepts one or multiple property names as strings: DeletePropAni "variable" or DeletePropAni [{"variable", ...}]	DeletePropAni

Main & Workspace

Command	Signature	Description	Example
TimerBeat		BEYOND supports three types of events used in scripts, images, and effects: audio, manual, and timer beats. All three serve the same purpose - to initiate a process or event that can be used elsewhere. TimerBeat initiated by system metronome located on main toolbar of BEYOND. One event every beat.	TimerBeat
AudioBeat		BEYOND supports three types of events used in scripts, images, and effects: audio, manual, and timer beats. All three serve the same purpose - to initiate a process or event that can be used elsewhere. Audio beat events are generated by the audio input using an FFT-based peak detector. Manual events result from user interaction, while timer events are produced by the system-wide metronome. AudioBeat command initiate audio-beat event, same as audio analyzer. You may need if you have external audio peak detector.	AudioBeat

Command	Signature	Description	Example
ManualBeat		BEYOND supports three types of events used in scripts, images, and effects: audio, manual, and timer beats. All three serve the same purpose - to initiate a process or event that can be used elsewhere. ManualBeat command initiate manual-beat event.	ManualBeat
CueDown	cuedown(integer, integer), cuedown(integer, integer, integer)	Command perform "press" action on the Grid (workspace cue). Exact action depends on Grid mode - toggle, flash, etc. Command is not depending on the Grid (main or secondary) as well as from current page.	CueDown 1,1 // simulate client on 1st page, 1st cue
CueUp	cueup(integer, integer), cueup(integer, integer, integer)	Command perform "release" action on the Grid (workspace cue). Exact action depends on Grid mode - toggle, flash, etc. Command is not depending on the Grid (main or secondary) as well as from current page.	CueUp 1,1 // first page, first cue
StopCueNow	stopcuenow(integer, integer)	Main Stop command. Stop playback of the cue.	StopCueNow 1, 1 // page 1..255, cue 1..255
StopAllSync	stopallsync(float)	Stops Cue playback in the Grid, ProTrack, and the Effect/Group pool.	StopAllSync 0.5
StopAllAsync	stopallasync(float)	Stops Cue playback in the Grid, ProTrack, and the Effect/Group pool with specified fade out time.	StopAllAsync 0.5 // duration in seconds
StopAllNow		Stops playback of all cues and effects, including the Grid player pool, ProTracks, Effects, Zone Effects, and the DMX Server. Playback is stopped immediately, without a soft stop. This is the primary stop command used internally by BEYOND.	StopAllNow
StopCueSync	stopcuesync(integer, integer, float)	Stops playback of the specified Cue. This is legacy command. Do not use.	StopCueSync 1, 1, 0.5

Command	Signature	Description	Example
EnableLaserOutput	Notes:, Warnings:	Enable Laser Output. This command change internal mode of BEYOND to streaming. Streaming means output to laser projector, which initiate creation of calculation threads.	EnableLaserOutput
DisableLaserOutput	Warnings:	Disable Laser Output. This command change internal mode of BEYOND and stop output to all projectors, release threads and decrease CPU usage.	DisableLaserOutput
MasterPause	masterpause(integer)	Set the playback on pause. Applied to laser and audio playbacks.	MasterPause Toggle // ON, OFF, TOGGLE
BlackOut	Warnings:	System wide reset. Has most impact when laser output enabled. Command has configuration for how many systems it reset. Main purpose - universal stop and reset.	BlackOut
VirtualLJ	virtualLj(integer)	Command enable, disable of toggle Virtual LJ (see button on main toolbar).	VirtualLJ Toggle // ON, OFF, TOGGLE
BeatTap		Tap tempo. Set the master BPM by means of tapping. Made for external MIDI control. BEYOND approximate last 4 taps. Max BPM is 600\, Use SetBPM command if you want to set exact value. You can read and write current BPM using Master.BPM property.	BeatTap
BeatResync		Resets the master beat timer and sets its phase to zero. This command is used to synchronize the master beat timer and takes effect immediately. No parameters are required. Reset master beat timer, set phase to zero. Command made to synchronize the master beat timer. Act immediately. No parameters required.	BeatResync
OneCue	Notes:	The OneCue command sets the one cue mode. In this mode, the Grid allows one cue to be started.	OneCue

Command	Signature	Description	Example
MultiCue		The MultiCue command enables multi-cue mode. In this mode, the Grid allows multiple cues to be started and played in parallel.	MultiCue
Transition	transition(integer)	Command enable, disable or toggle transition between cues in the Grid.	Transition Toggle // ON, OFF, TOGGLE
ClickSelect	clickselect(float)	Enable Select click mode of Live / Content Grid.	ClickSelect
ClickToggle	clicktoggle(float)	Enable Flash click mode of Live / Content Grid.	ClickToggle
ClickRestart	clickrestart(float)	Enable Restart click mode of Live / Content Grid.	ClickRestart
ClickFlash	clickflash(float)	Enable Flash click mode of Live / Content Grid.	ClickFlash
ClickLive	clicklive(float)	Enable Live click mode of Live / Content Grid.	ClickLive
ClickSoloFlash	clicksoloflash(float)	Enable Solo-Flash click mode of Live / Content Grid.	ClickSoloFlash
PauseCue	pausecue(integer, integer, integer), Notes:	Command finds the cue player, and set/remove the pause state. Command use default soft pause durations.	PauseCue 1,1, TOGGLE // page index, cue index, action (ON, OFF, TOGGLE)
VLJFX	vljfx(integer, integer)	Command enable/disable VLJ of QuickFX tab.	VljFX 1, Toggle // index of FX-VLJ 1..8, and action: ON, OFF, TOGGLE
ClickScrollZoom	clickscrollzoom(float)	Zoom slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollZoom -5
ClickScrollSize	clickscrollsize(float)	Size slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollSize -5

Command	Signature	Description	Example
ClickScrollFade	clickscrollfade(float)	Fade slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollFade -5
ClickScrollVPoints	clickscrollvpoints(float)	Visible points slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollVPoints -5
ClickScrollScanRate	clickscrollscanrate(float)	ScanRate slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollScanRate -5
ClickScrollColor	clickscrollcolor(float)	Color slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollColor -5

Command	Signature	Description	Example
ClickScrollAniSpeed	clickscrollanispeed(float)	Animation speed slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollAniSpeed -5
ClickScrollR	clickscrollr(float)	Red slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollR -5
ClickScrollG	clickscrollg(float)	Green slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollG -5
ClickScrollB	clickscrollb(float)	Blue slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollB -5

Command	Signature	Description	Example
ClickScrollA	clickscrolla(float)	Alpha slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollA -5
OnePer		The OnePer command sets the one-per-zone mode. This is legacy mode, which is not recommended to use. Such model was applicable at BEYOND version 1 or 2\). With increased calculation chain length, this mode become very hard to use and support.	OnePer
SetBpm	setbpm(float)	Sets the master BPM - the global tempo of the BEYOND application. There is only one tempo clock for the entire system.	SetBpm 120
SetBpmDelta	setbpmdelta(float)	Increment or decrement current BPM. You can have access to it using Master.BPM property.	SetBpmDelta 1
RestartCue	restartcue(integer, integer)	If a cue is currently playing in the Grid or ProTrack, this command resets its playback time to the beginning.	RestartCue 1,1 // page, cue
ClickLCTabMode	clicklctabmode(float)	ClickLcTabMode command is same as clicking Master, Cue, Zone, ProTrack buttons in FX Tab. Command defined which Live Control is connected to Live Control tab (UI). Use following constants: 1-Master, 2-Cue, 3-Zone, 4-ProTrack	ClickLCTabMode 1 // 1-Master, 2-Cue, 3-Zone, 4-ProTrack

Command	Signature	Description	Example
ClickFXTabMode	clickfxtabmode(float)	ClickFXTabMode command is same as clicking Master, Cue, Zone, ProTrack buttons in FX Tab. Command defined which Live Control FX is connected to QuickFX tab. Use following constants: 1-Master, 2-Cue, 3-Zone, 4-ProTrack	ClickFXTabMode 1
MasterSpeed	masterspeed(float)	Sets Master.MasterSpeed property, which is general purpose speed multiplier.	MasterSpeed 1 // From zero to one
StartPrevious		Performs the same action as the Back button on the main toolbar. This command starts the previously played Cue.	StartPrevious
TogglePrevious		A cue click history-based operation. The command starts the previously played cue, acting as a kind of "swap" function.	TogglePrevious // "SWAP" button
UnselectAllProTracks		Clears the selection state of all ProTracks. After this command, no ProTracks remain selected.	UnselectAllProTracks
SelectProTrack		Select one or multiple ProTracks by index or by the name. String arguments used as names, numeric as indexes (1 to 8)	SelectProTrack 1
UnselectProTrack		Clears the selection state of the specified ProTracks. Command accept multiple arguments - strings as ProTrack name, and numbers as ProTrack index.	UnselectProTrack 1,2,3,4 // unselect 1,2,3 and 4th ProTrack
ToggleSelectProTrack		Toggles the Selected state of the specified ProTrack. The command accepts multiple arguments: strings are treated as ProTrack names, and numeric values are treated as ProTrack indexes.	ToggleSelectProTrack 1
UnmuteAllProTracks		Removes the muted state from all ProTracks.	UnmuteAllProTracks
MuteProTrack		Mute specified ProTracks. Command accept multiple arguments - strings as ProTrack name, and numbers as ProTrack index.	MuteProTrack 1 // index(es) or name(s)

Command	Signature	Description	Example
UnmuteProTrack	Notes:	Removes the muted state from specified ProTrack. Command accept multiple arguments - strings as ProTrack name, and numbers as ProTrack index.	UnmuteProTrack 0,1,2,3 // unmute first four protracks
ToggleMuteProTrack		Toggles the muted state of the specified ProTrack. The command accepts multiple arguments: strings are treated as ProTrack names, and numeric values are treated as ProTrack indexes.	ToggleMuteProTrack 1 // index(es) or name(s); ToggleMuteProTrack 1,2,3,4 //
UnselectAllFixt		Clears the selection state of all Fixtures. After this command, no Fixtures remain selected.	UnselectAllFixt
SelectFixt		Set Selected state of the Fixture object	SelectFixt 0 // index(es) or name(s)
UnselectFixt		Clears the selection state of the specified Fixtures. Command accept multiple arguments - strings as fixture name, and numbers as fixture index.	UnselectFixt 0,2, "Spot 1" // two indexes and one name of the Fixture
ToggleSelectFixt		Toggles the Selected state of the specified Fixtures. The command accepts multiple arguments: strings are treated as Fixture names, and numeric values are treated as Fixture indexes.	ToggleSelectFixt
ClickFXStopAll		ClickFXStopAll is combine action command, which send StopFX command, and stop FX VLJ actions.	ClickFXStopAll
ClickFxVlj	clickfxvlj(float), clickfxvlj(float, float)	ClickFxVlc is action performed by 8 VLJ buttons in QuickFX tab. It enable, disable or toggle corresponding VLJ button state. You must specify line index. Default action - toggle.	ClickFxVlj 2 // 0 off, 1 on, 2 toggle.

Command	Signature	Description	Example
FXCellDown	fxcelldown(integer, integer)	This command performs an FX table cell click using absolute indexing, which does not account for any horizontal shift of the table. It predates the FXCellClick command — in earlier versions, the number of effects was small and the FX grid was not scrollable. (The content grid is still not scrollable.) Consider this command as operating on absolute indexes, while FXCellClick is more UI-oriented.	FXCellDown 1, 1 // Layer number: 1..8, Effect number
UnmuteAllZone		Removes the muted state from all Projection Zones.	UnmuteAllZone
MuteZone		Mute specified Projection Zones. Command accept multiple arguments - strings as Zone name, and numbers as Zone index.	MuteZone 0
UnmuteZone	Notes:	Removes the muted state from specified Projection Zone. Command accept multiple arguments - strings as zone name, and numbers as zone index.	UnmuteZone 0
ToggleMuteZone	Warnings:	Toggles the muted state of the specified Projection Zone. The command accepts multiple arguments: strings are treated as ProTrack names, and numeric values are treated as ProTrack indexes.	ToggleMuteZone 0
ToggleMuteZoneOfProjector	togglemutezoneofprojector(float), togglemutezoneofprojector(float, float)	Toggle mute state of all zones associated with the projector.	ToggleMuteZoneOfProjector 1 // 1st projector; ToggleMuteZoneOfProjector 1, 0 // 1st projector, 0 unmute
ClickLockSize	clicklocksize(float)	Enable or disable synchronized change of Size X and Size Y sliders in Live Control tab. Basically, moving both sliders together.	ClickLockSize 1 // 0,1,2 Off, On, toggle
StartAudioRecord		Start audio recording in QuickAudio tab (for spectrum, and audio triggers).	StartAudioRecord
StopAudioRecord		Stop audio recording in QuickAudio tab.	StopAudioRecord
SetAudioRelease	setaudiorelease(float)	Set Audio Release speed (Quick Audio tab)	SetAudioRelease 75

Command	Signature	Description	Example
SetAudioGain	setaudiogain(float)	Deprecated. FFT Audio compensation (1 to 15). Is not related to audio playback.	SetAudioGain 10
MuteSelected	muteselected(integer)	Sets the muted state for the selected Projection Zones, ProTracks and Fixtures.	MuteSelected 2 // 0,1,2
SetGridSize	setgridsize(float, float)	Command set the main grid size.	SetGridSize 8,5
MasterPauseTime	masterpausetime(integer)	Sets master pause duration (the soft pause)	MasterPauseTime 1000 // set pause/unpause time to 1 second
SetGrid1Page	setgrid1page(integer)	Set the Main Grid page by index.	SetGrid1Page 1
SetGrid2Page	setgrid2page(integer)	Set the Secondary Grid (bottom side) page by index.	SetGrid2Page 1
MuteAllZones		Mute all Projection Zones. Muted zone has no output to laser projector.	MuteAllZones
PreviewNxN	previewnxxn(integer), previewnxxn(integer, integer, integer)	Sets the Main Preview mode, same as context menu. Modes: 1 - laser 2 - 2x2 3 - 3x3 4 - 4x4 11- Custom 1 12-Custom 2 13- Custom 3 In custom modes you can specify number of rows and columns.	PreviewNxN 4 // 1 - laser, 2 - 2x2, 3 - 3x3, 4 - 4x4, 11- Custom1, 12-Custom2, 13- Custom 3
DisplayPopup	Notes:	Shows popup window. This is pure UI oriented command. Take any number of parameters of any type.	DisplayPopup "Hello world!"; DisplayPopup Master.BPM; DisplayPopup 123, "hello!"
DisplayPreview		Display message in Main Preview Panel located in Right/Top corner of BEYOND window. Recommend to use for debug messages, or action confirmation where you do not need a window based message. Command take any number parameters of any type.	DisplayPreview "Hello world!", 0x0000ff
HoldClick	holdclick(integer)	Enable HOLD mode for Cue playback.	HoldClick 1
PreviewMaster		Sets the Main Preview panel to Master View mode. This action also clears the zone selection, since zones cannot be viewed or selected in Master View mode.	PreviewMaster

Command	Signature	Description	Example
StopCueType	stopcuetype(integer), stopcuetype(integer, float)	Stop cue playback of a specified type. 1 - general images 2 - timeline 4 - DMX 8 - Animator 16 - Beam sequences 32 - Capture 64 - Chains (not supported now) 128 - Script (not supported now) 256 - Records (not publicly available yet) 512 - Zone effects 1024 - effect pool (group mode)	StopCueType 255
FXScroll	fxscroll(integer)	Set FX table horizontal scroll position	FXScroll 1
FXClick	fxclick(integer, integer)	Click on FX cell without using current UI left/right scroll. This index is absolute FX index	FXClick 1,1
FXCellClick	fxcellclick(integer, integer)	The FX table is wider than the available UI space. This command performs an FX table cell click using relative indexing, which accounts for any horizontal shift of the table. In practice, it clicks by column position rather than by the absolute effect index.	FXCellClick 1,1
TapByCueClick	tapbycueclick(integer)	Enables, disables, or toggles Tap-Tempo on Cue click. When enabled, clicking a Cue in the Grid generates a Tap-Tempo (BPM) click. The command can turn this option on, off, or toggle its current state.	TapByCueClick On // options: On,Off, Toggle
ResyncByCueClick	resyncbycueclick(integer)	This command sets the beat synchronization mode that is applied when a cue is clicked. You can use the predefined constants OFF, ON, and TOGGLE. The command is related to the master BPM (metronome) control, which also includes functions such as tap and resync.	ResyncByCueClick On

Command	Signature	Description	Example
SynchronizePlayerToBeat	synchronizeplayertobeat(integer)	Sets the global option for players to synchronize to the beat. Synchronization occurs on player start and restart. The idea is similar to quantize in audio software: it is nearly impossible to click exactly on the beat, so the timing will always be slightly off. This option corrects that by using the current sub-beat time as the alignment reference.	SynchronizePlayerToBeat On
ResetCuesFX		Iterates through all cues in the workspace and performs the following actions: clears assigned FX, resets the FX Action parameter and timing, and clears the Drop-effect list All these operations are part of the LiveControl object. The purpose of this command is to reset the entire playback FX state.	ResetCuesFX
ResetZonesFX		Iterates through all Projection zones and performs the following actions: clears assigned FX, resets the FX Action parameter and timing, and clears the Drop-effect list All these operations are part of the LiveControl object. The purpose of this command is to reset the entire playback FX state.	ResetZonesFX
ResetProTrackFX		Iterates through all ProTracks and performs the following actions: clears assigned FX, resets the FX Action parameter and timing, and clears the Drop-effect list All these operations are part of the LiveControl object. The purpose of this command is to reset the entire playback FX state.	ResetProTrackFX

Command	Signature	Description	Example
ClickTCTabMode	clicktctabmode(float)	ClickTcTabMode command is same as clicking Master, Cue, Zone, ProTrack buttons in FX Tab. Command defined which Live Control is connected to Time Control tab. Use following constants: 1-Master, 2-Cue, 3-Zone, 4-ProTrack	ClickTCTabMode 1 // 1-Master, 2-Cue, 3-Zone, 4-ProTrack
ResetMasterFX		Reset Master Live Control FX only.	ResetMasterFX
PlayListStop		Stop playback of the show in PlayList	PlayListStop
PlayListPlay		Start playback of the current show from the PlayList.	PlayListPlay
PlayListFirst		Go to first show from the PlayList	PlayListFirst
PlayListLast		Go to last show from the PlayList	PlayListLast
PlayListNext		Go to next show from the PlayList	PlayListNext
PlayListPrev		Go to previous show from the PlayList.	PlayListPrev
PlayListSetPos	playlistsetpos(float)	Select the Show from PlayList by index (1..N)	PlayListSetPos 1
PlayListSetTime	playlistsettime(float)	Jump to time position of the current show	PlayListSetTime 0
LoadPlaylist	loadplaylist(string), Warnings:	Open file into BEYOND Playlist window.	LoadPlaylist "c:\MyPlaylist.BeyondSL"
InvertPlayersTime		The BEYOND content player supports time direction control, allowing playback to bounce between time points A and B as part of the Time Control section. This command toggles the playback direction for players located in the Grid (dynamic pool) or the ProTrack array. The action depends on the message source—it may apply to selected zones, selected cues, selected ProTracks, or, in the case of a Master/General source, to all ProTracks and Grid players.	InvertPlayersTime

Command	Signature	Description	Example
PlayersSetJump	Notes:	This command stores the current player time on the first call, and on the second call jumps back to the stored position. It is another variation of DJ-style live time effects.	PlayersSetJump
PlayersResetJump	Notes:	Clears the time stored in player by command PlayerSetJump. Command might be applied to ProTracks and Grid players, depending on selection.	PlayersResetJump
PlayersSetLoop	playerssetloop(integer), Notes:	The idea of this feature is to loop the player between two points and allow back-and-forth playback. It provides three actions in total: 1\ Store point A. 2\ Store point B (looping begins). 3\ Clear the loop settings.	PlayersSetLoop -1 // automatic mode, set A, set B, clear.
PlayersDisk	playersdisk(float), playersdisk(float, float), Notes:	Time Control command. The DJ Disk located in the Time Control tab can slow down the playback as well as to time shift (scratch). If you need time shift only action look at PlayerDiskShift command.	PlayersDisk 1, 0.1 // speed scale, the shift
PlayersDiskShift	playersdiskshift(float), Notes:	Time Control command. This command is simplified version of PlayerDisk. Shift applied to clock and beat time components .	PlayersDiskShift 0.1
InvertProTrackTime		Inverts the playback direction of the selected ProTracks. The main idea is to reverse the playback temporarily, allowing quick back-and-forth motion—similar to a DJ-style effect.	InvertProTrackTime
ProTrackSetJump		This command stores the current player time on the first call, and on the second call jumps back to the stored position. It is another variation of DJ-style live time effects.	ProTrackSetJump
ProTrackResetJump		Clears the time stored in player by command PlayerSetJump. Command always applied to the ProTracks only.	ProTrackResetJump

Command	Signature	Description	Example
ProTrackSetLoop	Notes:	The idea of this feature is to loop the player between two points and allow back-and-forth playback. It provides three actions in total: 1\ Store point A. 2\ Store point B (looping begins). 3\ Clear the loop settings.	ProTrackSetLoop -1 // automatic mode, set A, set B, clear.
ProTrackDisk	protrackdisk(float), protrackdisk(float, float)	Time Control command, applied to ProTracks only. The DJ Disk located in the Time Control tab can slow down the playback as well as to time shift (scratch). If you need time shift only action look at PlayerDiskShift command.	ProTrackDisk 1, 0.1
ProTrackDiskShift	protrackdiskshift(float)	Time Control command, applied to ProTracks only. This command is simplified version of PlayerDisk. Shift applied to clock and beat time components .	ProTrackDiskShift 0.1
MasterAudioVolume	masteraudiovolume(float)	Controls Timeline audio players volume.	MasterAudioVolume 100 // volume in percent.
MasterShowBrightness	mastershowbrightness(float)	Set the Timeline/Show brightness. Specific for this content. Main intention - quick and universal way to fade the show playback not matter of place.	MasterShowBrightness 100
MasterClockShift	masterclockshift(float)	MasterClockShift sets Master.MasterClockShift property. MasterClockShift is part of the base offset value applied to cue calculations on a per-zone basis. The offset is sum of the master offset and the cue level offset. Simple way to see it - set cue routing to multiple zones, and set the shift value.	MasterClockShift 0 // in seconds
MasterMetroShift	mastermetroshift(float)	MasterMetroShift sets Master.MasterMetroShift property. MasterMetroShift is part of the base offset value applied to cue calculations on a per-zone basis. The offset is sum of the master offset and the cue level offset. Simple way to see it - set cue routing to multiple zones, and set the shift value.	MasterMetroShift 0

Command	Signature	Description	Example
MasterEffectClockShift	mastereffectclockshift(float)	Defines global offset for effects (clock component). This shift works if effect has enabled master shift, and work as additional offset to other offsets.	MasterEffectClockShift 1 // 0 is default
MasterEffectMetroShift	mastereffectmetroshift(float)	Defines global offset for effects (metronome component). This shift works if effect has enabled master shift, and work as additional offset to other offsets.	MasterEffectMetroShift 1 // beats
MasterCueSpeed	mastercuespeed(float), Notes:	The MasterCueSpeed command sets the CueSpeed value of the Master object. This parameter acts as a speed multiplier for all Player objects. The value is measured in percent, with 100 as the default.	MasterCueSpeed 200 // twice faster
MasterCueLcSpeed	mastercuelcspeed(float), Notes:	The MasterCueLcSpeed command sets the CueLcSpeed value of the Master object. This parameter acts as a speed multiplier for all Live Control objects, including those embedded in cues, zones, and other elements. The value is measured in percent, with 100 as the default.	MasterCueLcSpeed 100 // default speed
MasterLCSpeed	masterlcspeed(float)	Sets the Master.MasterLcSpeed property. MasterLcSpeed is a multiplier applied to the time delta during time accumulation for Live Control objects	MasterLCSpeed 100
MasterFXSpeed	masterfxspeed(float)	Sets the Master.MasterFxSpeed property. MasterFxSpeed is a multiplier applied to the time delta during time accumulation for FX layers.	MasterFXSpeed 100
MasterZoneFxSpeed	masterzonefxspeed(float)	Sets the Master.MasterZoneFxSpeed property. MasterZoneFxSpeed is a multiplier applied to the time delta during time accumulation for FX layers.	MasterZoneFxSpeed 100 // in percents

Command	Signature	Description	Example
MasterShowSpeed	mastershowspeed(float)	Set the Timeline/Show playback speed. Specific for this content type. Main intention - quick and universal way to control the timeline speed.	MasterShowSpeed 100
MasterAudioVolumeMute	masteraudiovolumemute(float)	Mute or unmute Timeline audio players.	MasterAudioVolumeMute 1 // On(1), Off(0), Toggle(2)
EnableFb3StyleDmxIn	enablefb3styledmxin(float)	EnableFb3StyleDmxIn is a legacy oriented command. BEYOND has multiple modules which react on DMX IN Data. One of oldest is FB3 Style DMX IN, which came into BEYOND from QuickShow. This is very simplified approach to DMX IN, considering modules like DMX to PangoScript and DMX Server of BEYOND. Anyway, this is standard disable/enable/toggle command.	EnableFb3StyleDmxIn 1
MasterShowShift	mastershowshift(float)	Made to add some offset to laser calculation. Audio may have delay, at least due to distance. A client requested to be able to delay laser a bit	MasterShowShift 0
MuteAllProTracks		Mute all ProTracks of BEYOND. In muted state the calculation still happens, but output to laser disabled.	MuteAllProTracks
StopProTrack		Stop Playback of the specified ProTracks. Command accept multiple arguments - strings as ProTrack name, and numbers as ProTrack index.	StopProTrack 1,2,3,4
StopProTrackFX		No arguments - stops FX on all ProTracks. One argument - stops FX on the specified ProTrack (by index or name). Multiple arguments - stops FX on the specified ProTrack, with the second and subsequent arguments specifying the layer indexes to stop (multi effect per line mode).	StopProTrackFX; StopProTrackFX 1
FocusProTrack	focusprotrack(string), focusprotrack(float)	Command defines the focused ProTrack in the UI.	FocusProTrack 1 // Track index or name

Command	Signature	Description	Example
SetProTrackZones		Defines routing information for the specified ProTrack. Routing consists of one or more Projection Zones. The first argument is the ProTrack name or index. If the name does not exist or the index is out of range, the command exits silently. The second and subsequent arguments are the Projection Zones, specified either by name or index. String values are treated as names; numeric values are treated as indexes.	SetProTrackZones 1, "Scanner 1 Main", "Scanner 2 Main" // Track index/name, Zone name(es)/index(es)
UnSoloAllProTrack		Removes the Solo state from all ProTracks.	UnSoloAllProTrack
SoloProTrack		Command sets the Solo state on specified ProTrack. Command accept multiple arguments - strings as ProTrack name, and numbers as ProTrack index.	SoloProTrack 1
UnSoloProTrack		Command removes the Solo state from specified ProTrack. Command accept multiple arguments - strings as ProTrack name, and numbers as ProTrack index.	UnSoloProTrack 1 // index(es) or name(s)
ToggleSoloProTrack		Toggles the Solo state of the specified ProTrack. The command accepts multiple arguments: strings are treated as ProTrack names, and numeric values are treated as ProTrack indexes.	ToggleSoloProTrack 1
ClickTrack	clicktrack(float)	Enable Track click mode of Live / Content Grid.	ClickTrack
DisplayPopupOnTop		A version of DisplayPopup with higher visibility priority.	DisplayPopupOnTop "Hello world!"
GoGridMode		Switches BEYOND to the Grid view, equivalent to clicking the corresponding button on the main toolbar.	GoGridMode
GoTimelineMode		Switches BEYOND to the Timeline view, equivalent to clicking the corresponding button on the main toolbar.	GoTimelineMode

Command	Signature	Description	Example
GoPlayListMode		Switches BEYOND to the Play List view, equivalent to clicking the corresponding button on the main toolbar.	GoPlayListMode
GoUniverseMode		Switches BEYOND to the Universe view, equivalent to clicking the corresponding button on the main toolbar.	GoUniverseMode
CloseUmax		UMAX is a full-screen "universe" window designed to cover the entire display. It responds only to the Alt + F4 shortcut and normally prevents manual closing. To close it from a MIDI command or script, use the Close UMAX command.	CloseUmax
EnableDmxIn	enabledmxin(float)	EnableDmxIn is a dual-action command that enables, disables, or toggles the DMX IN state, while also resetting the state of all DMX Server players. The reset of DMX Server players is always performed. When DMX IN is enabled, the command updates the state of the Script core and the FB3-style DMX IN object. It also updates the Master.DmxInButtonDown property and the corresponding UI button on the main toolbar.	EnableDmxIn Toggle // 0,1,2 Toggle; EnableDmxIn 0
SetEffectChannelAction	seteffectchannelaction(float), seteffectchannelaction(float, float)	Defines channel action parameter which impact channel calculation.	SetEffectChannelAction 0 // 0..100
SetChannelToChannelRatio	setchanneltochannelratio(float)	Channel is a shell of 4 values and 4 multipliers with auto-gain correction. ChannelToChannel is the multiplier for original channel value. Additional parts are - DMX, VDJ and FFT multipliers. Overall logic: Gain = ChannelRatio + DmxRatio + FFTRatio + VdjRatio; OutValue = ChannelVal $\sqrt{\frac{\text{ChannelRatio}}{\text{Chain}} + \frac{\text{DmxVal}}{(\text{DmxRatio}/\text{Gain})} + \frac{\text{FFTVal}}{(\text{V}/\text{FFTRatio}/\text{Gain})} + \text{VdgValue} \cdot (\text{VdjRatio}/\text{Gain})}$	SetChannelToChannelRatio 100

Command	Signature	Description	Example
SetDmxToChannelRatio	setdmxtochannelratio(float)	BEYOND Channels hide several internal mechanisms, one of which is the ability to mix DMX data with direct channel values. A channel is a general-purpose, global array of variables that can be used in Images and Effects. You can view them in the Channels tab. This command sets the multiplier (ratio) that defines how DMX data contributes to the Channel values. See SetChannelToChannel for more details.	SetDmxToChannelRatio 50; SetDmxToChannelRatio 100
SetFFTTToChannelRatio	setffttochannelratio(float)	Command sets the ratio of Audio/FFT data in the output channel value. See SetChannelToChannel for more details.	SetFFTTToChannelRatio 100
SetVdjToChannelRatio	setvdjtochannelratio(float)	Command sets the ratio of VDJ data in the output channel value. See SetChannelToChannel for more details.	SetVdjToChannelRatio 0 // 0 to 100.
TimelinePlay		Command from the Timeline Editor group. Plays the timeline from the current position.	TimelinePlay
TimelineStop		Command from the Timeline Editor group. Stops timeline playback.	TimelineStop
TimelineJumpToStart		Command from the Timeline Editor group. Jump the the timeline beginning.	TimelineJumpToStart
TimelineJumpToEnd		Command from the Timeline Editor group. Jump the the timeline ending.	TimelineJumpToEnd
TimelineJumpDelta	timelinejumpdelta(float)	Shift timeline cursor position by specified number of seconds	TimelineJumpDelta 1
TimelineSetViewRange	timelinesetviewrange(float, float)	Set timeline visibility range (timeline view area from left to right)	TimelineSetViewRange 1,5
TimelineShiftViewRange	timelineshiftviewrange(float)	Shift timeline visibility left or right.	TimelineShiftViewRange 1 //one second shift
TimelineSetPos	timelinesetpos(float)	Sets timeline position in seconds.	TimelineSetPos 10 // 10th second.
TimelineShowItNow	timelineshowitnow(float)	Command from the Timeline Editor group. Defined state of ShowItNow button (output to laser).	TimelineShowItNow 1

Command	Signature	Description	Example
TimelinePlayFromMarker	timelineplayfrommarker(string)	Command from the Timeline Editor group. Find the marker by name and start playback from marker position.	TimelinePlayFromMarker "MyMarker" // you should have the name of marker in your timeline.
TimelineQuickSave	Notes:	Save current timeline to file. Command use existing file name.	TimelineQuickSave
TimelineFirstTab		Command from the Timeline Editor group. Go to the first timeline/tab of the editor.	TimelineFirstTab
TimelineNextTab		Command from the Timeline Editor group. Go to next timeline/tab of the editor.	TimelineNextTab
TimelinePrevTab		Command from the Timeline Editor group. Go to previous timeline/tab of the editor.	TimelinePrevTab
TimelineLastTab		Command from the Timeline Editor group. Go to the last timeline/tab of the editor.	TimelineLastTab
TimelineSetTabIndex	timelinesettabindex(float)	Command from the Timeline Editor group. Go to timeline/tab by index.	TimelineSetTabIndex 1
TimelineSetTabName	timelinesettabname(string)	Command from the Timeline Editor group. Go to timeline/tab by index.	TimelineSetTabName "MyShow"
TimelineEnableTC	timelineenabletc(float)	Command enables or disable using of incoming timecode for the playback.	TimelineEnableTC
TimelineNextMarker		Command from the Timeline Editor group. Go to next marker (legacy name - log). Marker is vertical line, a time position on the timeline.	TimelineNextMarker
TimelinePrevMarker		Command from the Timeline Editor group. Go to previous marker.	TimelinePrevMarker
TimelineNextEditPoint		Command from the Timeline Editor group. Go to next edit point such as event boundary, marker, etc.	TimelineNextEditPoint
TimelinePrevEditPoint		Command from the Timeline Editor group. Go to previous edit point such as event boundary, marker, etc.	TimelinePrevEditPoint

Command	Signature	Description	Example
SetLocation	setlocation(integer, float, float, float)	Locations are at prototype phase. Please do not use this command, it may change or might be removed.	SetLocation 1, 0,0,0
SetZoneMeshPointPos	setzonemeshpointpos(float, float, float, float, float)	This command provides control over the Free-Form Mesh component of the Projection Zones geo-correction system.	SetZoneMeshPointPos 1, 1,1,0,0
ClickScrollHue	clickscrollhue(float)	Hue slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollHue -5
ClickScrollHueShift	clickscrollhueshift(float)	Hue Shift slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollHueShift -5
ClickScrollSaturation	clickscrollsaturation(float)	Saturation slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollSaturation -5

Command	Signature	Description	Example
ClickScrollBeamBrush	clickscrollbeambush(float)	Beam brush slider of Live Control tab. The slider includes two buttons on the left and right sides that control its movement toward the boundaries. Use a negative step value to move toward the lower boundary and a positive value to move toward the upper boundary. A higher absolute value results in faster motion. Use zero to stop the movement.	ClickScrollBeamBrush -5
SetGridView		Go to Grid view (main form, mode/view buttons)	SetGridView
SetTimelineView		Go to Timeline view (main form, mode/view buttons)	SetTimelineView
SetPlayListView		Go to Play List view (main form, mode/view buttons)	SetPlayListView
SetUniverseView		Go to Universe view (main form, mode/view buttons)	SetUniverseView
UnselectAllProjectors		Command depreciated. Unselect all projectors.	UnselectAllProjectors
FXScrollDelta	fxscrolldelta(integer)	Shift FX table horizontal scroll position (left or right)	FXScrollDelta 8
SelectProjector		Command depreciated. Do not use.	SelectProjector 1 // index(es) (1..N) or name(string). Can be used more than one argument
UnselectProjector		Command depreciated. Unselect projectors by index or name.	UnselectProjector 1 // index(es) (1..N) or name(s)
ToggleSelectProjector		Command depreciated. Toggle selected/unselected state of all projectors.	ToggleSelectProjector 1 // index(es) (1..N) or name(s)
FocusProjector	focusprojector(integer), focusprojector(string)	Command depreciated. Do not use.	FocusProjector 1 // Projector index, 1..N.
StartTvMode	starttvmode(float)	Start TV output mode. TV Mode is global fixed FPS mode applied to all projectors.	StartTvMode 30 // 30 FPS
StopTvMode		Stop TV mode of BEYOND. This is companion command to StartTvMode.	StopTvMode
GroupCue		Enables the Group mode of the Grid. Group mode is a playback and cue start/stop mode that organizes cues into groups, visually represented by colors. Only one cue can play per group, although behavior also depends on the cue type—content, destination, and effects.	GroupCue

Command	Signature	Description	Example
Set4FxPerLine		Set QuickFX tab in extended four-effect per line mode.	Set4FxPerLine
SetDropFxMode		Set FX Table into “drop effect” mode, which is 3rd mode.	SetDropFxMode
Set1FxPerLine		Set QuickFX tab in standard one-effect per line mode.	Set1FxPerLine
PreviewAsUninverse	previewasuninverse(integer)	Command enable or disable Universe in Main Preview panel	PreviewAsUninverse 1
PreviewZoneGrid		Command under development/prototype phase.	PreviewZoneGrid
LockScreen		The LockScreen command places BEYOND into a locked state, preventing unauthorized access or unintended changes to the user interface.	LockScreen
UnLockScreen		Unlock the BEYOND UI. Used as pair for command LockScreen.	UnLockScreen
MoboNotify	mobonotify(string)	Send text message to all connected MoboLaser applications.	MoboNotify “Hello!”
SetUiFPS	setuifps(integer)	The command sets the frequency of the BEYOND UI repaint. The default frequency is relatively low—only 25 FPS. If you use BEYOND primarily for show design, it may be a good idea to increase the FPS to 50 or 60. Keep in mind that a higher FPS can increase CPU load and may slightly reduce keyboard responsiveness in the worst case. BEYOND is designed to handle dropped frames and will not lock up, but this is a reminder to maintain a balance and avoid overloading the main thread.	SetUiFPS 50

Command	Signature	Description	Example
Fb4DiscoveryMode	fb4discoverymode(integer), Notes:	Command set FB4 detection mode. By default BEYOND always searching for FB4 and establish connection as soon as it detected. However, it can be changed. Mode values: 0 - always. Default mode. 1 - auto. Check is there other BEYOND in the network, if yes then do not start discovery. If no then start search of hardware. 2 - manual. Enable or Disable discovery by command. Good option if you want to control it manually.	Fb4DiscoveryMode 0 // 0-always, 1-auto, 2-manual
StartFb4Discovery		Enable discovery of FB4 in the networks.	StartFb4Discovery 0
StopFb4Discovery		Disable discovery of FB4 in the networks.	StopFb4Discovery
SetVdgPlugin	setvdgplugin(integer)	This command enables, disables, or toggles the data-retrieval state of the VirtualDJ plugin.	SetVdgPlugin 2 // 0 disable, 1 enable, 2 toggle
SetCueCaptionColor	setcuecaptioncolor(float, float, float)	Set the cue caption background color	SetCueCaptionColor 1,2, 0x0000FF // page 1, cue 2, color (windows, 24bit color)
ToggleUMaxPage	toggleumaxpage(float), toggleumaxpage(string)	This command performs page navigation (UMAX window). A numeric argument triggers a relative action (go to the previous or next page). A string argument selects a page by its name.	ToggleUMaxPage 1 // next page
PulseEvent		Pulses one or more Events. Each parameter is a string containing the Event name to pulse. This command is used to trigger other scripts that are waiting for an Event using WaitForEvent.	PulseEvent "MyEventName" // OTHER SCRIPT WaitForEvent "MyEventName" DisplayPopup "It works!" restart
RefreshDmxIn	Notes:	Force BEYOND to regenerate commands for state update. Related to PangoScript to DMX and legacy FB3 style DMX IN.	RefreshDmxIn
DmxInMute	dmxinmute(float, float), dmxinmute(float, float, float)	BEYOND provides the ability to mute incoming DMX channels. The DmxInMute command allows you to mute or unmute one or multiple channels.	DmxInMute 1, 2048, 0 // from, to, state; DmxInMute 1, 2048, 1 // from, to, state; DmxInMute 100, 0 // channel, state

Command	Signature	Description	Example
ToggleUEditPage	toggleueditpage(float), toggleueditpage(string)	This command performs page navigation (Universe editor). A numeric argument triggers a relative action (go to the previous or next page). A string argument selects a page by its name.	ToggleUEditPage 1 // next page
ToggleUCenterPage	toggleucenterpage(float), toggleucenterpage(string)	This command performs page navigation (Universe docked to main window). A numeric argument triggers a relative action (go to the previous or next page). A string argument selects a page by its name.	ToggleUCenterPage 1 // next page
ToggleUToolPage	toggleutoolpage(float), toggleutoolpage(string)	This command performs page navigation (Universe docked into QuickTools area). A numeric argument triggers a relative action (go to the previous or next page). A string argument selects a page by its name.	ToggleUToolPage 1 // next page
ToggleURightPage	toggleurightpage(float), toggleurightpage(string)	This command performs page navigation (Universe docked into right side tabs). A numeric argument triggers a relative action (go to the previous or next page). A string argument selects a page by its name.	ToggleURightPage 1 // next page
ToggleUPreviewPage	toggleupreviewpage(float), toggleupreviewpage(string)	This command performs page navigation (Universe docked into main Preview panel). A numeric argument triggers a relative action (go to the previous or next page). A string argument selects a page by its name.	ToggleUPreviewPage 1 // next page
SetUEditPage	setueditpage(float), setueditpage(string)	This command selects a page of the Editor Universe panel, using either its index or its name.	SetUEditPage 1
SetUCenterPage	setucenterpage(float), setucenterpage(string)	This command selects a page of the Universe panel, located in the central area, using either its index or its name.	SetUCenterPage 1 // set first page
SetUToolPage	setutoolpage(float), setutoolpage(string)	This command selects a page of the Universe panel, located in the QuickTools area, using either its index or its name.	SetUToolPage 1

Command	Signature	Description	Example
SetURightPage	seturightpage(float), seturightpage(string)	This command selects a page of the Universe panel, located in the right side area, using either its index or its name.	SetURightPage 1
SetUPreviewPage	setuppreviewpage(float), setuppreviewpage(string)	This command selects a page of the Universe panel, located in the preview area, using either its index or its name.	SetUPreviewPage 1
SetUMaxPage	setumaxpage(float), setumaxpage(string)	This command selects a page of the Universe panel, located in the UMAX, using either its index or its name.	SetUMaxPage 1
ShutDownWindows		This command is intended for companies that operate permanent installations, where BEYOND needs to run on a schedule or be controlled remotely. The command requires the ShutDownWindows.exe application to function. On the BEYOND side, it simply calls ShellExecute().	ShutDownWindows // require ShutDownWindows.exe application
SelectCue	selectcue(string), selectcue(float, float), selectcue(float, float, float)	Controls the Selected property of a cue. You can identify a cue either by its Page/Cue indexes or by its name. BEYOND first searches for the cue name in the current page of the main Grid. If the cue is not found there, the search continues from the first page of the workspace.	SelectCue "Cue01" // Enter correct Cue name!
StartCue	startcue(integer, integer), startcue(string), startcue(string, string)	Starts playback of a Cue. The Cue can be specified either by its page and cue index, or by its name. In Multi-Cue mode, this command may start more than one Cue. In Single-Cue mode, only one Cue will be started.	StartCue 1,1
StopCue	stopcue(integer, integer), stopcue(string), stopcue(string, string)	Stops playback of the specified Cue.	StopCue 1,1
ToggleCue	togglecue(integer, integer), togglecue(string), togglecue(string, string), Notes:	Finds a cue by name, by page name and cue name, or by index. Once the cue is located, the command toggles its playback state (play/stop).	ToggleCue 1,1

Command	Signature	Description	Example
CaptureToClipboard		BEYOND can create a "Capture" content object from the currently playing cues and place it into the clipboard. The CaptureToClipboard command triggers the corresponding user interface action to perform this operation.	CaptureToClipboard
PasteToCue		Paste clipboard top object into current cue.	PasteToCue
UnselectAllZones	Notes:	Clears the selection state of all Projection Zones. After this command, no zones remain selected.	UnselectAllZones
SelectZone		The command accepts multiple arguments. A string is treated as a zone name, and a numeric value is treated as a zone index. You can specify multiple names or indexes in any order. The command adds each Projection Zone to the current Master Selection Array. The order of the zones in this array is important for time-shift operations.	SelectZone 1,2,3, "Scanner 2 main"
UnSelectZone		Removes the Selected state from specified Projection Zone. Command accept multiple arguments - strings as zone name, and numbers as zone index.	UnSelectZone 1 // index or names
ToggleSelectZone		Toggles the Selected state of the specified Projection Zone. The command accepts multiple arguments: strings are treated as Zone names, and numeric values are treated as Zone indexes.	ToggleSelectZone 1
SelectZoneName		Command depreciated (act same as SelecteZone).	SelectZoneName "Main Graphics"
UnSelectZoneName		Removes the muted state from specified Projection Zone. Command accept multiple arguments - strings as zone name, and numbers as zone index.	UnSelectZoneName "Main Graphics", "Scanner 1 Main"
MuteSelectedZones	muteselectdzones(integer)	Sets the muted state for the selected Projection Zones only.	MuteSelectedZones OFF

Command	Signature	Description	Example
FocusZone	focuszone(integer), focuszone(string)	Put zone in focus. Related to QuickFX tab and Effect tab.	FocusZone 1
SelectAndFocusZone	selectandfocuszone(integer), selectandfocuszone(string)	Find Projection zone by name or index, make it selected and update Effect and QuickFX tabs.	SelectAndFocusZone 1
MuteZonesOfProjector		Mute zones of one or many specified projectors. Use indexes of the projector.	MuteZonesOfProjector 1
UnMuteZonesOfProjector		Unmute zones of one or many specified projectors. Use indexes of the projector.	UnMuteZonesOfProjector 1,2,3 // projector 1,2 and 3
SetActiveGrid	setactivegrid(float)	Sets the active Grid to either the first (main) or second (secondary) Grid. The active Grid concept is used in many parts of BEYOND. In practice, the rule is simple: the last Grid you interacted with becomes the active one.	SetActiveGrid 1 // 1 or 2
StartTalkClient		Start Talk client - UDP based script messaging. The client send messages.	StartUdpTalkClient
StopTalkClient		Stop Talk client - UDP based script messaging. The client send messages.	StopUdpTalkClient
StartTalkServer		Start Talk server - UDP based script messaging. The server receive messages.	StartUdpTalkServer
StopTalkServer		Stop Talk server - UDP based script messaging. The server receive messages.	StopUdpTalkServer
SelectNextCatGrid1	Notes:	Go to the next category in the workspace. The command is always associated with the Main Grid.	SelectNextCatGrid1
SelectPrevCatGrid1		Go to the previous category in the workspace. The command is always associated with the Main Grid.	SelectPrevCatGrid1
SelectNextPageGrid1	Notes:	Go to the next page in the workspace. The command is always associated with the Main Grid.	SelectNextPageGrid1
SelectPrevPageGrid1	Notes:	Select previous Page of the Grid. The command is always associated with the Main Grid.	SelectPrevPageGrid1

Command	Signature	Description	Example
SelectNextTabGrid1	Notes:	Go to the next tab in the workspace. The command is always associated with the Main Grid.	SelectNextTabGrid1
SelectPrevTabGrid1	Notes:	Select previous Tab of the Grid. The command is always associated with the Main Grid.	SelectPrevTabGrid1
RebootConnectedFB4		Send Reboot command to all connect FB4\.. Might help of you want to reboot the hardware remotely. Also might be required if you just uploaded firmware on the SD card and need to start update process.	RebootConnectedFB4
UnselectAllCue		Clears the selection state of all workspace cues. After this command, no cues remain selected.	UnselectAllCue
Pub	pub(string, string), pub(string, string, integer), pub(string, string, integer, integer), pub(string, float)	Internal	pub "data/", "test text"
SubProp	subprop(string, string)	Internal	
SubJson	subjson(string, string)	Internal	
SubCmd	subcmd(string)	Internal	
PubObject	pubobject(string, string), pubobject(string, string, string)	Internal	
Chat	chat(string), chat(string)	Internal Prototype stage command. Under construction. Do not use.	
SetProp	setprop(string, string), setprop(string, integer), setprop(string, float)	Set the property value. Idea - universal setter, where you can use string variable with property name and second with property value.	SetProp "Master.bpm", 123
StartCueMulti	startcuemulti(integer, integer), startcuemulti(string), startcuemulti(string, string)	Starts playback of a Cue. The Cue can be specified either by its page and cue index, or by its name. Command always in multi-cue mode. You are responsible to stop the cue.	StartCueMulti 1,1
ToggleCueMulti	togglecuemulti(integer, integer), togglecuemulti(string), togglecuemulti(string, string)	Finds a cue by name, by page name and cue name, or by index. Once the cue is located, the command toggles its playback state (play/stop). Allows multiple cues to be started simultaneously, which may lead to system overload if used excessively.	ToggleCueMulti 1,1

Command	Signature	Description	Example
SetLimiterProfile	setlimiterprofile(integer), Notes:	Choose between 3 limiter profiles	SetLimiterProfile 1
SetLimiterPerZone	setlimiterperzone(integer), Notes:, Warnings:	This command belongs to the SetLimiter... group. Live/Grid playback uses a dedicated pool of cue players, which operates under specific rules such as limiters that define the maximum number of simultaneously playing cues by type.	SetLimiterPerZone 3 // do not use this
SetLimiterPerGrid	setlimiterpergrid(integer), Notes:	This command belongs to the SetLimiter... group. Live/Grid playback uses a dedicated pool of cue players, which operates under specific rules such as limiters that define the maximum number of simultaneously playing cues by type. SetLimitPerGrid defined total limit.	SetLimiterPerGrid 10
SetLimiterFlash	setlimiterflash(integer), Notes:	This command belongs to the SetLimiter... group. Live/Grid playback uses a dedicated pool of cue players, which operates under specific rules such as limiters that define the maximum number of simultaneously playing cues by type. SetLimiterFlash defines max number of the simultaneously playing cues in FLash mode.	SetLimiterFlash 4
SetLimiterHold	setlimiterhold(integer), Notes:	This command belongs to the SetLimiter... group. Live/Grid playback uses a dedicated pool of cue players, which operates under specific rules such as limiters that define the maximum number of simultaneously playing cues by type.	SetLimiterHold 3

Command	Signature	Description	Example
SetLimiterBeam	setlimiterbeam(integer), Notes:	This command belongs to the SetLimiter... group. Live/Grid playback uses a dedicated pool of cue players, which operates under specific rules such as limiters that define the maximum number of simultaneously playing cues by type. SetLimiterBeam defines max number of the simultaneously playing QuickBeam-cues.	SetLimiterBeam 5
SetLimiterDMX	setlimiterdmx(integer), Notes:	This command belongs to the SetLimiter... group. Live/Grid playback uses a dedicated pool of cue players, which operates under specific rules such as limiters that define the maximum number of simultaneously playing cues by type. SetLimiterDMX defines max number of the simultaneously playing DMX-cues.	SetLimiterDMX 5
SetLimiterShow	setlimitershow(integer), Notes:	This command belongs to the SetLimiter... group. Live/Grid playback uses a dedicated pool of cue players, which operates under specific rules such as limiters that define the maximum number of simultaneously playing cues by type. SetLimiterShow defines max number of the simultaneously playing timelines-cues.	SetLimiterShow 2
QLog		Displays a message in the Quick Log panel. The command accepts any number of arguments and concatenates them into a single string without separators. The message is added as an Info-level entry.	QLog "Hello World!"
LogInfo		Sends an Info message to the QuickLog tab. Any number of parameters can be provided—they will be concatenated into a single string.	LogInfo "My message to QLog tab"

Command	Signature	Description	Example
LogWarning		Sends a Warning message to the QuickLog tab. Any number of parameters can be provided—they will be concatenated into a single string.	LogWarning “My message to QLog tab”
LogError		Sends an Error message to the QuickLog tab. Any number of parameters can be provided—they will be concatenated into a single string.	LogError “My message to QLog tab”
AddSms	addsms(string), addsms(string, string), addsms(string, string, string), addsms(string, string, string, integer)	Command adds a new message to the SMS FIFO queue for display on the laser projector. You can view the message FIFO in the QuickSMS tab, located at the bottom of the main window. This module was originally designed for compatibility with the Pangolin IMU software, which is now discontinued. However, the ability to add messages through scripting allows connectivity with various external input sources.	AddSMS “This is my demo message”; AddSMS “Red text message”, “”, “”, 0x0000FF
ShowItNowSMS	showitnowsms(integer)	Enable/disable laser output from QuickSMS tab.	ShowItNowSMS 1
ExitBEYOND		The ExitBEYOND command closes the BEYOND application, equivalent to clicking the close button on the main window. Execution of this command is asynchronous.	ExitBEYOND
RegisterOscFeedback	registeroscfeedback(string, string)	This command allows monitoring the state of a variable and sending an OSC message when the value of an object property changes. Implementation details: BEYOND maintains a dedicated list of property names to watch. Their values are checked periodically by a timer. When a value changes, BEYOND sends an OSC message automatically. This approach is somewhat CPU-intensive, but it is universal and works with all object properties accessible in PangoScript.	RegisterOscFeedback “/beyond/master/brightness”, “master.brightness”

Command	Signature	Description	Example
SetMassSlider	setmassslider(float)	Command control Physics slider (mass) of Master tab.	SetMassSlider 4
SetAttractionSlider	setattractionslider(float)	Command control Physics slider (attraction) of Master tab.	SetAttractionSlider 10
SetFrictionSlider	setfrictionslider(float)	Command control Physics slider (Friction) of Master tab.	SetFrictionSlider 10
SetPhysicsCheckbox	setphysicscheckbox(integer)	This command enables, disables, or toggles the Physics checkbox located in the Live Control tab.	SetPhysicsCheckbox TOGGLE
ShowMasterHelpFile		Show BEYOND help file. Legacy command.	ShowMasterHelpFile
SetAudioInGain	setaudioingain(float)	FFT Audio compensation (1 to 15). Is not related to audio playback. Gain on input of FFT.	SetAudioInGain 10
SetAudioInRelease	setaudioinrelease(float)	FFT analyzer parameter - peak release speed, decrease intense jumping of the value up and down.	SetAudioInRelease 75
ShowHint	showhint(string)	This command displays a popup window with hint text. It is useful for showing which slider is currently in motion and its value. This type of popup is used by the QuickShow MIDI-mapping system, and BEYOND allows you to use the same feature for any slider you are mapping.	ShowHint "Hello!"
LoadZoneFromBlob	loadzonefromblob(integer, boolean)	Internal Load workspace from BLOB. This is HTTP oriented command, do not use in script directly.	<internal command>
LoadCueFromBlob	loadcuefromblob(integer, integer, boolean), Warnings:	Internal Command designed for special project, do not use in PangoScript.	<internal command>

Recording

Command	Signature	Description	Example
StartQuickRecord		Start recorder located in QuickRecord tab. Feature is not publicly available yet.	StartQuickRecord
StopQuickRecord		Stop recorder located in QuickRecord tab. Feature is not publicly available yet.	StopQuickRecord

Command	Signature	Description	Example
SetRecordFile	setrecordfile(string)	Command sets the file name for the recording. The recording feature is in not released state yet.	SetRecordFile "C:\BEYOND50\MyFile.best"

Wait & Triggers

Command	Signature	Description	Example
Sleep	sleep(integer), Notes:	The Sleep instruction puts the script into a waiting state for the specified number of milliseconds.	Sleep 1000
WaitForMidi	waitformidi(integer, integer, integer), Warnings:	WaitForMIDI instruction put scripiter into waiting state for MIDI message param(1) - first byte of MIDI message (msg) param(2) - second byte of MIDI message (data1) param(3) - third byte of MIDI message (data2) Data2 also stored as extvalue in normalized form	WaitForMidi 0x90, -1, -1 DislayPopup param(1), param(2), param(3) restart
WaitForDmx	waitfordmx(integer), Warnings:	param(1) - DMX channel index param(2) - DMX channel value	WaitForDmx 1 DisplayPopup param(2) restart
WaitForChannel	waitforchannel(integer), Warnings:	WaitForChannel put script into waiting state for channel value. param(1) - channel index param(2) - channel value	WaitForChannel -1 displaypopup param(1), param(2) restart

Command	Signature	Description	Example
WaitForCueStart	waitforcuestart(integer, integer), Warnings:	The WaitForCueStart command puts the script into a waiting state until a cue playback starts. This command is not related to UI cell interaction. The hook is placed in the playback engine that manages player allocation. If you click on an empty grid cell to start a new cue, this waiting operation will not respond. When triggered, the fired cue and page indexes are stored as script parameters. Use param(1) to get the page index and param(2) to get the cue index. Note that BEYOND uses zero-based indexing internally, so both param(1) and param(2) return zero-based values.	WaitForCueStart -1,-1 displaypopup param(1)+1, param(2)+1 restart
WaitForCueStop	waitforcuestop(integer, integer), Warnings:	The WaitForCueStop command puts the script into a waiting state until a cue playback stops. The hook is placed in the playback engine that manages player allocation. When triggered, the stopped cue and page indexes are stored as script parameters. Use param(1) to get the page index and param(2) to get the cue index. Note that BEYOND uses zero-based indexing internally, so both param(1) and param(2) return zero-based values.	WaitForCueStop 1, 1 //page 1, cue 1 displaypopup "stopped..." restart

Command	Signature	Description	Example
WaitForCellDown	waitforcelldown(integer), Notes:, Warnings:	Wait for click (cell down) on the content grid. Use -1 to wait for any click (cell down) on the grid. Otherwise specify exact cell index. function Param(1) returns zero based cell index	// run this script in PangoScript tab // click on left/top cell of content grid. waitforcelldown 1 displaypopup "cell down" restart; // run this script in PangoScript tab waitforcelldow -1 // -1 means any cell displaypopup param(1)+1 // index is zero based internally restart
WaitForCellUp	waitforcellup(integer), Notes:, Warnings:	Wait for click release (cell up) on the content grid. Use -1 to wait for any click (cell up) on the grid. Otherwise specify exact cell index. function Param(1) returns zero based cell index	// run this script in PangoScript tab // click on left/top cell of content grid. waitforcellup 1 displaypopup "cell down" restart; // run this script in PangoScript tab waitforcellup -1 displaypopup param(1)+1 // index is zero based internally restart
WaitForTime	waitfortime(integer, integer, integer, integer), Warnings:	The WaitForTime command puts the scripter into a waiting state until the specified time is reached. Once the system clock passes the defined time, the waiting ends and the next command is executed.	// NOTE: THIS IS NOT GOOD FOR SCHEDULER. // at 15:30 WaitForTime 15,30,0,0 // hours, minutes, seconds, ms StartCue 1,1 // at 15:40 WaitForTime 15,40,0,0 // hours, minutes, seconds, ms StartCue 1,2

Command	Signature	Description	Example
WaitForTimePos	waitfortimepos(integer, integer, integer, integer), Warnings:	This wait command wait for time passed from script execution. This is relative time command. Restart command reset the local script time.	// beginning WaitForTimePos 0,0,0,0 StartCue 1,1 // minute later WaitForTimePos 0,1,0,0 StartCue 1,2 // in 2 minutes WaitForTimePos 0,2,0,0 StartCue 1,3
WaitForHotKey	Warnings:	Each scripiter has hotkey. This waiting command allows organize waiting for the hotkey press and doing some action after it/	WaitForHotKey DmxOut 1, 255 Sleep 1000 DmxOut 1, 0 restart
WaitForAudioBeat	waitforaudiobeat(integer), Warnings:	Put Scripiter into waiting state for the audio-beat event	WaitForAudioBeat 1 // waiting instruction, every beat master.color = randomin(32,255) // example of action - random color of master lc. restart // jump to script start, and immediatelly run into waiting state
WaitForManualBeat	waitformanualbeat(integer), Warnings:	Put Scripiter into waiting state for the manual-beat event	WaitForManualBeat 1 // every manual beat displaypopup "Manual beat!" // ui message as an action restart // jump to script start, and immediatelly run into waiting state
WaitForTimerBeat	waitfortimerbeat(integer), Warnings:	Put Scripiter into waiting state for the timer-beat event. Best approach for waiting scripts - put them into PangoScript tab, and use structure like: WaitFor... // your action, executes once waiting completed. Restart	WaitForTimerBeat 4 // waiting instruction master.brightness = randomin(0,100) // example of action - random brightness restart // jump to script start, and immediatelly run into waiting state

Command	Signature	Description	Example
WaitForBeat	waitforbeat(integer, integer)	Put Scripter into waiting state . WaitForBeat instruction allows to wait for beat defined by the mask parameter.	WaitForBeat 7, 1 // any and every of beats displaypopup "Boom!" // ui message as an action restart // jump to script start, and immediately run into waiting state
WaitForEvent	Warnings:	WaitForEvent is a general-purpose waiting command that works in conjunction with the PulseEvent command. It must include at least one event name, defined as a string of your choice. Multiple event names can be specified. As soon as any of the listed events is pulsed, the script resumes execution. The main purpose of this command pair is to enable coordination and interaction between multiple scripts. The syntax: WaitForEvent "Event1" WaitForEvent "Event1", "Event2", "Event3"	// script 1: WaitForEvent "MyEventName" DisplayPopup "MyEventName pulsed" restart // script 2 PulseEvent "MyEventName"; // script 1: codename "DrumAction" var a a = 0 LoopPoint: WaitForEvent "Kick", "Snare" DisplayPopup "Boom!" goto LoopPoint // script 2 DrumAction.a = 100 PulseEvent "Kick" // script 3 DrumAction.a = 200 PulseEvent "Snare" PulseEvent "Snare"

Command	Signature	Description	Example
WaitForTC	waitfortc(integer, integer, integer, integer), Warnings:	The WaitForTC command puts the scripter into a waiting state until the specified TIMECODE time is reached. Once the incoming timecode passes the defined time, the waiting ends and the next command is executed. param(1) has current timecode time in seconds.	// NOTE: THIS IS NOT TIMELINE. THIS JUST SEQUENTIAL WAITING // at 00:00:00:000 WaitForTC 0,0,0,0 // hours, minutes, seconds, ms StartCue 1,1 // at 00:00:30:000 WaitForTime 0,0,30,0 // hours, minutes, seconds, ms StartCue 1,2
WaitForPageChange	Warnings:	WaitForPageChange command put into waiting state, which ends when main content grid page is changed. It does not applicable to secondary grid. param(1) returns current page index (zero based)	WaitForPageChange

12. Import into other tools

Attach or download openapi.json (OpenAPI 3.0.3) and import it into Postman, Insomnia, Bruno, or a code generator. The same file ships in <exe>\www\beyonddswagger\openapi.json.

Example pages (cue grid, playlist, live control) ship in the installer www folder.

13. Common mistakes

- Using 127.0.0.1 from another device. Use the BEYOND PC LAN IP.
- Calling paths without the /apiv1/ prefix.
- Sending {"code": "..."} to execscript instead of raw text/plain.
- Treating + as a space on ps/cmd. Use %20.
- POSTing ugc or preview to /apiv1/zone — that returns 400. Use execscript.
- Leaving static Root blank but expecting files next to BEYOND.exe instead of in <exe>\www. The 5.8 installer fills www.

From:

<https://wiki.pangolin.com/> - **Complete Help Docs**

Permanent link:

https://wiki.pangolin.com/doku.php?id=beyond:http_server&rev=1790685747

Last update: **2026/09/29 14:42**

