nativeKONTROL ClyphX Pro M4L Interface Documentation

Version 1.1.4

(for use with ClyphX Pro v1.1.4 or later)

1 CHANGES IN THIS VERSION

This is the initial version of this documentation and so there are no changes to report at this time.

2 INTRODUCTION

This document describes the interface that ClyphX Pro exposes for use in Max for Live. The relevant ClyphX Pro Functions and Properties as well as the relevant functions and properties of its Controls and Components are all documented here. Note, however, that we do not cover every function, property, control and component of ClyphX Pro that is accessible via Max for Live as many are simply not relevant, only applicable in very specialized/advanced use-cases or are subject to potentially breaking changes. While you can certainly use those undocumented aspects of ClyphX Pro if you like, it is not something that we can provide support or documentation on.

In addition to referring to this document, it's recommended that you take a look at the Max for Live devices provided in the ClyphX Pro Live Lessons to see some practical examples of what is described in this document.

IMPORTANT NOTE: Technically speaking, ClyphX Pro is not a Control Surface Script and so you should not try to it access it through the LOM like you would a Control Surface Script. Instead, you should use the `clyphx_pro_interface` Javascript file that we provide in ClyphX Pro's Max for Live Integration Lesson.

Abbreviated Terms

The tables in this document use the following abbreviated terms:

`id` – The ID of an object that has functions and/or properties.

`int` – An integer (whole number). Unless otherwise noted, the interface uses 0-based numbering. So, for example, when specifying or getting the index of an item, 0 is the first item, 1 is the second and so on.

`sym` – A symbol (one or more words). In Max, a single word on its own is a symbol. Multiple words separated by space would be considered as multiple symbols unless they are enclosed in quotes. For example, "I'm a single symbol".

Arguments

The tables in this document that describe functions include an ARGUMENT(S) column that describes the argument(s) that the functions accept (if any). If more than one argument is accepted, the arguments will be described in a bulleted list. The name of the function indicates the type of argument(s) it accepts. For example, `some_function sym` accepts a single symbol argument.

Some functions may accept optional arguments. These are arguments that do not have to be passed to the function. Optional arguments are indicated by square brackets. For example, `some_function int [sym]` accepts an integer argument and an optional symbol argument. So you could call it like this:

```
some_function 5
```

Or like this:

```
some_function 5 hello
```

Return

The tables in this document that describe functions and properties include a RETURN column that indicates the type of value(s) the functions or properties return (if any) and a DESCRIPTION column that describes the returned value(s). If more than one value is returned, the values will be indicated/described in bulleted lists.

3 FUNCTIONS

The following table covers the top-level functions of ClyphX Pro.

NAME	ARGUMENT(S)	RETURN	DESCRIPTION
`get_binding_component int`	The index of the BindingComponent to return.	`id`	Returns the ID of the BindingComponent for the given index. Note that ClyphX Pro as well as any active XT Scripts will have their own BindingComponent. Index range: 0 to 5
`get_device_manager_for_track_path sym`	The LOM path to the Track.	`id`	Returns the ID of the DeviceManagerComponent for the Track.
`trigger_action_list sym`	The Action List to trigger.	`-`	Triggers the given Action List. Typically, you should not call this directly. Instead, simply pass the Action List to the clyphx_pro_interface so that it can handle the heavy lifting.

4 PROPERTIES

The following table covers the top-level properties of ClyphX Pro. These simply return a value or list of values. They do not provide set or observation access.

NAME	ARGUMENT(S)	RETURN	DESCRIPTION
`num_binding_components`	-	`int`	Returns the number of active BindingComponents. Note that ClyphX Pro as well as any active XT Scripts will have their own BindingComponent.

5 CONTROLS/ELEMENTS

When used with the ClyphX Pro Bindings Accessory, ClyphX Pro uses two types of elements to represent physical controls; **BindableButtonElements** and **BindableEncoderElements**, which are collectively referred to as bindable elements. Through these elements, you can address physical controls. The following table covers the properties of these elements, which can be observed.

NAME	ARGUMENT(S)	RETURN	DESCRIPTION
`parameter`	-	`id`	Returns the ID of the parameter the element is bound to.
`parameter_name`	-	`sym`	Returns the name of the parameter the element is bound to. When observed, this will send notifications if/when the name of the parameter changes.
`parameter_value`	-	`sym`	Returns the value (as a symbol) of the parameter the element is assigned to. This is meant for display purposes.

6 COMPONENTS

ClyphX Pro is composed of a variety of components. This section covers some of the core components of ClyphX Pro that are relevant when creating devices for use with ClyphX Pro.

BindingComponent

BindingComponent is the primary component involved in bindings. Through this component, you can access elements and parameters involved in bindings.

Functions

NAME	ARGUMENT(S)	RETURN	DESCRIPTION
`last_tweaked_parameter`	-	`id`	Returns the ID of the last parameter that was adjusted via a bindable element.
`get_button int`	The index of the BindableButtonElement to return.	`id`	Returns the ID of the BindableButtonElement for the given index. Index range: 0 to num_buttons - 1
`get_encoder int`	The index of the BindableEncoderElement to return.	`id`	Returns the ID of the BindableEncoderElement for the given index. Index range: 0 to num_encoders - 1

Properties

These simply return a value or list of values. They do not provide set or observation access.

NAME	ARGUMENT(S)	RETURN	DESCRIPTION
`num_buttons`	-	`int`	Returns the number of active BindableButtonElements.
`num_encoders`	-	`int`	Returns the number of active BindableEncoderElements.

DeviceManagerComponent

DeviceManagerComponent manages all of the Devices (nested and non-nested) on a Track. ClyphX Pro provides one of these components for each Track in a Set. The point of these component is to provide notifications when any changes occur to Devices on the Track at any nesting level. So, it will notify when Devices/Chains are added/removed or renamed.

The following properties are available. Aside from `devices`, which can be observed, the properties listed here just return a value or list of values. They do not provide set or observation access.

NAME	ARGUMENT(S)	RETURN	DESCRIPTION
`all_devices`	-	`list of id`	Returns the list of IDs of all of the Devices on the Track.
`devices`	-	-	Notifies observers any time Devices or Chains are added to or removed from the Track and also any time Device or Chain names change.
`rack_devices`	-	`list of id`	Returns the list of IDs of all of the Rack Devices on the Track.
`track`	-	`id`	Returns the ID of the Track that is managed by this component.