

# Live Object Model Changes - Live 8 -> Live 9

This document only reflect the state of the LOM at the time of writing - for up-to-date documentation please refer to the LOM documentation included in Max for Live.

## Application

- *open\_dialog\_count* - Get/listen to the number of dialog boxes shown.
- *current\_dialog\_message* - Text of the current message box; empty if none is shown at the moment.
- *current\_dialog\_button\_count* - Number of buttons on the current message box.
- *press\_current\_dialog\_button(index)* - Press the button with the given index on the current dialog box.

## Application.View

- *scroll\_view* for the "Browser" view
- *toggle\_browse()* - Reveals the device chain, the browser and starts hot swap for the selected device. Calling this function again stops hot swap.
- *browse\_mode* - True if HotSwap mode is active for any target

## Song

- *create\_scene(index)* - Index determines where the scene is added; it is only valid between 0 and len(song.scenes); -1 will add the new scene at the end of the list
- *swing\_amount* - 0.0 - 1.0; right now only affects midi recording quantization and direct calls to Clip.quantize(...).
- *capture\_and\_insert\_scene()* - Optional argument that can either be 'Song.CaptureMode.all' or 'Song.CaptureMode.all\_except\_selected'; the latter will capture all playing clips except for the selected one, leaving the slot of the selected track free in the new scene.
- *create\_midi\_track(index)*
- *create\_audio\_track(index)*
- *duplicate\_scene(index)*
- *duplicate\_track(index)*
- *session\_record* - Get/set/listen to the state of the session overdub button.
- *arrangement\_overdub* - Get/set/listen to the state of the arrangement overdub button.
- *session\_record\_status* - An integer that changes when the state of Live's new session recording button changes. Can be observed but not set.
- *create\_return\_track()* - Adds a new Return track at the end.

- *delete\_track(track\_index)* - Delete the track in the 'tracks' property at the given index
- *stop\_all\_clips* now has an optional argument 'quantized' (default is true). Calling the function with false will stop all clips immediately, independent of the launch quantization.
- *session\_automation\_record* - Get/set/listen to the state of the new Automation Arm button in Live
- *re\_enable\_automation\_enabled* - Get/listen to the state of the 'Back to Session Automation' button.
- *re\_enable\_automation()* - Trigger 'Back to Session Automation', re-activating automation in all running session clips.
- *delete\_scene(index)* - Deletes the scene at the given index. Raises an error when there is no scene at given index or the index is invalid.
- *trigger\_session\_record* (optional: *record\_length*) - Starts recording in either the selected slot or the next empty slot, if track is armed. If *record\_length* is provided, the slot will record for the given length in beats. If triggered while recording, it will stop recording and start to play back the clip.

## Track

- *delete\_device(device\_index)* - Delete the device in the 'devices' property at the given index
- *duplicate\_clip\_slot(index)* - Works like 'Duplicate' in a clip's context menu;

## Track.View

- *device\_insert\_mode* - Get/set/listen to where a device will be inserted when loaded from the browser. Can be 'default' (add device at the end), 'selected\_left' (add device to the left of the selected device), or 'selected\_right' (add device to the right of the selected device).

## ClipSlot

- *create\_clip(length)* - Length is given in beats, must be bigger than 0.0; can only be called on empty clip slots in MIDI tracks.
- *delete\_clip()* - Deletes the contained clip
- *fire(record\_length, launch\_quantization)* - The resulting clip will be re-fired after the given *record\_length*; *launch\_quantization* overrides the global quantization if provided.
- *is\_playing, is\_recording, will\_record\_on\_start* will now return the respective state of the contained clip if there is one (they used to throw errors in that case)

## Clip

**Reminder:** Note format is (pitch, time, duration, velocity, is\_muted)

- *get\_notes*(from\_time, from\_pitch, time\_span, pitch\_span) - Returns a list of notes that start in the given area just like *get\_selected\_notes*
- *remove\_notes*(from\_time, from\_pitch, time\_span, pitch\_span) - Deletes all notes that start in the given area
- *set\_notes*(list\_of\_notes) - Will apply the given notes to the clip, overwriting existing notes  
**Note:** Accessing these properties if *is\_midi\_clip* is false will generate an error
- *duplicate\_loop* - Makes the loop two times longer by moving *loop\_end* to the right, and duplicates both the notes and the envelopes. If the clip is not looped, the clip start/end range is duplicated. The function has no parameters. Available for MIDI clips only.
- *is\_overdubbing* - Get/listen to whether the clip is overdubbing
- *file\_path* - Get the location of the audio file represented by the clip **Note:** Accessing this property will generate an error if *is\_audio\_clip* is false
- *quantize*(quantization\_grid, amount) - Quantizes all notes in the clip to the *quantization\_grid* (one of *Live.Song.RecordingQuantization*) taking the song's *swing\_amount* into account; throws a runtime error if called on an audio clip.
- *start\_marker* - Get/set/listen to the clip's start marker in beats, independent of the loop state. Raises an error if set behind the end marker.
- *start\_time* - The time the clip has been started, relative to the global song time, the value is in beats
- *end\_marker* - Get/set/listen to the clip's end marker in beats, independent of the loop state. Raises an error if set before the start marker.
- *gain* - Get/set/listen to the clip's gain setting; range is 0.0 to 1.0. Raises an error if called on a MIDI clip.
- *gain\_display\_string* - Get the clip's gain display value as a string (e.g., "1.3 dB"). Raises an error if called on a MIDI clip.
- *available\_warp\_modes* - Returns the list of indexes of the warp modes available for the clip; Raises an error if called on a MIDI clip.
- *warp\_mode* - Get/set/listen to the clip's warp mode as an integer index. Raises an error if called on a MIDI clip. **Note:** Available warp modes are 0(Beats), 1(Tones), 2(Texture), 3(Repitch), 4(Complex), 5(Rex), 6(ComplexPro).
- *clear\_envelope*(device\_parameter) - Removes the clip's automation for the given parameter.
- *clear\_all\_envelopes* - Removes all automation in the clip.
- *has\_envelopes* - Get/listen to whether the clip has any automation.
- *quantize\_pitch*(pitch, quantization\_grid, amount) - Same as *quantize*(*quantization\_grid*, *amount*), but only for notes in the given pitch.

## Clip.View

- *grid\_is\_triplet* - Get/set whether the clip is displayed with a triplet grid
- *grid\_quantization* - The grid quantization; possible values can be found in `Live.Clip.GridQuantization`
- *show\_loop* - If the clip is visible in Live's detail view, this function will make the current loop visible there

## Device

- *can\_have\_drum\_pads* (only true for drum racks)
- *drum\_pads* - Read-/listenable list of all 128 DrumPads for the top most drum rack, inner drum racks return a list of 0 entries).
- *visible\_drum\_pads* - Read-/listenable list of all 16 visible DrumPads for the top most drum rack, inner drum racks return a list of 0 entries).
- *class\_display\_name* - Get the device's original name (e.g., Operator, Auto Filter).
- *type* - The device's type. Possible types are 0(undefined), 1(instrument), 2(audio\_effect), 3(midi\_effect).

## Device.View (read-only, of type `Live.Device.Device.View`)

- *drum\_pads\_scroll\_position* (int) - Lowest row of pads visible, range: 0 - 28 **Note:** Accessing this property will generate an error if *can\_have\_drum\_pads* is false or the drum rack is nested
- *selected\_chain* - The currently selected Chain (only available in instrument racks)
- *selected\_drum\_pad* - The currently selected DrumPad (only available in drum racks)

## DrumPad

- *chains* - Read-/listenable list of contained chains
- *name* - Read/listenable
- *note* - Read-only
- *mute* - Read-/write-/listenable
- *solo* - Read-/write-/listenable, exclusivity needs to be maintained manually, same as in tracks
- *name* - Read-/listenable

## Chain

- *delete\_device(device\_index)* - Delete the device in the 'devices' property at the given index.
- Added *has\_audio\_input*, *has\_audio\_output*, *has\_midi\_input*, and *has\_midi\_output* all read-only.

## Scene

- *fire* and *fire\_as\_selected* now have an optional argument 'force\_legato' (default is false). Calling these functions with true will launch all clips immediately in legato, independent of their launch mode.
- *fire(force\_legato, can\_select\_scene\_on\_launch)* now has a second parameter *can\_select\_scene\_on\_launch* (default is True). If set to False/0, the scene is fired without selecting it.
- *is\_empty* - Is true if none of the slots in the scene is filled.

## Push

The Push control surface now has functions that allow Max for Live to use its controls more easily.

- *get\_control\_names()* - Returns the list of alls controls' names.
- *get\_control(name)* - Returns the control with the given name.
- *grab\_control(control)* - Releases all Push functionality from the control so that it can be used exclusively from Max for Live.
- *release\_control(control)* - Re-establishes the Push functionality for the control.