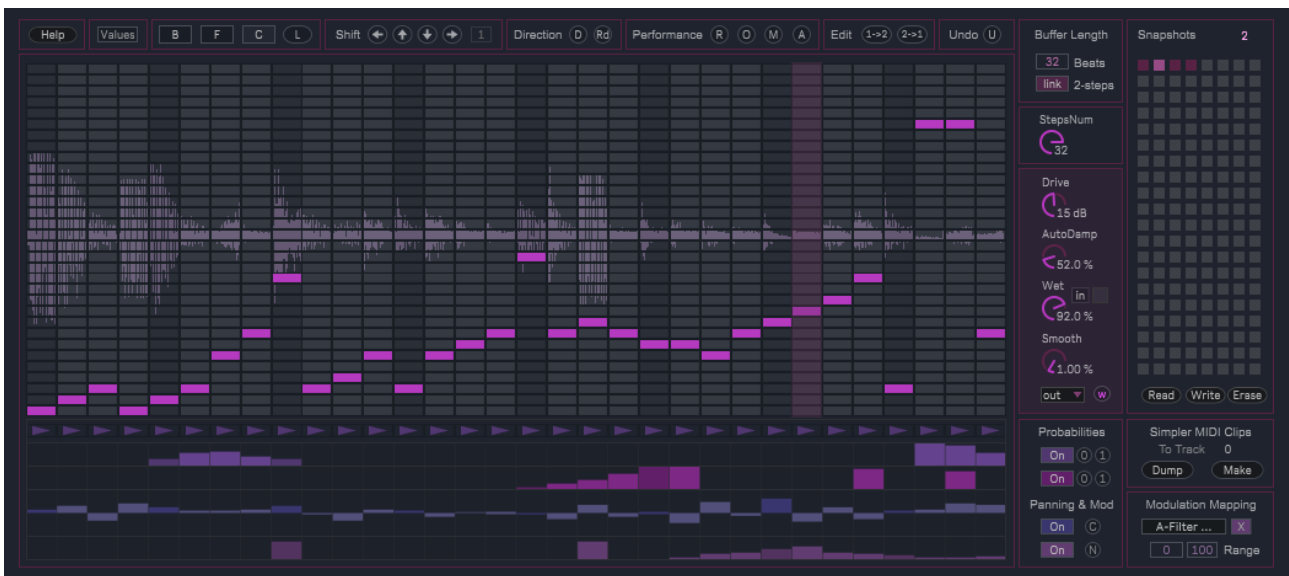


J74 SliceShuffler

An audio slice sequencer effect for Ableton Live



User Manual

How to Install the J74 SliceShuffler device

J74 SliceShuffler is an *audio effect* for Ableton Live. It requires a Max for Live license to run (as in Ableton Live Suite) and it will run properly only in Ableton Live 10 or higher (older versions of Live, such as Live 8 or Live 9, are not supported).

Installation of the tool is extremely easy: unzip the download pack and copy the files to your computer hard drive, any place you like. To start it up, drop the plug-in device (.amxd) on an audio (or instrument) track in Ableton Live.

J74 SliceShuffler overview

J74 SliceShuffler is an audio “slice sequencer” effect. The plugin works as an audio effect (audio in, audio out) and can be used on any track which generates or receives audio (an audio track or, eventually, an instrument track). The SliceShuffler device operates by buffering incoming audio (in the background, continuously) and slicing it for re-sequencing and manipulating of the audio slices. The device is made with rhythmic material in mind (e.g. drum loops or sequences), but can be also used with anything else (vocals or guitars for instance) to add a rhythmic feel.

SliceShuffler: slice sequencing

SliceShuffler allows free re-sequencing of the captured audio buffer, so that any slice can be played at any step. This allows you to remix the original phrase of the incoming audio with new time ordering. Moreover, sequencing also offers the possibilities of inserting probability based variations, so that a different slice “may” be randomly selected (or muted) with a certain probability. Sequencing is enhanced by useful editing & performance functions which are available for quick editing and performing (including a very useful performance-oriented undo function).

SliceShuffler: audio mangling and looping

The device processes also the slices sound. Each sequenced-slice can be independently reversed, muted and/or panned. Like for slice order selection, also slice play-back direction can be sequenced with a probability of variation. At the input stage you can capture (and loop) the last bars received or keep feeding live input. At the output stage (global to the device) a drive gain & a compressor (attenuator) are available, able of pushing sound into overdrive and saturation territory (enhancing in particular low frequency tones into bass-like drones). At the output stage also a dry/wet control is present so that the original phrase and the effect sound can be mixed together.

Made for real-time performance

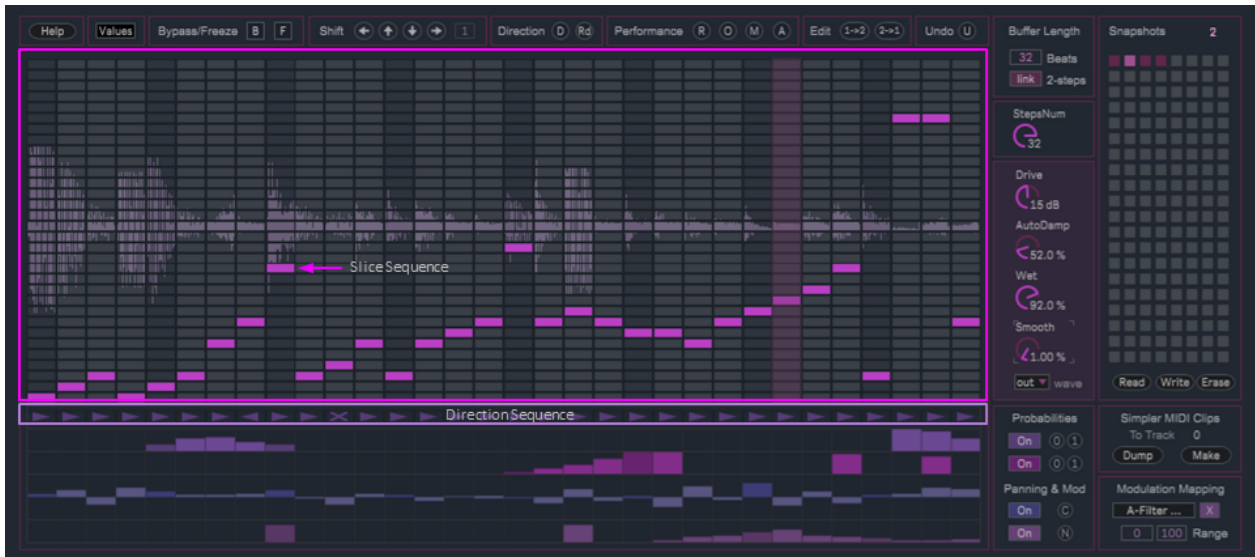
J74 SliceShuffler has been built with performance in mind. Anything (any control, any editing, anything) can be done without audio interruptions: from recalling snapshots (jumping around snapshots as a way to remix a track) to loading banks, applying edit functions and so on. Finally a performance-oriented undo function is also available and can be used at any time to revert back to the initial snapshot state.

Next in this user manual we will proceed by looking at the individual parameters, explaining their meaning and providing tips about their optimal use.

Tip: the device has a built-in *help window*. Use the help window while learning the device work-flow.

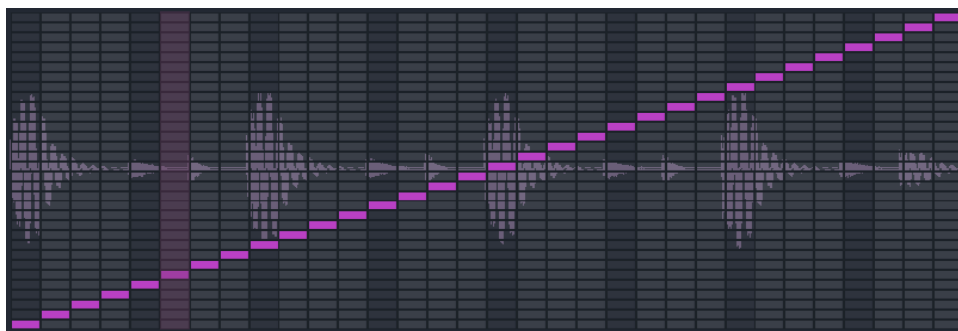
The slice sequencing area: sequencing slice selection and play-back direction

The [Slice Sequence] area covers the largest part of device. This section is structured as a *grid* (matrix): on the horizontal axis you will find the sequencing *steps* (their number defined by the [Steps Number] parameter) and on the vertical axis the selected *slice* (audio portion of the input audio buffer) to be played at each step.



For instance selecting *slice 13* on *step 9*, as in the picture above, means that at step number 9 (in time) the slice number 13 (of the input buffer) will be played back. By sequencing slices in the grid you essentially re-order the input audio buffer, in real-time, remixing the incoming audio phrase.

This also means that reverting to the original order is achieved by setting slice selectors on a diagonal-line shape, incremental from left to right, as in the picture below (this is also how the device starts up by default).



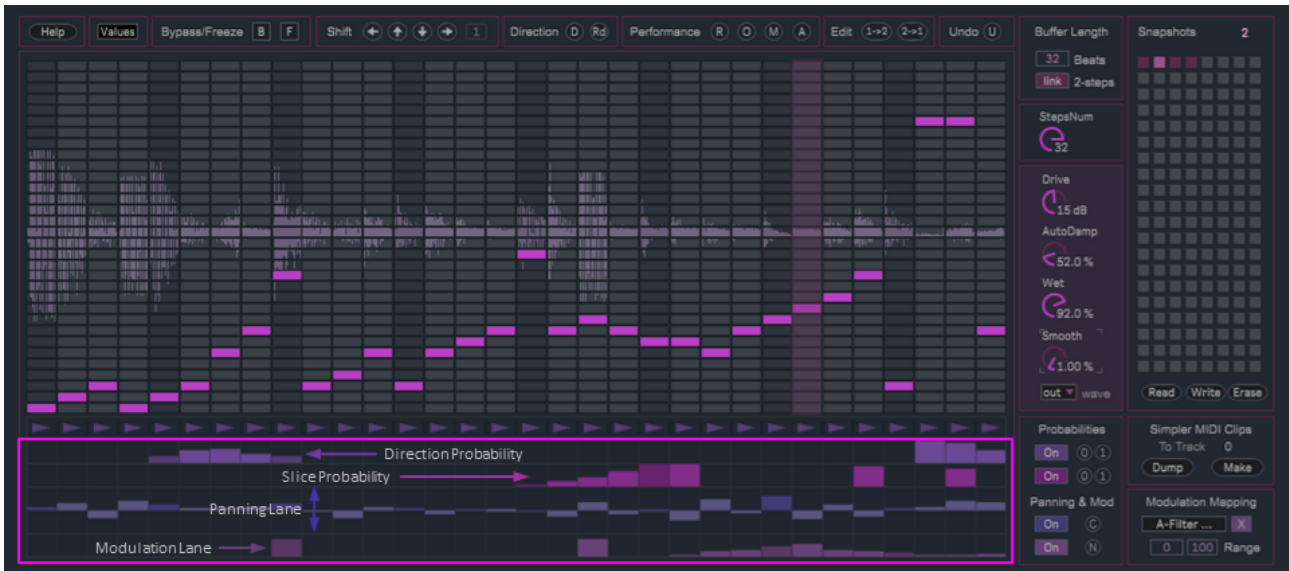
Besides slice reordering, the sequencing area also offers a [Direction Sequence] lane, located just below the [Slice Sequence] section (as you can see on the picture at the begin of this section). On this lane you select, for each step individually, the direction of playback for the sequenced slice.

Three options are available: normal, reversed or muted. The selection is represented with an *arrow pointing right* in the “normal” case, with an *arrow pointing left* in the “reversed” case and with a diagonal *cross* in case the selection is “muted”. On the referenced example (picture) you can see that at step 8 the selected slice will be reversed, at step 11 the slice will be muted and on all other steps playing back will happen in the normal direction.

The Probability lanes

Sequencing can easily become boring when it repeats itself endlessly. Here the additional *Probability lanes* (plus the additional *Panning and Modulation lanes* introduced by the next paragraphs) come to aid.

Just below the slide and direction sequencing area you can find the additional lanes. The top two lanes in the section are dedicated to sequencing *probabilities*, for direction and slice selection respectively. Their use-case is simple: when playing a step you may want to add variation to slice selection and/or to its play-back direction, but you want this to happen “sometimes” and not to repeat itself every time. This is where probabilities come into play.



The [**Direction Probability**] lane, which is the first visible from top to bottom (referencing to the picture above, highlighted section), sequences the *probability of a change* for direction selection. When set to 0% (empty step) the probability of a change is none, and the step will always play its direction as sequenced in the [Direction Sequence] lane. When set to full 100%, the probability of a change is certain and the direction will be changed randomly.

Any value between 0% and 100% introduces a “weighted chance”. Let's say you set the [Direction Probability] of step 6 to about 66% (as in the picture above). This means that the probability of a change of direction for that step is 66,6% (or 2 out of 3 times, statistically). When the step comes to play the probability of a change will be evaluated this way:

- A random seed is created for each step. If the seed wins against the sequenced probability value, the original direction is played and nothing is changed. In our example this will happen about a third of the times (33,3%).
- If the seed loses against the sequenced probability value, the original direction is changed and a random direction selection will be made. In our example this will happen about two thirds of the times (66,6%). Please notice that in this case there is still a chance the random direction will still be the same as the sequenced one.

Tip: use the latter at your advantage. For instance program a sequence so that the direction for a step is “muted” and then add a relatively high amount of probability for a change, let's say 66,6%. This means that about half of the times the step will be muted (33% due to the seed, and another part due to random selection) and in the other cases the mute will be overridden and the step will be played (with half chance of being normal or and half to be reversed).

The [**Slice Probability**] lane works in a similar fashion. The lane is the second visible from top-to-bottom (referencing to the picture above, highlighted section) and sequences the probability of a change for slice selection. Please notice the color coding in use: this is a reference to the slice sequencing area color (same tonality) for easy visual association.

When set to 0% (empty) the probability of a change is none, and the step will always play the slice as sequenced in the [Slice Sequence] area. When set to full 100%, the probability of a change is certain and the slice selection will be changed, randomly. Any value between 0% and 100% introduces a “weighted chance”, in the same fashion as for the [Direction Probability] lane. So sometimes the slice selection remains the same, sometimes it changes, depending on how high the probability is set.

Note: random selection for both [Slice Probability] and [Direction Probability] are evaluated again for each new sequence iteration (as defined by the [Beats] setting). So variations will occur at each new iteration, endlessly.

Tip: used in a sparse way, these probabilities can add a touch of variation without overdoing things. For instance select only very few up-beat spots in the sequence (like odd beats) and add a relatively high [Slice Probability], so that down beats always remain unchanged and the cadence of the sequence remains intact while changes do happen on less dominant spots. Or add lower probability levels (33% and below) for multiple spots (mainly up-beats), so that most of the times the selection remains unchanged and just one or two spots change (at each sequence repetition).

The Panning Lane

The [**Panning Lane**] is third from top in the mentioned picture and is quite straight forward: it allows you to define the panning for each step, individually and without chances or variations. It is kind of similar to traditional automation, with a twist: it is attached to the step and not to the slice (or its direction) so that variations to the latter can be applied, while the panning value stays.

Note: be aware that panning does not take care of sample zero-crossing.

Tip: extreme panning settings, like full-left on a step and full-right on the next, can be source of interesting rhythmic effects as they (may) introduce clicks. Used carefully this can be used as an interesting rhythmic effect.

The Modulation Lane

As this is an Ableton Live Max for Live plugin, it is possible to extend processing beyond the traditional AU/VST limits (that is, process only what traverses the device itself), adding processing outside the device, elsewhere in Ableton Live.

This is what the **[Modulation Lane]** is for. The [Modulation Lane] is a sequence lane which let's you sequence values for a parameter of any *other* device (or mix control) in Ableton Live. As this is attached to the SliceShuffler effect and its internal sequencing, this allows for interesting applications such as external effect modulation or mix control ducking “slaved” to the sequencer of SliceShuffler.

The use of the modulation feature is quite simple:

- A. Map the parameter to be target of the modulation, anywhere in Ableton Live
- B. Sequence its value in the [Modulation Lane]

For creating a mapping to a parameter, look at the bottom/right section of SliceShuffler. There you will find the following control section as depicted below:



The section provides the following controls:

- **[Map]** Function for creating a mapping. When enabled, the next parameter clicked in Ableton Live will be mapped for value modulation.
- **[Un-map]** Releases the parameter mapping. A new parameter mapping may be selected.
- **[Range Min]** Minimum modulation value
- **[Range Max]** Maximum modulation value

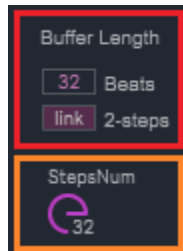
Anything in Ableton Live can be controlled, as long as this parameter is MIDI mappable in Ableton Live. Examples are audio effect parameters, mixer sends, master parameters and so on. The [Range Min] and [Range Max] controls are useful to define the *range for modulation*, in percentage of the target parameter. If [Range Min] is set lower than [Range Max], then modulation happens in the up direction (high values in the [Modulation Lane] increase modulation values); if [Range Min] is set higher than [Range Max] the modulation direction is inverted.

Tip: there are many creative applications for the [Modulation Lane]. A few examples: map a send control on the same track where SliceShuffler is placed, so that send amounts are locked into the sequence, like opening the send to a reverb effect for ambiance or to a send with a delay effect for rhythmic repetitions; map the mixer volume of any track setting [Range Min] to 100% and [Range Max] to a lower value to “duck” this track on certain steps; map a dry/wet control of an effect on the master track so that certain steps highlight the entire track; and so on.

Note: only one target parameter for modulation is possible in the current version of SliceShuffler.

Beats and Steps

As you may have grasped already, SliceShuffler operates by slicing an audio buffer captured from the input. This happens in the background and continues indefinitely (buffer is immediately available for processing, also half way a bar). The controls which define the buffer length and the number of slices are the [Beats] and [Steps Number] parameters. These parameters have an implicit relation to the Ableton Live tempo, as explained next.



The following applies:

- **[Beats]** sets the length of the captured buffer, in beats. This is synchronized to the Ableton Live tempo.
- **[Steps Number]** sets the number of steps/slices for the sequence. This parameter is global to all lanes.
- **[Link]** enables linking of number of [Beats] to the [Steps Number] (forcing the same value)

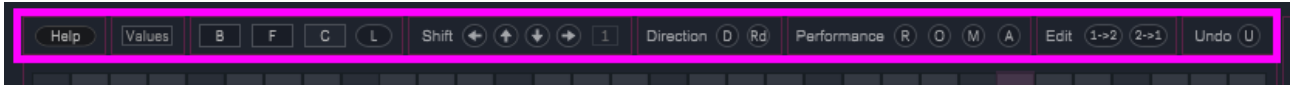
Note: changing the [Beats] number requires the device to *re-fill its captured buffer*. In order for this to happen in the background, without audio dropouts, a trick is used in SliceShuffler: when [Beats] is changed SliceShuffler *temporary switches to the pre-recorded output of the the previously processed bars*. This way output is still offered while the buffer gets re-filled. The device shows this is happening by changing its *background tonality to red* (for as long as it needs to re-fill). Once the re-fill has completed the device sets the background back to the usual tonality and switches back to the normal operation. During a re-fill editing is allowed, but does not affect the output result.

Tip: in many cases leaving the [Link] setting as “enabled” (the default) will make more sense, as it allows to sequence the same amount of time for each slice at each step. This is indeed the default mode of operation. Anyway more unusual ways of operations are possible by changing [Link] to “disabled” so that [Beats] number and [Steps Number] can be set independently. If [Steps Number] for instance is set higher than [Beats], then the “resolution” of the slices will be higher than the one of a single beat: an example is [Steps Number] = 32 and [Beats] = 16, which allows you to manipulate steps in 1/32th time intervals; instead setting [Steps Number] = 1,5 * [Beats] allows you to sequence triplets; or [Steps Number] set lower than [Beats], like in a 0,5 ratio, makes a half-tempo cadence easy to create.

Note: changing the [Steps Number] alone does not affect the re-fill of the input buffer. This can be done at any time.

Performance: editing functions

The SliceShuffler device has been built with performance in mind. Anything can be changed in real-time without audio interruptions. This applies also to the editing functions located at the top section of the device (see picture below).



These controls can be used at any time, while the buffer is being sequenced, to dramatically alter results. The presence of an *undo function* is key in this: you can push things anywhere you like, into chaos or madness and still be able to revert back to the snapshot state, on the fly and at any time, if you wish to. Ideal as a performance tool.

Note: the [Undo] function applies to the *state of a snapshot*. So, remember, create a snapshot *before* going into performance madness (see further in this manual about creating and recalling snapshots).

Editing controls include:

- **[Bypass]/[B]** Bypasses the effect altogether, instantaneously (this also bypasses the output stage)
- **[Freeze]/[F]** Freezes audio to the last played slice (with a one-step granularity)
- **[Capture]/[C]** Capture and loop the input (step color changes to CYAN when the input buffer is in use)
- **[Loop]/[L]** Acts as a count-in for [Capture] recording (with red [L] pulsing as count-down)
- **[Shift L/U/D/R]** Shifts the triggers around of the amount given by the **[Shift Amount]**
- **[Random Directions “D”]** Randomly changes step directions in the sequence, at once
- **[Reset Directions “Rd”]** Resets the direction of all steps to the normal direction (left to right)
- **[Random Steps “R”]** Randomizes steps slice selection in the sequence, at once
- **[Original Steps “O”]** Resets steps to the original triggers (diagonal-line shape)
- **[Mute all “M”]** Mutes all steps
- **[All-to-one-slice “A”]** Sets all steps to the same (random) slice
- **[Edit 1->2]** Copies the first half of steps into the second half of steps
- **[Edit 2->1]** Copies the second half of steps into the first half of steps
- **[Undo]** Undo for all changes, reverting to the last recalled or saved *snapshot* state.

Tip: [Freeze] can be used to manually inject repetitions and multiple hits exactly when you wish to.

Tip: Shifting has a twist in the SliceShuffler case, as the amount of shifting can be decided upfront. For instance it can be useful to set the [Shift Amount] to an integer divisor of the [Beats] or [Steps Number], for example with [Beats] = 32 having [Shift Amount] = 8 or 16: this allows shifting up (or down) to move to a different “region” of the input buffer, a region having different slices but with a similar down-beat to up-beat relationship.

Tip: [Random Direction] injects instantaneous chaos and can be effective for fills, if coupled to using the [Undo].

Tip: [Random Steps] injects unexpected rolls and can be effective for fills, if coupled to using the [Undo].

Tip: [Edit 1->2] and [Edit 2->1] are useful as both editing and performance functions. In editing terms [Edit 1->2] is useful when “doubling” the [Beats] value (e.g. from 16 to 32) as it allows to repeat the first half of the steps in the second half. During performance it can be useful too, as a way to temporary cut down the length of the sequence period in half (as steps will be repeated). These functions also affect probability, panning and modulation lanes.

Next the [Capture] and [Loop] features are described more in detail.

Use the device as a Live Looper or as a Parallel Audio Processor device with the [Capture] and [Loop] features

In normal operation the device keeps feeding the live input in and applies the slice processing on the continuously changing live feed. This can be useful if the input keeps changing (or if you loop it upfront, for instance running an audio clip loop as input). The [Capture] and [Loop] features explained here offer a different approach for use cases such as:

- {SliceShuffler as a live looper}: you could use SliceShuffler as a creative looper and loop the input (up to two bars). You can for instance arm the track and play a musical phrase in, and then enable [Capture] to retain that phrase looped. If the [Wet] is lower than 100% you may want to keep the [DrySelector] on the (in) selection (live input) so to play live “on top” of the loop (which can then be sliced creatively). In alternative you may change the [DrySelector] on the (cp) (capture) so that both dry and wet signal are from the [Capture] buffer.
- {SliceShuffler as a parallel audio processor}: you could use SliceShuffler as a parallel audio processor. By using the [Wet] and [DrySelector] controls you can achieve different processing results. For instance a drum loop could be rearranged with the slices and if [Wet] is set somewhere in the middle / 50% you could set the [DrySelector] to (in) and change the input drum loop used for the “dry” and this way mix a different drum loop with the captured (and processed) loop; or if you instead have [DrySelector] to (cp) you can have original (captured) loop and processed loop in parallel.

In normal operation (with [Capture] disabled) an input buffer is continuously being recorded (and overridden), all the time, although not audible. This makes possible switching back and forth between live feed and captured feed as explained in this paragraph when you activate [Capture], essentially moving the buffer from recording/non-audible to playback/audible.

There are two main ways to activate and use the [Capture] buffer: directly and with the loop count-down.

If you enable [Capture]/[C] directly (you press the [C] directly *without* the use of the [Loop] count-down), the plug-in will keep recording the input buffer till the last step and then *switch* from the live feed to the buffer playback (with this approach in mind it is recommended to have the [DrySelector] set not to follow the [Capture] state, toggle on/purple).

If you then disable [Capture]/[C] the inverse will occur: the plug-in will keep playing the pre-recorded [Capture] buffer till the last step and then switch from the buffer to the live feed at the steps restart.

Tip: Using [Capture] “directly” in this way is good if something is getting in as input and you “like to keep it” (for working on the input, such as running a new audio clip). If the [Wet] is lower than 100% you may want to also move the [DrySelector] from (in) to (cp) so that changes on the input will not be directly audible.

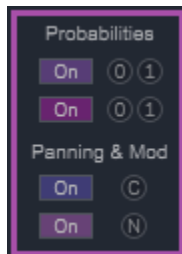
Another way to use the [Capture] buffer is to activate it passing through the [Loop]/[L] button. In this case a *count-down* is performed before the [Capture] gets recorded and activated. This process facilitates live-play looping:

- First a *count-down* will run till the end of the current steps sequence (you can see the red [L] pulsing, but the step color does not change and nothing audible changes);
- Then, at step restart, the step color becomes red and the buffer is recorded till the last step completes;
- Finally the plug-in automatically moves from recording to playback: the recorded buffer is now in playback, with the step color becoming in cyan.

Tip: Using the [Loop]/[L] button is useful as you can plan ahead the loop recording. Once the loop is recorded you can play around with the [Wet] level and the [DrySelector]. If the latter is set to (in) you can jam on top; if it is set to (cp) you can mix the original phrase with the slice-sequenced phrase. With this approach in mind it is recommended to have the [DrySelector] set to follow the [Capture] state, toggle off/gray.

Performance: ON-OFF lane selectors and all-step modifiers

Besides the editing functions, also the sequencing of the additional lanes (probability for direction and slice, panning and modulation) can be altered in real time and reverted to a known state with the [Undo] function . These areas of the audio sequencer also have dedicated controls, which can be found at the bottom of the device floating window (as shown in the picture below).



The following MIDI mappable controls are available:

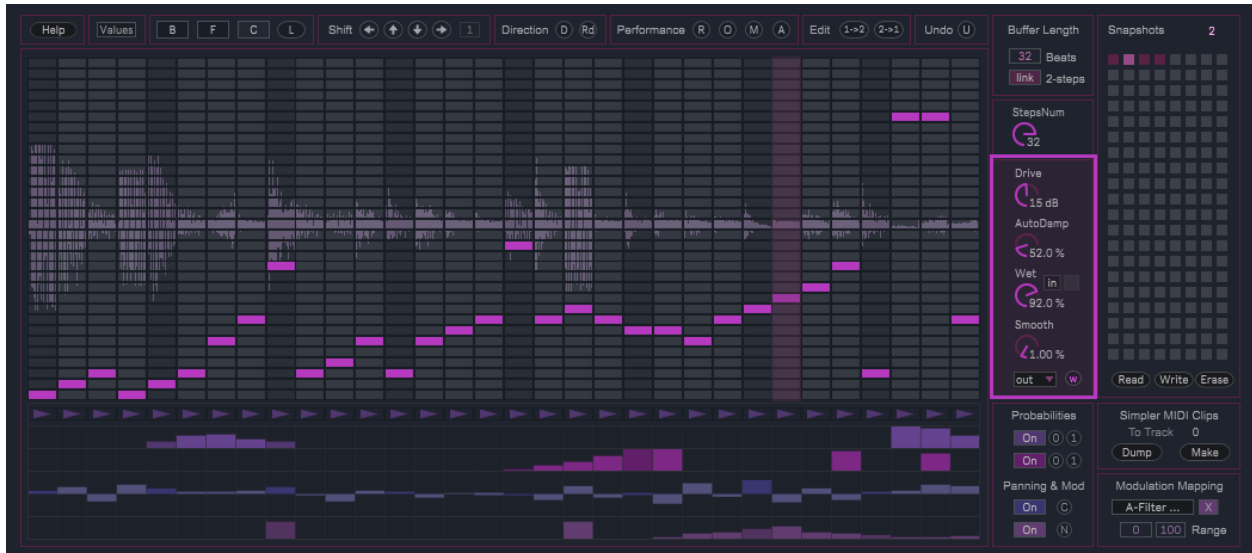
- **[ON/Off - Direction Probability]** enables/disables the [Direction Probability] lane
- **[All DP 0%]** All [Direction Probability] values set to zero
- **[All DP 100%]** All [Direction Probability] values set to max
- **[ON/Off – Slice Probability]** enables/disables the [Slice Probability] lane
- **[All SP 0%]** All [Slice Probability] values set to zero
- **[All SP 100%]** All [Slice Probability] values set to max
- **[ON/Off – Panning Lane]** enables/disables the [Panning Lane]
- **[All Pan C]** All [Panning Lane] values set to center
- **[ON/Off – Modulation Lane]** enables/disables the [Modulation Lane]
- **[All Mod N]** All [Modulation Lane] values set to zero

Tip: The [All DP 100%] and [All SP 100%] functions can be used to inject chaos into the sequence in a somehow similar fashion to the [Random Directions “D”] and [Random Steps “R”] controls, but with one substantial difference: these “All” functions apply to the probabilities, not to the sequenced values, so selections (which in these cases will be randomized for each step) will be randomized again and again at each new sequence iteration.

Tip: Use the [Undo] function to revert at once to the known state of a snapshot after applying any “All” function.

The output stage

The output stage of the device (as highlighted in the picture below) collects a few global controls which affect sound. In particular this section allows you to control the overdrive stage and the dry/wet mixing for the entire effect.



Available controls are:

- **[Drive]** Total output level, with overdrive above 0dB.
- **[Auto-Damp]** Compress (dampen) the overdrive stage causing saturation at no/reduced global volume change
- **[Wet]** adjusts the balance between incoming and processed signals (before the drive and compress stages)
- **[DrySelector]** selects the dry signal, live input (in) or capture (cp). The toggle defines selection sync to [Capture]
- **[Smooth]** Smooths value transitions from step to step in the audio buffer
- **[Display]** Select the waveform display (input, output or none)
- **[WriteSample]/[w]** Save the output buffer as a new sample (.aif)

Note: values of [Drive], [Auto-Damp], [Wet] and [Smooth] parameters can be recalled by loading a snapshot bank file if the snapshot bank file name has a section formatted like #<X><NUM> (where <X> is either (d, a, w, s) as selector). **Concrete example:** snapshot bank with name “drum_tests#d25#a75#w90#s5.maxpresets” will recall [Drive] to 25dB, [Auto-Damp] to 75%, [Wet] to 90% and [Smooth] to 5%.

Tip: The [Drive] gain and [Auto-Damp] compressor stages can push sound into overdrive and saturation, injecting new life into the audio input. This is specially useful on the low frequency spectrum as saturation can enhance low frequency tones into bass-like drones.

Tip: The [Smooth] control can be useful, especially if the audio input is other than a drum loop.

Tip: The [Wet] control allows you to mix the dry input (live or captured) and the effect results together. When set as [Wet] = 100% the SliceShuffler operates as a pure audio slice sequencer; when set to [Wet] with high values (like 90%) a shadow of the original sound is sent through, allowing resemblance; with low [Wet] levels (like 33% or less), the effect results are more subtle (the [Drive] and [Auto-Damp] are still applied).

Note: By default (toggle off) [DrySelector] (in) / (cp) follows [Capture]. This can be changed by setting the toggle on.

Audio Sample export: save the output to a new sample

You can use the device as a sample mangler to create new ideas and export them directly as a new samples. This is possible using the [WriteSample]/[w] button in the output stage section. You will be presented with a file save pop-up window and asked to save the file with a name (in the “.aif” format).

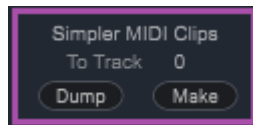
Creating samples is particularly useful for consolidating ideas and coming up with unusual material starting from simpler input material. The audio are looped with the original BPM.

Note: Please note that always (4) bars will be exported using the [WriteSample]/[w] button. This is done to allow probabilistic variations to have potentially different randomization effects if a random modifier is applied (if not, the bars will be repeating). Of course you can use a shorter section in the audio file later on.

Tip: The [Capture], [Wet] and [DrySelector] features can be combined for making variations and each time exported as new audio samples to be used in your project later on as audio loops.

MIDI clip dump (for use with Ableton Live Simpler)

A little feature, which may come handy, is the MIDI clip dump function. The name of this feature (MIDI?) may raise one of your eyebrows, but no worries, there is a use for it: the re-sequencing of slices, as done by the [Slice Sequence] area (so without probability variations and/or other functions which alter sound, such as direction or output stage), can be reproduced by creating a MIDI clip which drives a standard Simpler Ableton Live instrument (assuming the same input is – statically – loaded in the Simpler device as in the case of audio clip loops).



The MIDI clip dump functions (as highlighted above) target exactly this use-case. They create MIDI clips for a MIDI track, clips which can be used for driving a Simpler device holding the same audio (clip) as SliceShuffler is processing.

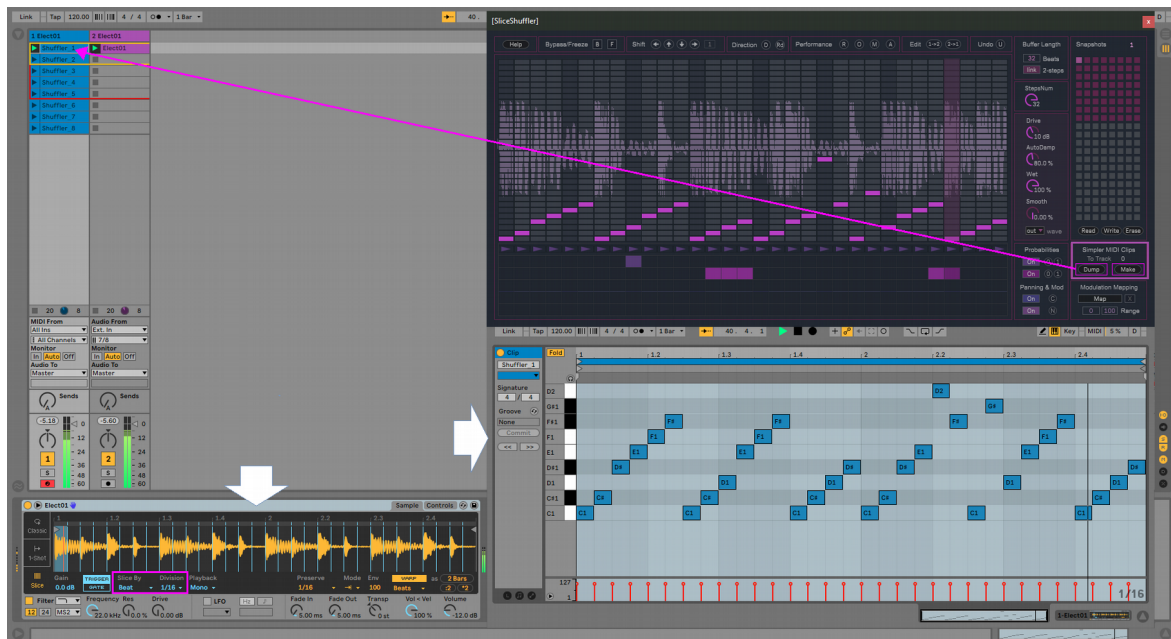
In particular:

- **[Dump]** Dumps the entire snapshot bank to MIDI clips for use with a Simpler device.
- **[Make]** Dumps the selected snapshot to a single MIDI clip for use with a Simpler device.

Note: A MIDI track must be selected before using [Dump] or [Make] by clicking on a MIDI Clip slot.

Once MIDI clips have been created, a Simpler device can be controlled on the MIDI track in question.

Tip: This feature essentially makes SliceShuffler behave as an editor for cutting and re sequencing loops into Simpler.

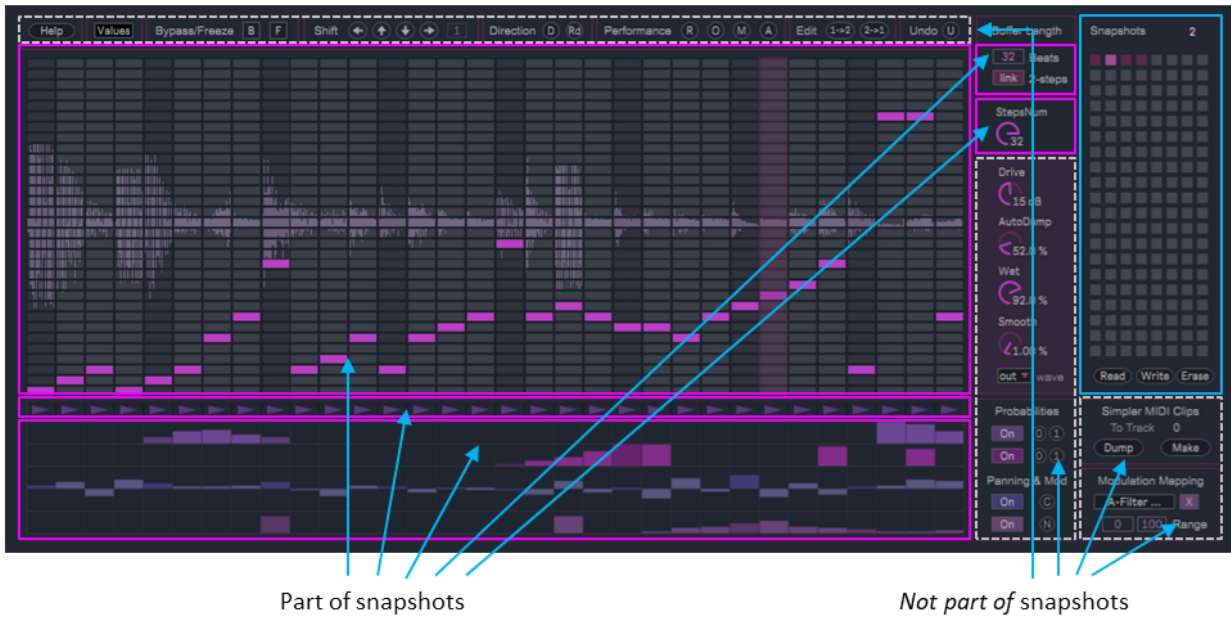


Note: set the slicing mode on the Simpler device as “By Beat” with “Division” = 1/16 (assuming SliceShuffler [Beats] and [Steps Number] were set to the same value). You can then of course use Simpler normally, without SliceShuffler.

Snapshots

The SliceShuffler device provides a snapshot functionality *aimed specifically to performance*. In fact snapshots in SliceShuffler include only a specific sub-selection of parameters so that, at performance time, useful mixing options become possible.

In the picture below the sections of the device which are *snapshot-enabled* are highlighted in purple. These are sections which relate directly to sequencing (sequenced values, probabilities, steps & beats settings) and their choice has been made so that transitions from snapshot to snapshot (and bank to bank) are *smooth* AND combinations can be created by having controls for ON/OFF of probability, panning and modulation lanes be set freely and independently from the snapshot (as well as a keeping the output stage unrelated to the individual snapshot).



How snapshots work:

- **SHIFT + CLICK** on a slot to create a new snapshot (or replace an existing one)
- **CLICK** on a slot to recall a stored snapshot
- **SHIFT + {ALT|OPTION} + CLICK** on a slot to delete a stored snapshot
- Snapshots banks can be exported using the **[Write]** operation.
- Snapshots banks can be imported back using the **[Read]** operation.
- All Snapshots from the current bank can be erased using the **[Erase]** operation.
- On the top/right corner you can also find a MIDI mappable **[Snapshot Number]** for recalling snapshots

Important: the file extension for snapshot files MUST be “.maxpresets” in order for the files to be readable.

Note: the complete state of the device (every parameter) is saved with the Ableton Live project, independently from snapshots. This also applies to parameters not part of the snapshot feature and to the mapped modulation destination.

Questions?

If you still have questions or issues, please send an email to: [info \[at\] fabriziopoce.com](mailto:info@fabriziopoce.com)