JAX LOOP RECORDER (AudioUnit)

PREVIEW MANUAL

JAX LOOP RECORDER

The JAX LOOP RECORDER is a very simple to use audio unit, which provides 10 slots for audio recordings (loops) inside AudioUnit environments and allows synchronized playing of recorded and loaded(*) audio files. Recorded files are saved to disk immediately and can be shared between different sessions, instances and external apps afterwards.

(*)The loaded audio files should fit to each other in loop size, sample rate and musical pitch because there is no deeper intelligence built in here. However, the internal players will adopt all loaded sounds to the current hosts sample rate. The original files will not be affected by this realtime process. Recorded files are saved with the current host sample rate.

Creating such kind of audio units is quite difficult, as it requires allot of care for the threading and possible realtime audio interruptions. The audio streaming flow must not be broken in any way, but it is an absolute requirement to allocate memory and doing disk operations for loading and saving files while realtime operation of such a unit.

There are 10 available audio slots for short loop-based files with the specified parameters for tempo and loop size. This means, each file should have the same size or a multiple length or a division of the others and have a corresponding musical pitch and tempo base to be musically senseful. The files are expected to play together seamlessly but basically can include any sound sources for more experimental results.

The easiest way is to load a rhythmic file for rhythmic guide and then adding successively one recording or file after another to the arrangement. The sound can be played in with external instruments and also include any effect chaining prior the recording input.

The output of the 10 channels is mixed together internally (possibly also via multi busses) and does not affect the input audio channel in any way, which is always perfectly isolated from the currently playing sound.

It very much works like a classic looper, where the user can activate one slot at a time for live recording of successively played-in sound fragments. The recorder will start synchronized to the currently playing files and the files will be loaded synchronized into the armed slots afterwards. The synchronisation is beat based and called up from the hosts transport info. Fractions may be possible too.

The unit is an audio unit effect and requires stereo audio input of any kind. This includes microphone input, generators, drum machines and synthesizers with or without additional effects. The routing of the setup is usually managed by the host application exclusively, not the plugin. It just will record, what is coming in. The files are cut at the internal synchronisation markers without any modifications (fade out, fade ins). However, the player instance ensures smooth fading of the segments in realtime.

The recorded files will be saved directly to disk in streaming format (32 bit floating point, Apple AIF sound format).

Realtime saving is here a conceptual behaviour and requires special threading considerations inside an audio unit environment. This way a recording never will be lost but can introduce small gaps in the audible audio flow. Also any file format conversion or compression is not applied during processing for certain reasons. This always can be done afterwards in an optimized manner, with specialized external tools.

Alternatively, the user can import and load any iOS supported files (including compressed formats with constant bit rate) from disk into the current player bank and construct tempo based sound collages this way, only from existing files or in any combination with the made realtime recordings.

In auto-recording mode, all free slots will be recorded one after another in a kind of batch, until there is no more free slot available. Please note, that there may be one cycle space between successive recordings, due to the fact, that the special threading requires some time for saving the results to disk. The realtime audio flow must not be broken in any way and therefore files are saved in a totally independent thread of the core remote audio processing and any file access must be synchronized therefore.

The state of currently loaded files is saved with a preset and also can be saved and loaded manually by the user. Files are linked to the preset file, so if files are removed or modified externally, it can invalidate the saved presets and setups. Missing files will result in silence, modified files will be completely recalculated upon opening a preset.

Files also can be explicitly exported for usage with other tools.

JAX Loop Recorder will consequently work with the hosts tempo and transport (loop) information for adjusting the current loop size and tempo of a running project. If the hosts do not support tempo and/or loop information at all, the files will result in fixed 8 beats long segments at the tempo of 120,00 bpm, based on the hosts sample rate. Tempo changes while operation are not supported and most likely will result in discontinuities of the entire sound result. However, it is possible to adjust the transport controls in a way, that delivers perfect multiplication or divisions of the resulting segment lengths.

Note: If files of different sample rates are loaded, the sample rate of the host will be applied while playback automatically. Some problems may apply to loaded files with a different recorded tempo than the currently loaded project settings.

Remarks: JAX Loop Recorder uses the Apple ScheduledFilePlayer class for its sound engine. Unfortunately this player has some known quirks, limitations and also certain problems, which is completely out of our control. We did our best to keep the unit as stable and useful as possible.