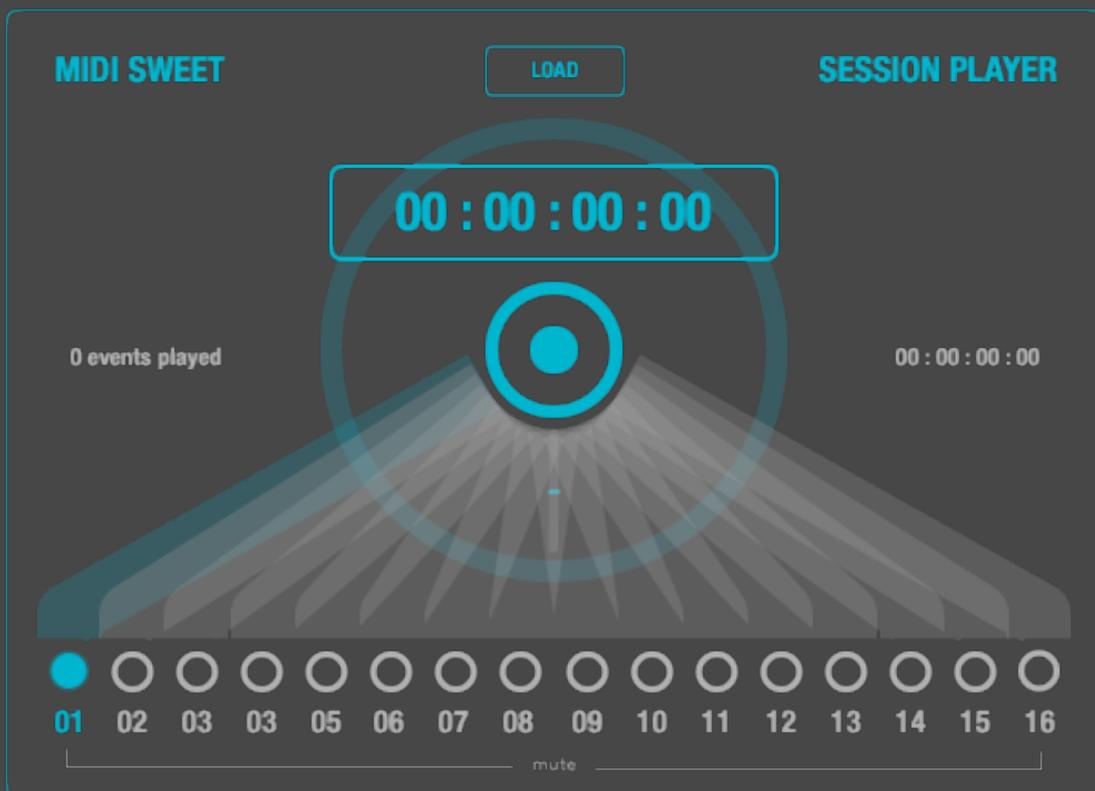




**MIDI SWEET SERIES**  
**MIDI SWEET : MIDI Session Player [ AUDIO UNIT ]**  
**MANUAL**

## MIDI SWEET SERIES : MIDI Session Player

MIDI Session Player continues our MIDI SWEET series and is one of the “missing link” tools, providing essential functions for MIDI processing in AudioUnit format.



## Playing Standard MIDI Files

MIDI Session Player Audio Unit is a **multi-channel MIDI player**, which is able to play back format 0 and 1 compatible Standard MIDI files.

Please note, that MIDI data is silent. MIDI commands are used to drive tone generators, synthesizers and effects, which consequently must understand MIDI.

MIDI Session Player is an instrument and has an inbuilt MIDI Port, that will pass all file events sample accurately thru the destination devices. The MIDI channels are passed thru as originally encoded inside the file. All tracks will be loaded into one of the 16 possible internal MIDI channels. If special remapping or filtering is required, the user must do this with the host application.

While MIDI Session player was mainly developed as an accomplishment for the MIDI Session Recorder, it basically is a standard conform MIDI Player, which can load and play back any multichannel MIDI file. The user interface mimics the inverted version of the MIDI Session Recorder, separately available, which can record live sessions and MPE inside audio unit environments.

The unit must be connected to destination tone generators or effects, that can understand the MIDI data. This is usually done with a host application, that allows manual routing of MIDI output and MIDI input connections. If MIDI Session Player is not connected, it still may play back MIDI files, but without any audible result.

~~Currently SysEx is not supported due to the fact, that many audio units do not support SysEx and SysEx is basically always device specific.~~

Special things are to say about tempo events of MIDI Session Player. Because of the fact, that this is an Audio Unit, it always will consequently adopt the current reported tempo of the host application. It is just impossible for any audio unit to dictate a host application any tempo change. So all tempo events inside a MIDI file will be ignored.

MIDI Session Player is able to load MIDI files of any length, that are supported to be loaded by iOS. The core will use Apples frameworks for loading and saving MIDI files. All realtime events will be generated directly inside the realtime audio kernel and are sample accurate. So there are no timers or threads used outside the core audio kernel, except for the user interface, which is completely separated from the core audio kernel process.

Users can save GUI performance, if the editor is closed.

Please note, that spontaneous bypassing by external processes (i.e. the host application) or system interruptions can cause hanging notes, as there is no chance for the audio unit to see or prevent this under all possible circumstances.

## **Integrating The iOS File System**

The complete iOS file app functionality was adopted to enable users to import and manage their MIDI files while operating with the Audio Unit inside a host application. But the imported MIDI files are located inside a special shared disk memory area of the mobile devices, because these files

must be accessible by all (multi user) host applications, where the unit is loaded into. Each unit basically is a user. iOS does not support free global disk access (everything is sandboxed here) and also does not support easily accessing files from anywhere. So files for playback with MIDI Session Player must be imported first.

Exporting MIDI Files is supported by MIDI Session Recorder.

(c) 2019 - 2021 [digitster.com](http://digitster.com)