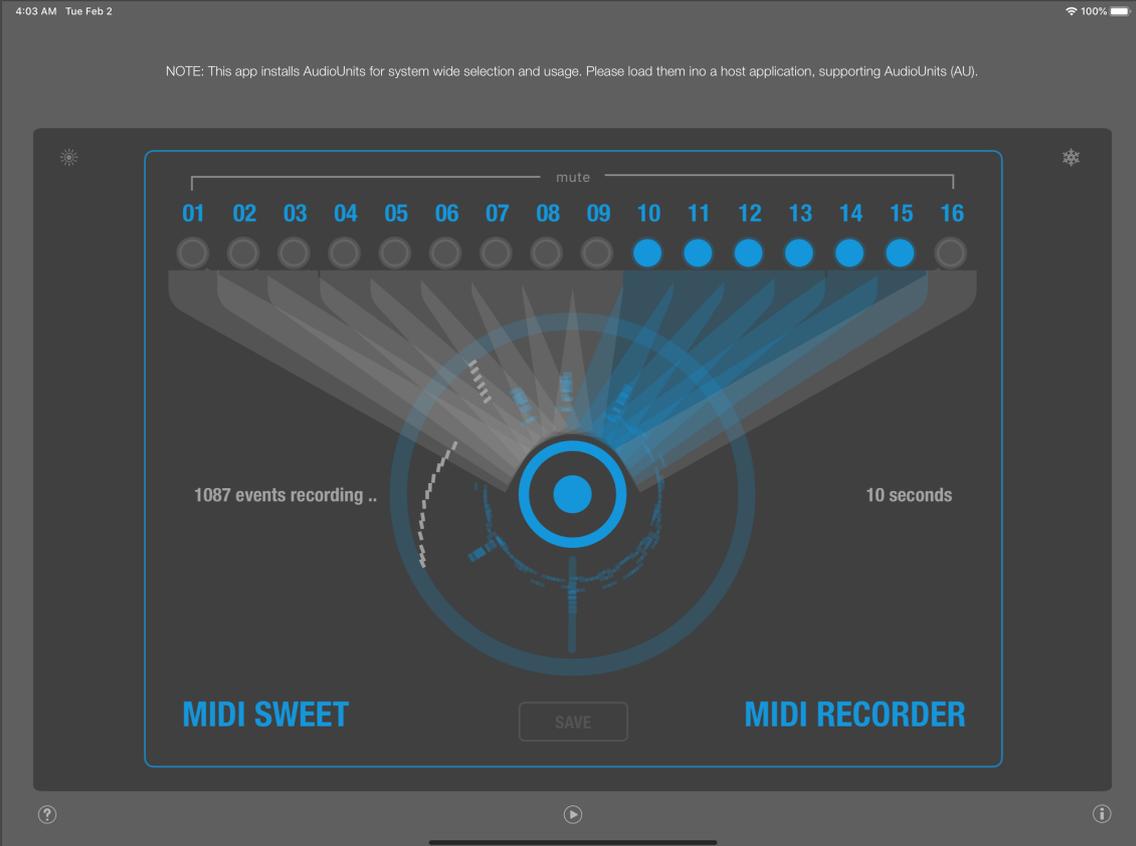




**MIDI SWEET SERIES**  
**MIDI SWEET : MIDI Session Recorder [ AUDIO UNIT ]**  
**MANUAL**



## MIDI SWEET SERIES : MIDI Session Recorder

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

## Producing Standard MIDI Files

MIDI Session Recorder Audio Unit is a **multi-channel MIDI recorder**, which is able to capture connected MIDI realtime stream(s) and provides a function for exporting the results to a Format 1 compatible Standard MIDI file.

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 Recorder has an inbuilt MIDI THRU function therefore, that will pass all incoming events sample accurately thru the destination devices. The MIDI channels are passed thru unchanged. If remapping or filtering is required, the user must do this with the host application.

MIDI Session Recorder does not save any setups of loaded or connected audio units, as this is an exclusive function of the host applications, not the audio units. It is not even possible for an audio unit to save such connections and setups. You must ensure that the unit is routed with any sense between senders and receivers and remember what the purpose of a certain recording was, if you export the resulting MIDI files.

Unlike many other products, which are often limited in functionality, MIDI Session Recorder is able to capture **all 16 Channels of MIDI Data input at ones**. Regardless from where these streams are coming. It can be used as a **live session recorder** or for the many devices and apps, that do not implement internal MIDI file recording and export.

So complete arrangements or the output of drum machines, sequenced synthesizers and arpeggiators and so on can be

recorded (assuming these devices and apps will output any MIDI data). **One special usage scenario is MPE (MIDI Polyphonic Expression) recording**, due to the fact that MPE possibly uses more than one MIDI channel for the expression data and note informations. Basically MPE is just normal MIDI data and nothing very special. Merely the receivers must be able to translate this specially encoded MIDI data correctly and correctly adopt it to their tone generation.

NOTE: MPE data streams mostly will be not usable with any other MIDI playback mechanism due to their special encoding.

The main requirement for using MIDI Session Recorder is the routing inside the host environment. All streams that have to be captured with the MIDI recording must be connected to the MIDI input port of MIDI Session Recorder. The MIDI Thru port of MIDI Session Recorder can forward the received MIDI data unchanged in realtime and sample accurate, but with the unavoidable latency of 1 sample block.

So (as always) shorter processing buffers are required to keep latencies small. This latency with MIDI processing is a technical design issue with all audio unit environments and hosts. And it is impossible to be compensated by an audio unit itself. And if other developers claim, they do not have such issue, they just are liars (and do not know anything).

The unit will always export a 16-track Format 1 MIDI file of virtually unlimited length for later usage with all Standard MIDI File compatible applications on all platforms. We created MIDI Session Player specially for the purpose of loading and playing back MIDI files, made with the MIDI Session Recorder. It is just an inverted version of the MIDI Session recorder. But there is no need to use the MIDI Session Player at all.

However, the Session Player can virtually load and play back any Standard MIDI file inside an audio Unit Host environment.

Please note, that MIDI Session Recorder may capture silence for virtually unlimited length, if running in background. However, there is a validation function that checks for available data upon MIDI export. If data was there, the silent parts will be still in that resulting MIDI file.

The unit can be silently used in background to capture a whole session over a long amount of time. When the editor is closed, also much less GPU performance will be occupied. The fancy animations will be bypassed then.

A session must be explicitly saved/exported, otherwise the recorded data gets lost.

Upon export, the single MIDI channels will be grouped into corresponding 16 MIDI tracks. The Tempo of the recording is set to current host bpm, regardless how the tempo of the host is changing while recording. Tempo events are not captured this way with our session recorder unit. If you change the tempo while a session is running, all the data will be saved with the last valid tempo of the host.

Also the MIDI Session Player will exclusively use the current tempo information of the host, if even available. Default tempo is always 120 beats per minute.

The user interface does generate visual feedback while operation and also allows to apply a channel filter. So you may limit the results to the channels of interest.

Please note, that MIDI Session Recorder currently does not support recording special MIDI realtime and system

messages of any kind or SysEx data. The latter will possibly be implemented with a later update. But usually SysEx data is not well supported by many audio units and therefore can cause problems, as there are special realtime requirements for SysEx data blocks, which are dynamically allocated inside the core audio processing kernel. Special MIDI system messages, like clock or MIDI Time code messages do not fit into the MIDI file specifications and will not be recorded therefore.

## **Integrating the iOS File System**

The complete iOS file app functionality was adopted to enable users to export and manage their MIDI files while operating with the Audio Unit inside a host application. But the recorded MIDI files are located inside a special shared disk memory area of the mobile devices, because these files must be accessible by all host applications, where the unit is loaded into. 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.

The files generated by MIDI Session Recorder can be exported, opened and edited with any Standard MIDI File compatible application afterwards and these files also can be opened with any MIDI File Player directly and (for instance) played back with a routed instance of a multi-timbre MIDI Module, i.e. MIDI Module Unit from our MIDI Sweet Series.

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