SAWStudio64 Latest Release Info

Introduction

Welcome to SAWStudio 64 Bit. This environment spins off from the 32 bit version with all features and updates thru version 5.7. Version numbers for the 64 bit versions will begin at 1.0.

Please take some time to explore the built-in HelpFile and also download the .pdf version of the manual, available at ***www.RMLLabs.com***. The navigation capabilities are powerful and complex and will require some getting used to. Reading, exploring and practicing inside the environment will greatly enhance your experience of the interface. We feel the rewards gained for the time invested, will be great. Have fun!

Version History Follows:

>====<

Version 2.4

Enhancements

* The Ctrl-Backspace Key has been enhanced to recognize Multitrack modes and act accordingly. Even with the Record Remote active, if Select Mode is active, selected entries are duplicated at the cursor position aligning the Key Entry with the cursor... if Automaion Display mode is active, then a duplicate of any marked automation is copied to the cursor position... if the Control Track is the active track, then a duplicate of any marked Ctrl entries is copied to the cursor position... otherwise Retake-All is the result.

* The Comm Port Settings have been enhanced to support 115200, 128000, and 256000 baud rates.

Bug Fixes

* The automatic .wav Datalength detection has proven buggy because many .wav files include chunks that can reside after the data chunk and create extra junk data at the end of the .wav data in many files. So, the .wav datalength now defaults back to the actual header stamped .wav datalength to properly truncate .wav data to actual stamped length values. But this now defeats the ability to fix corrupted .wav file headers that contain .wav datalength longer than is allowed in the Windows .wav file length header storage, thereby cutting off data much shorter than the actual extended length of the data in files that have been recorded for too long of a time.. So, I have now added a new File menu option called "Ignore Audio Data Length For All New File Opens" which can be used to help fix headers that are corrupted with datalength values that have exceeded the allowable datalength value size because you recorded a

live session for too long without splitting to new files. This option will not save with the preference file and will always revert back to OFF when restarting the program. It is meant to be turned ON when you need to fix corrupted headers and then immediately turned OFF when done. When this option is Checked, opening new files will ignore and reset the .wav datalength to the end of the file, thereby accessing the full length of the file which may also add some junk data at the end of the .wav. data. This can salvage corrupted recordings from files that are too long compared to the max data limit. Once opened you should now be able to see the extended long datalength of the .wav data and you can create regions that are within the max length allowed and then split and extract the files into part 1 and part 2 etc. This then allows you to create a new session header using the smaller region lengths as separte multiple files to create a fixed legal header. Make sure to turn this option OFF when you are done for normal file operation on future sessions.

* The VST3 .ini scanning code has been modified to allow and detect plugin names that include a single or multiple periods in the .ini path data line. This helps solve waves loading issues and other plugin installs that may use periods with version numbers in the path data line

>---<

Version 2.3

Enhancements

*

Bug Fixes

* Modified code to fix path name length to handle vst3 installs with extra long path lengths due to multiple folder nesting of the install path.

>---<

Version 2.2

Enhancements

*

Bug Fixes

* Modified code to fix buffer size change chatter that could occur with certain vst3 plugins during automation changes. Some vst3 plugins required an extra variable adjustment during live playback to properly accomodate buffer size changes which can occur during automation events like volume fades.

>---<

Version 2.1

Enhancements

*

Bug Fixes

* Enhanced code to stop window corruption crashes that could occur with the newer Windows 10 and 11 updates.

>---<

Version 2.0 ($100 Paid Update - July 2025)

Enhancements

* The introduction of VST3 Plugin support. It took a while, but its finally here. VST3 Plugins can now be loaded directly into the program with the use of VST3 .ini support files. This code can also support Waves VST3 Shell loading.

  While the use of .ini file links remains optional for VST2x plugins, although recommended, they are now required to use for VST3 plugins. The .ini file is a simple text file that links the actual plugin installed path to the SAWStudio64 interface.

  The use of the .ini file eliminates the need for the program to scan harddrives looking for and collecting all locations of installed VST3 plugins and also allows you to control which plugins appear in the FX Plugin listboxes in SAWStudio64.

  The VST3 Plugin programming interface is much more complex than the older VST 2x versions and for the most part now attempts to control default installation paths in a more restricted manor. In most cases, the default install path in Windows for a VST3 plugin can be found at "C:\Program Files\Common Files\VST3". Within that folder, you may find more sub-folders, depending on the various companies involved, which expand out further until the actual .vst3 plugin progaram file is finally found. Although this is a common default behaviour, VST3 plugins can and many times are installed in many various folders, depending on the companies involved.

  For instance, a UAD LA3A default install path might look like this:

  C:\Program Files\Common Files\VST3\uaudio_la3a.vst3\Contents\x86_64-win
  \uaudio_la3a.vst3

  This can get quite complicated to keep track of when you might have dozens or more different VST3 plugins installed from many different companies. The .ini files keep things neatly organized. Each plugin requires its own .ini file that points directly to the actual installation path and VST3 file. The .ini file is a simple text file created in the notepad or similar text editor which has one line of text stating the complete install path, including the actual .vst3 plugin file itself. The .ini file is saved with its

name being the actual name that will display in the FXChoices list of all available plgins. The name will be displayed in a deep green color and listed as a VST3 Plugin.This file is saved in the VST_Plugins folder which is under the main SAWStudio64 install folder.

As an example, the UAD LA3A compressor listed above would have an .ini file that could be named "UAD LA3A Compressor.ini" saved in the "C:\SAWStudio64 \VST_Plugins" folder. The .ini file would contain the one line of text as follows:

C:\Program Files\Common Files\VST3\uaudio_la3a.vst3\Contents\x86_64-win \uaudio_la3a.vst3

In the FX Choices listbox in SAWStudio64, you would see an entry labeled "{VST3} UAD LA3A Compressor" in a deep green color.

Waves plugins introduced the Shell concept to an already complex system of design. The Waves plugins use a shell file that contains many different separate plugins wrapped inside the shell file. This is handled slightly differently in SAWStudio64, in order to maintain compatibility with the way presets and sessions are saved and recalled.

In general, the shell file will install to the default location as defined above, but the individual plugin files will install in a Waves independent scheme of folders and locations, generally not directly available to the host interface. To handle this, we use a slightly modified version of the .ini file form. The main shell file can be opened for reading from the FXChoices listbox, but to use the plugins, we need to create a separate .ini file for each plugin we want to access.

So, first you would create an .ini file for the shell itself, which you would probably find at:

C:\Program Files\Common Files\VST3\WaveShell1-VST3 16.0_x64.vst3

Now when you create the .ini file for this shell file, you will change one thing to denote this as a shell and not an actual plugin link. Create the .ini file and name it "Wave Shell v16.ini" or something similar and on its one line of text within the file, add an "s" onto the .vst3 at the end of the line so it would actually look like this:

C:\Program Files\Common Files\VST3\WaveShell1-VST3 16.0_x64.vst3s

This lets SAWStudio64 know that this is a shell file and not a plugin link. When you select it in the FXChoices listbox, it will list out all plugins that are contaned within it. You might find many different versions of each plugin you purchased, such as:

WLM Plus 5.0
WLM Plus 5.1

WLM Plus Mono
WLM Plus Stereo
WLM Meter 5.0
WLM Meter 5.1
WLM Meter Mono
WLM Meter Stereo

This is what gets installed when I only purxhased the WLM Meter. You should make a note of each plugin name that you wish to use. You must match the name exactly as you create a separate .ini file to link to the plugin from the shell file. For instance, I only need access to the WLM Meter Stereo plugin, and do not need to clutter my FXChoices listbox with all the other versions that I might never use. The actual .ini file might be named "Waves WLM Meter Stereo" and would contain the one line text path plus the extra shell information needed to extract the proper plugin from the shell:

C:\Program Files\Common Files\VST3\WaveShell1-VST3 16.0_x64.vst3s WLM Meter Stereo

Notice the "s" added to the shell filename to properly access the shell file and then the actual name as listed inside the shell of the actual plugin file. This .ini file will actually load the plugin listed from within the Waves Shell mechanism. Do this for each Waves plugin you wish to use from the shell file. I recommend leaving the shell .ini file in the FXChoices as each time you purchase new Waves plugins they could be appended in the same shell file and you will need to list it again to note the actual new plugin installed names. But note that depending on plugin versions, Waves could create a different new shell file that handles only certain plugin versions you purchase, so you may have to create a new Waves shell .ini file for that just so you can see the actual included plugin names.

Hang in there getting thngs all configured and setup. In the end, you will have more control in what gets listed in your FXChoices Window and you can control exactly what and how it displays in a less cluttered Plugin selection view.

* Added 64 bit file pointer handling to allow multitrack control of huge file sizes beyond the .wav file header limit of 32 bit file position pointers. If you record live events that go un-interrupted for multiple hours, you could easily run past the wav file header max limits of a 32 bit file position pointer, even within 64 bit code. This might show up as a corrupted wav header for session files saved that are past this size limit... which could be maxed out somewhere around 2 to 4 hours, depending on file format and samplerate. This corruption could result in the session opening and the pointer wrap-around would only allow display and playback of around 30 mins of the event. This new code enhancement allows SAWStudio64 the ability to detect this corruption and rebuild the corrupted pointers internally allowing display and playback of the complete > 4 hour files. At this point you can cut the files in half or quarters and resave them with no corruption rebuilding the huge session with smaller and not

corrupted source files, thereby saving the session.

Bug Fixes

* Includes fix for VST3 and VST settings save if plugin window has not been moved before saving.

* Includes fix for VST3 settings save tweaks.

* Includes fix for VST3 automation Bypass not working.

* Extended the size of VST settings buffer to help handle plugins that require large amounts of data storage for preset saves and session saves.

* Fixed code to correct a possible crash when connecting a remote to the host with an empty session in the host.

* Added Window Caption Names to window views that were missing caption names.

>---<

Version 1.4

Enhancements

*

Bug Fixes

* Fixed code to correct a possible crash when opening edl sessions with FX automation entries and re-ordered FX plugins.

* Fixed the Browse For Folder dialog to correctly enable and disable the combobox options.

>---<

Version 1.3

Enhancements

*

Bug Fixes

* Fixed code to correct wrongly calculated wav file data length written to created wav files.

>---<

Version 1.2

Enhancements

*

Bug Fixes

* Fixed code to stop crashes that could occur when switching layers live during playback when the track also includes automation changes.

>---<

Version 1.1d

Enhancements

*

Bug Fixes

* Fixed code to properly list the SACLink Record Busses in the popup Record Input Selection menu.

>---<

Version 1.1c

Enhancements

*

Bug Fixes

* Fixed code to solve video sequence entry data corruption that could cause crashes when editing and moving video sequence entries.

>---<

Version 1.1b

Enhancements

* Expanded the max Output Channels for VST Plugins to 64 which should help with compatibility for plugins like Kontakt 6 and others.

Bug Fixes

* Adjusted code to handle even larger sizes for VST plugins data storage.

* Fixed code in the Random Dither function that could cause a crash when playback was engaged.

>---<

Version 1.1a
Enhancements

Bug Fixes
* Fixed code to allow vst compatibility with plugins like Melodyne 4x and others that depend on accurate vstTimeInfo data.

* Adjusted code to compensate for VST plugins that store very large amounts of data with their settings in edls and other file saves. This code should now fix some plugins ability to save settings with session and other template files.

* Fixed code to correctly parse data strings entered in the Control Track Built-In Commands dialogs.

>---<

Version 1.1
Enhancements
* Changed code to drastically improve performance on Win 7, 8 and 10 for all shared SACLink operations between SAC and SAWStudio. CPU loads when using the SACLink are no longer heavily increased as data is passed between the two programs during recording and playback. This now matches, or even exceeds the SACLink performance that was achieved with the 32 bit versions in XP.

* The helpfile now includes specific Win 10 tweaks.

Bug Fixes
* Fixed code that could overrun a few array variables when Max Count input and output devices are used.

>---<

Version 1.0g
Enhancements

Bug Fixes
* Fixed code to correct the Export Track(s) To SoundFiles function.

>---<

Version 1.0f
Enhancements

Bug Fixes
* Fixed code to correct MT Soft Clipping.

* Fixed code to correct certain internal Data Structure Offsets.

* Fixed code to process Command Line options correctly.

EQ Plugin Ver 1.0b
* Fixed code to correct an asm string routine.

Echo Plugin Ver 1.0a
* Fixed code to correct an asm string routine.

Video Track Viewer Plugin Ver 1.0a
* Fixed code to correct an asm string routine.

>---<

Version 1.0e
Enhancements

Bug Fixes
* Fixed code to correct missing Wide Mixer TitleBar Text.

* Fixed code to correct BuildMix and Record Setup Dialogs from opening only once after loading a new shade.

>---<

Version 1.0d

Enhancements

Bug Fixes

* Fixed code to correct VST Synth Multi-Output handling.

* Fixed code to maintain proper playback position when switching layers during MT playback.

>---<

Version 1.0c

Enhancements

Bug Fixes

* Fixed code to stop crashes when using the Rvs Audio function on a MT Region.

>---<

Version 1.0b

Enhancements

Bug Fixes

EQ Plugin Ver 1.0a

* Fixed code to correct Band 7 from shifting audio data left.

>---<

Version 1.0a

Enhancements

Bug Fixes

* Fixed the Progress Bar dialog window to behave properly on longer processes and properly count all the way up to 100% with no freezing or repeating.

>---<

Version 1.0

Enhancements

* Includes all features of the 32 bit version up thru 5.7.

>---<

>====<