# User Manual

This manual explains the set up procedure to operate the Automizer VST Plug-in using the host application REAPER. However, the principles of loading and using the plug-in for other VST compatible host applications should remain the same.

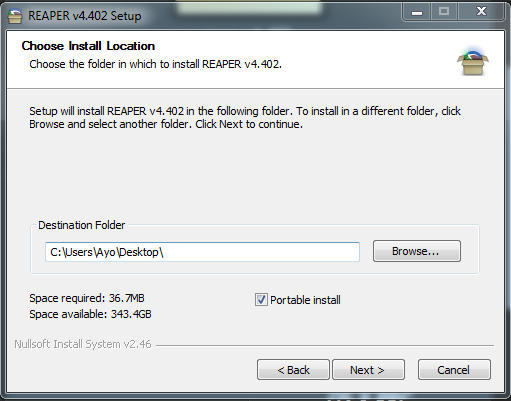
**Compatibility**

* This audio plug-in can be only be loaded into audio host programs that support VST2.4 and 3 plug-ins.
* The VST plug-in submitted has been complied in 32 and 64 bit and is compatible with Windows Operating Systems only. However, this can be compiled as an AU Plug-in to work on Apple Macs.
* It has been tested and confirmed to work on Windows 7,8 and 10 OS.
* At least a 2 GHz dual-core or faster processor with at least 1 GB of RAM is recommended.
* This plug-in is processor intensive, hence it is strongly advised to set your laptop/PC power plan to High performance mode.

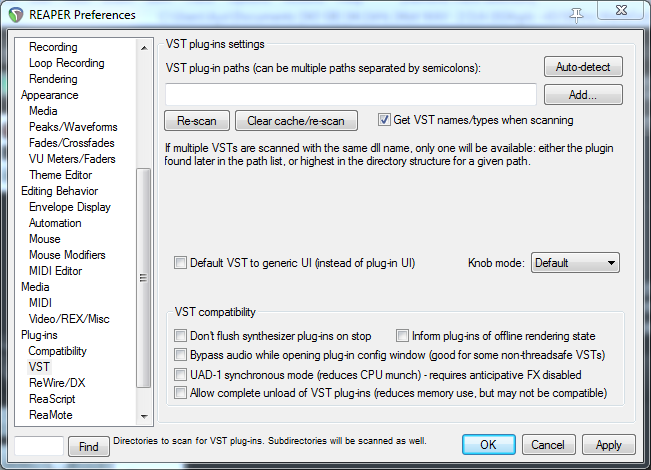
**Set Up**

* A digital audio workstation (DAW) MUST be installed on your system for the plug-in to work. If it is not possible to install software on the PC of use, here are steps to overcome that limitation.

1. Head on to <http://www.cockos.com/reaper/download.php> and download REAPER
2. During installation, make sure the portable install button is selected when it appears if you are using a machine where you don’t possess administrator privileges. Place the files in a folder of your choice



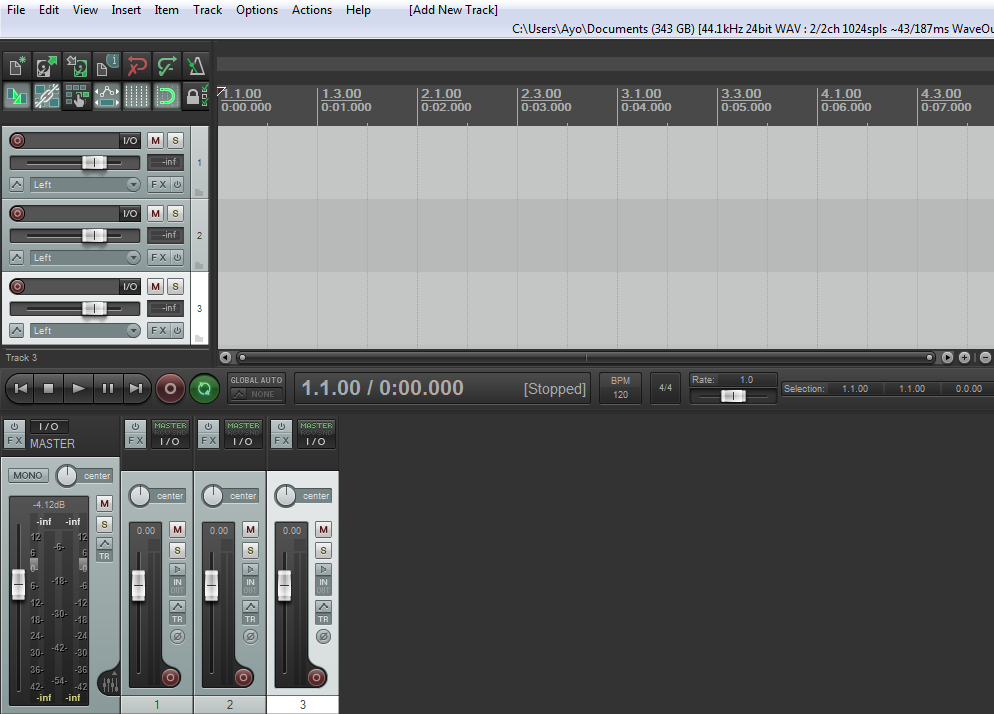
* In REAPER, click on *“Options”* in the ribbon menu and select *“Preferences”.*



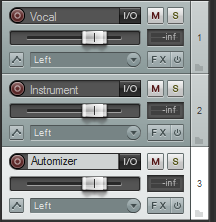
* In the plug-in paths, type the location of where you have stored the supporting files folder and then click the re-scan button. The plug-in will now be useable.

**Routing the VST Plug-in**

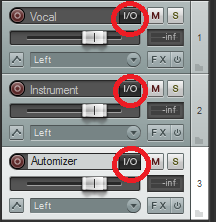
* Because the plug-in requires two inputs to work successfully, here is the correct method to set things up using REAPER.
* Create three tracks in REAPER by selecting “Track” then *“Insert New Track”.*

****

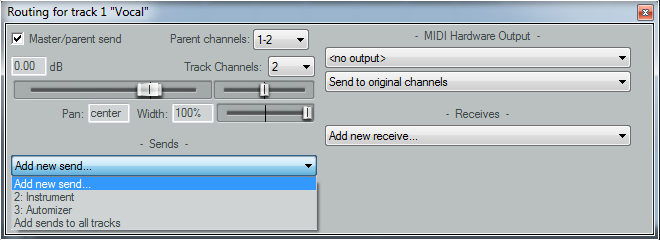
* Rename the tracks in this order *“Vocal”*, *“Instrument”* and *“Automizer”* respectively.

****

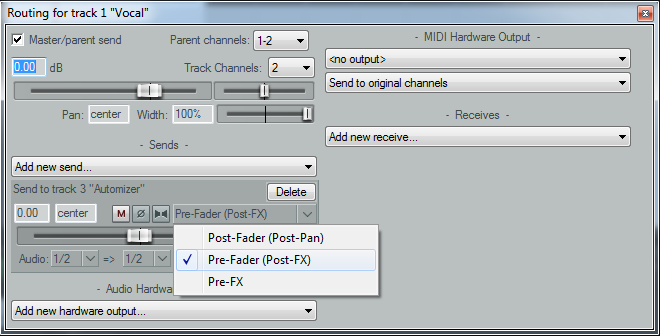
* Click on the *“I/O”* button on the vocal track, and follow the three step procedure. In newer versions of Reaper, the I/O button has been replaced with the “Route” button.



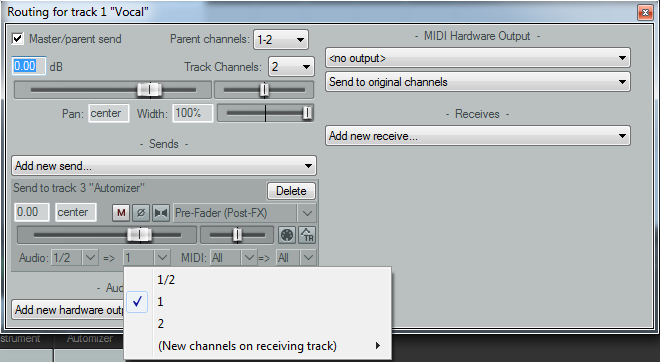
1. Under *“Sends”*, click add new sends, and select *“3:Automizer”.*



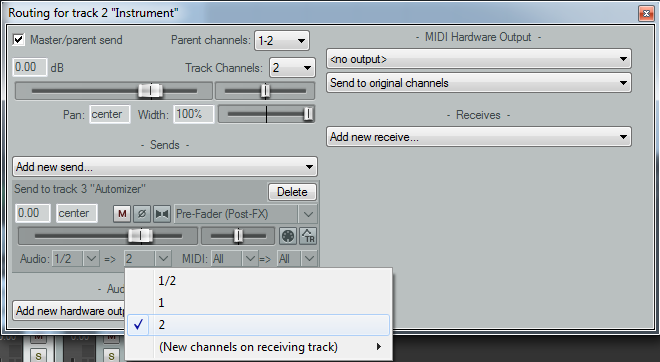
1. Then, click on the *“Post-Fader (Post-FX)”* button and select *“Pre-Fader (Post-FX)”*.



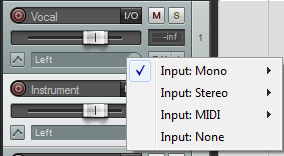
1. Click the middle *“1/2”* button and select *“1”.*



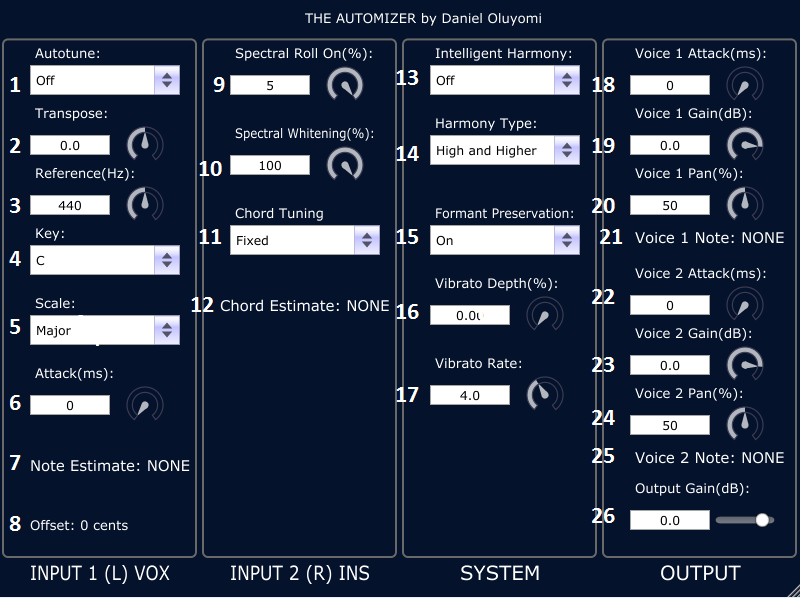
* Click on the I/O button on the instrument track and repeat the first two steps presented.
* On the third step, select *“2”* instead.



* The routing is now almost setup. For live use, on the vocal track, choose the microphone input that the vocal will be using. Do the same steps for the instrument track.



* Now select the FX button on the “*Automizer*” track where a list of all available software plug-ins to choose from will be displayed.
* Double click on *“VST: Automizer (Daniel Ayo)”* and the plug-in’s GUI will be displayed.
* **Ensure that the Recording monitor button is enabled on the vocal track in order to hear the output of the Automizer in real-time.**

****

**Automizer Controls:**

1. **Autotune: ON/OFF** - Switches the pitch correction component on or off. When off, the key, scale, reference and attack parameters are non-functional (disabled).
2. **Transpose: (+/- 12.0) -** Allows the user to adjust the frequency of the vocal input prior to pitch correction. A value of zero means that the signal is not transposed. This allows the possibility of generating diatonic harmonies based on the selected scale provided that the intelligent harmony parameter is switched off. Setting the transposition amount to a point in between intervals is important. For example, if you want to shift up a third, use a transpose setting of 3.5 semitones. Therefore the note will snap to a major third (4 semitones) or minor third (3 semitones) depending on the scale selected.
3. **Reference (Hz): (430-450)** – Standard tuning frequency. Default setting is 440Hz.
4. **Key: (C-B)** – Used to select the tonic of the input audio.
5. **Scale:** - The audio is pitched to the chosen scale notes.
6. **Attack: (0-100 ms)** – Determines the smoothness of the change in pitch. Smaller values cause the pitch shift to happen immediately.
7. **Note Estimate:** Gives visual feedback in form of a MIDI note representation of the currently detected pitch i.e. A4 = 440Hz.
8. **Offset:** Gives visual feedback on how close the detected pitch is to the nearest semitone.
9. **Spectral Roll On: (0-5%)** – Minimises low frequency noise. This parameter looks at the cumulative energy spectrum for each audio frame and sets frequency bins below the roll on value \* the total energy to zero. A value of 0 means no bins will be altered.
10. **Spectral Whitening: (0 – 100%)** – Spectral whitening is the process of making the magnitude frequency spectrum uniform. This parameter is should be experimented with when the chord estimate is not accurate.
11. **Chord Tuning:** - To improve chord detection, an attempt to automatically detect the standard tuning frequency has been implemented. It is advised however, to use the default setting of fixed.
12. **Chord Estimate:** - Gives a visual of the detected chord per frame. There will be periods where the wrong chord is detected. The chord detection algorithm capable of detecting up to 60 chord types. Major, Minor, Augmented, Diminished, Suspended. ONLY TRIADS CAN BE DETECTED.
13. **Intelligent Harmony: (ON/OFF)** – Switches the backing vocal harmonies on or off.
14. **Harmony Type**: - There are four variations on the harmony used to accompany the main vocal.

* **High and Higher –** Voice 1 = 3rd Above, Voice 2 = 5th Above
* **Low and Lower –** Voice 1 = 4th Below, Voice 2 =6th Below
* **Higher and Lower –** Voice 1 = 5th Above, Voice 2 =6th Below
* **High and Low -** Voice 1 = 3rd Above, Voice 2 =4th Below

1. **Formant Preservation: (ON/OFF)** – With the parameter off, the formants of the vocal input are raised and lowered with the pitch. This is used for weird vocal effects. With the parameter on, the formants are maintained keeping the characteristics of the audio input.
2. **Vibrato Depth:** - How strong the vibrato is on the voices is dependent on this parameter. When set to zero, there is no audible vibrato effect on the voices.
3. **Vibrato Rate: (0.1-10Hz)** – This parameter affects the rate of the pitch fluctuations.
4. **Voice 1 Attack: (0-100 ms)** - Determines the smoothness of the change in pitch from the original note detected at the input. Smaller values cause the pitch shift to happen immediately.
5. **Voice 1 Gain:** - The output volume of the first voice.
6. **Voice 1 Pan:** - Sets the stereo position for voice 1
7. **Voice 1 Note:** - This visual parameter shows the MIDI note representation of the harmonised note.
8. **Voice 2 Attack: (0-100 ms)** - Determines the smoothness of the change in pitch from the original note detected at the input. Smaller values cause the pitch shift to happen immediately.
9. **Voice 2 Gain:** - The output volume of the second voice.
10. **Voice 2 Pan:** - Sets the stereo position for voice 2.
11. **Voice 2 Note:** - This visual parameter shows the MIDI note representation of the harmonised note.
12. **Output Gain:** - The main output volume of the software plug-in.

**Compiling the Git hub source code (Windows Only)**

1. Download and install the latest free version of Microsoft Visual Studio from here:

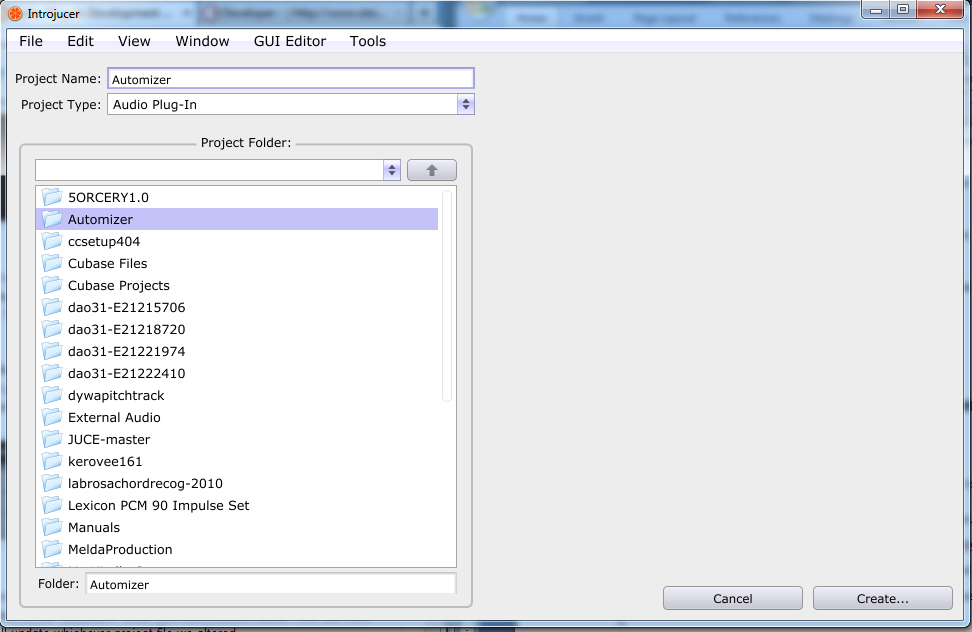
<https://www.visualstudio.com/downloads/download-visual-studio-vs>

1. The following external libraries must be installed for a successful build:

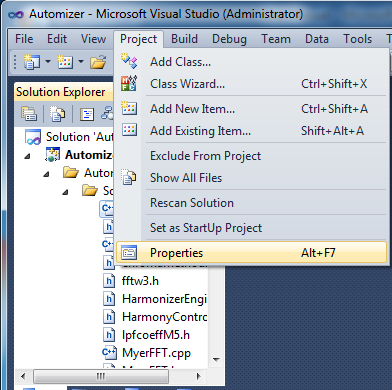
* The KissFFT library which can be downloaded here: <https://sourceforge.net/projects/kissfft/>
* The C++ boost library which can be downloaded here: <http://www.boost.org/users/download/>
* The latest JUCE framework which can be downloaded here: <https://www.juce.com/get-juce>
* The VST SDK which can be obtained here: <http://www.steinberg.net/en/company/developer.html>
* A plug-in host that supports VSTs such as REAPER or Steinberg Cubase.
* I would recommend extracting the files and storing them in a sensible place such as “C:/SDKs/….”

The next steps will show how to compile the source code using the JUCE library.

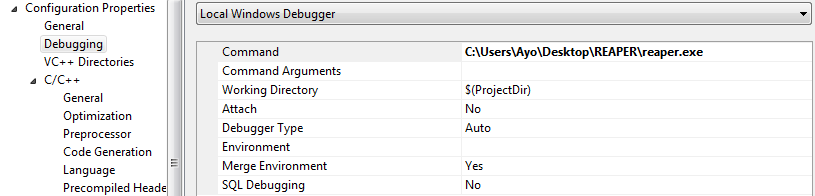
1. Navigate within the JUCE folder to the Projucer application. Open the Introjucer solution file, then build the solution. When it is done, there will be a “Debug” folder in the same Visual Studio project. Click on Introjucer.exe.
2. Create a new project and name it “**Automizer**” (VERY IMPORTANT) with the project type: **Audio Plug-in** and then click create.



1. On the next page, click on “Save Project and Open in Visual Studio...” and Visual Studio will open the Automizer project.
2. Next, locate the GitHub source code within the supporting files folder and copy and paste into the source folder of the newly created Automizer folder. Overwrite all current files. Furthermore, right click the source folder and select add, then existing item and select all the files that have been previously copied and pasted in the new Automizer folder.
3. Once the files have been successfully copied, click on “Project” in the ribbon header and select properties.



1. Under configuration properties, click on debugging, then under the command block, write the location of the exe file of your particular plug-in host.



1. Press F5 to debug the project using REAPER. To include the compiled VST follow the steps mentioned in page 6.
2. Within the configuration properties, under C/C++, click on General then edit the “Additional Include Directories” section to include the correct file path for the downloaded VST folders, and additional libraries for the KissFFT and boost respectively.