User’s Manual

September 14, 2005

CHAPTER 1: INTRODUCTION 2
System Requirements 2
Installation 2
Important Note 2

CHAPTER 2: GETTING STARTED 3
Running the Plug-In 3
Loading a WAV File 4

CHAPTER 3: MAIN PARAMETERS 5
Displacement 5
Random Displacement 6
Chunk Size 6
Chunk Size Interpolation 6
Gap Size 7
Gap Size Interpolation 7
Pitch 7
Pitch Interpolation 8
Pitch Rounding 8
Gain 8
Gain Interpolation 8

CHAPTER 4: BUTTON-PARAMETERS 9
Load 9
Random 9
Play 9
Reverse 10
Envelope 10

CHAPTER 5: MIDI PLAYBACK 11
Controlling Playback 11
Recording Playback 11

CHAPTER 1: INTRODUCTION

This manual is intended to instruct the reader as to how to operate the Stutter VSTi plug-in within Cubase.[1] As you read through this manual, you will learn how to install the program, as well as how to effectively use the parameters within the plug-in.

Please note that the following system requirements represent on what systems the plug-in has been tested. Although it may work on other platforms, the plug-in has only been tested within a Windows environment.

System Requirements

Windows 2000/XP
Cubase SL/SX
512 Mb RAM

Installation

Upon being provided with a zipped folder containing the plug-in files, unzip the folder to any location on you hard disk. You will be provided with two files named "Stutterer.dll" and "libsndfile.dll". It is important that you place the "Stutterer.dll" file within your VST Plug-ins folder, and the "libsndfile.dll" within Cubase's main program folder. Consult your Cubase manual for details on your VST Plug-ins folder.

Important Note

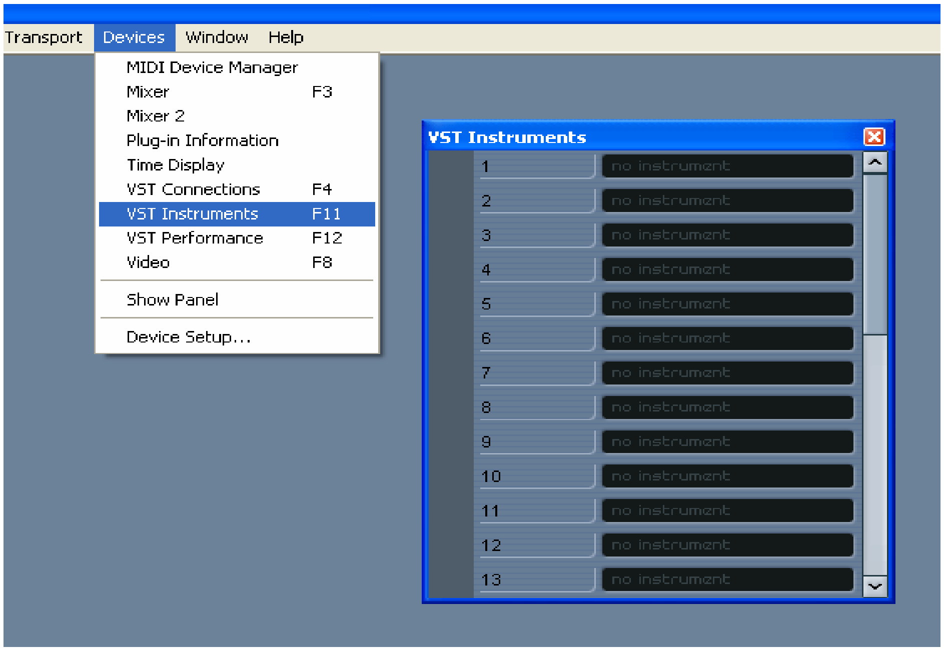
It is important to note that the screenshots shown throughout this manual are of the plug-in within the Cubase SL3 environment. Most versions of Cubase should appear similar in nature – but if you are using the plug-in with a host other than Cubase, please consult your user documentation for details on running VST-Instruments.

[1] VSTi means "Virtual Studio Technology Instrument", and is discussed in more detail in the Final Report, Section 4.0.

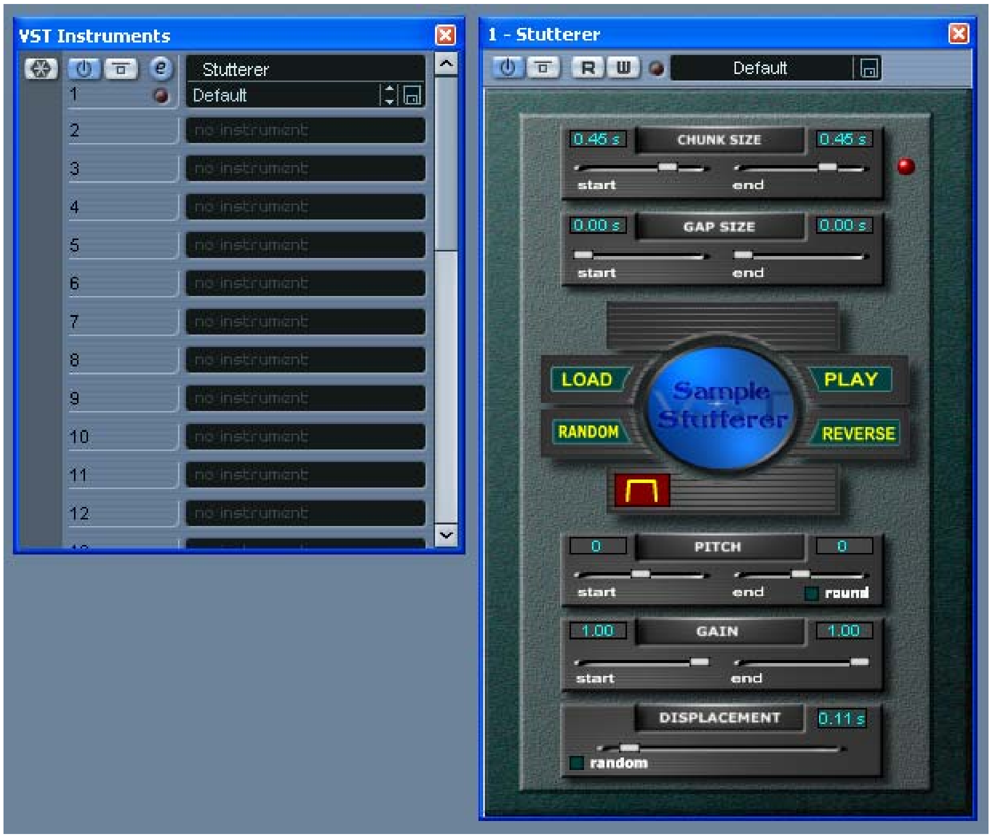
CHAPTER 2: GETTING STARTED

Running the Plug-In

Once you have followed the installation instructions, you will need to start up Cubase. Once Cubase has completely loaded and you have selected a template with which to work, select "VST Instruments" from the "Devices" menu, as shown in the following screenshot:



Once the VST Instruments dialog appears choose the Stutterer plug-in from the list of plug-ins in the first slot. Your screen should then look like the following:



You may need to follow the instructions in your Cubase manual for loading a Virtual Instrument, but you will typically link the Stutterer to a MIDI track. These details can be host-specific and will not be given here. Once the plug-in is loaded, you are ready to use the Stutterer.

Loading a WAV File

In order to use the plug-in, you first need to supply a WAV file.[2] The audio supplied audio stream is then stored in a buffer for future processing.

Once the file has been loaded, you could test the plug-in by pressing the "PLAY" button. You will then hear your audio stream as manipulated by the default parameter values. The next chapter of this manual will describe each parameter in detail.

[2] It is important to note that this program is intended for small audio files. Although it will accept larger ones, supplying an entire song to the plug-in is not ideal.

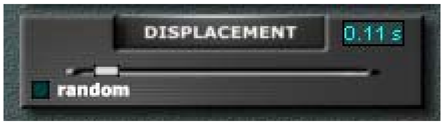
CHAPTER 3: MAIN PARAMETERS

This section of the manual will describe each user-editable parameter in detail, but is limited to those parameters controlled by sliders (and related buttons). The next chapter will discuss the parameters controlled by the buttons found on the middle of the interface.

Any parameter's value may be changed by using any of the sliders found on the interface itself.

Displacement

The following image shows the Displacement parameter as found on the interface of the plug-in:

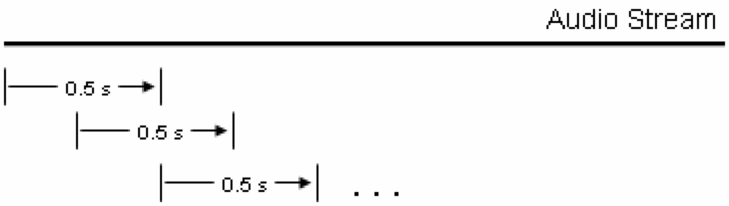


Although Displacement is arguably the most important of all the parameters, the slider for this parameter is found at the bottom of the interface.

Displacement is measured in fractions of a second and ranges from 0.01 to 1 second.[3]

The term "Displacement" refers to how much the starting cursor will *displace* after a chunk of samples has been played (and is measured in fractions of a second). For example, if the chunk size were 0.5 seconds, and the displacement were 0.25 seconds, then after each playback of a half-second chunk, the starting cursor will move forwards and then the playback of the next chunk *starting at the current position of the cursor* will begin.

The following figure gives a crude representation of how playback would "look" with a chunk size of 0.5 seconds and a displacement of 0.25 seconds:



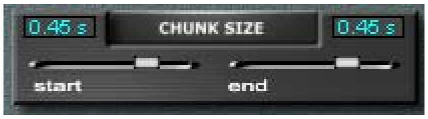
[3] A Displacement value of "0" would have the effect of an infinite loop of audio playback.

Random Displacement

The "random" button found directly below the Displacement slider refers to a random displacement of the starting cursor. The button is red in color when random displacement is activated (and green when deactivated). When activated, the starting point of each chunk of audio will be at some random location within the audio stream.

Chunk Size

The following image shows the Chunk Size parameter as found on the interface of the plug-in:



The Chunk Size parameter is located at the very top of the interface.

Chunk Size is measured in fractions of a second and ranges from -1 to 1 second.

"Chunk Size" refers to the length of each slice of the audio stream that will be played. Any positive chunk size will be played in the natural (forward) direction, *while any negative chunk size will be played in the reverse direction*.

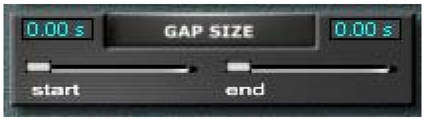
Chunk Size Interpolation

There are two sliders for chunk manipulation. This allows for interpolation between two different chunk sizes. The "start" slider specifies the size of the very first chunk to be played, while the "end" slider specifies the size of the very last chunk to be played. The size of intermediate chunks is interpolated based on the start and end values specified by the user. If no interpolation is desired, it is up to the user to ensure that both sliders are at the same position.

If interpolation is desirable, and if the first slider's value is negative, then chunks will be played in the reverse direction until (potentially) a positive chunk size value is reached during the interpolation.

Gap Size

The following image shows the Gap Size parameter as found on the interface of the plug-in:



Gap Size is measured in fractions of a second and ranges from 0 to 1 second.

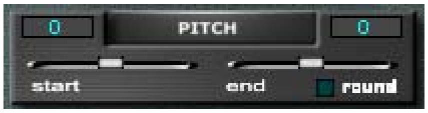
"Gap Size" refers to the length of silence that is to be inserted in between any two successive chunks of audio. If silence is undesirable, the gap sliders should both be placed at the left-most position.

Gap Size Interpolation

Like most parameters, interpolation is also an option with Gap Size. The "start" slider specifies the size of the very first gap of silence to be played, while the "end" slider specifies the size of the very last gap of silence to be played. The size of intermediate gaps of silence is interpolated based on the start and end values specified by the user. If no interpolation is desired, it is up to the user to ensure that both sliders are at the same position.

Pitch

The following image shows the Pitch parameter as found on the interface of the plug-in:



Pitch is manipulated in semitone increments, and ranges from -24 semitones (-2 octaves) to 24 semitones (+2 octaves).

"Pitch" refers to the relative pitch of the current sample chunk being played. If, for example, the pitch sliders are left at their default positions (0), then there will be no pitch difference in the playback of the audio stream. Thus it is recommended that the pitch sliders remain untouched if no pitch manipulation is desired.

Pitch Interpolation

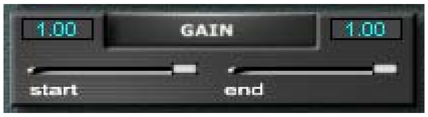
Interpolation is also an option with respect to the Pitch parameter. The "start" slider specifies the pitch of the very first audio chunk to be played, while the "end" slider specifies the pitch of the very last audio chunk to be played. The pitch of intermediate audio chunks is interpolated based on the start and end values specified by the user. If no interpolation is desired, it is up to the user to ensure that both sliders are at the same position. For example, if the user wishes to play the audio stream at one octave above the original pitch throughout the entire playback, then both sliders should be set to "12" semitones.

Pitch Rounding

This option can be toggled on/off via the "round" button found on the Pitch parameter panel. When "on" (i.e. when the button is red), any calculated pitch will be rounded to the nearest semitone (during interpolation only). When "off" (i.e. when the button is green), the calculated (floating-point) pitch value (as calculated during interpolation) will be used as the pitch for the current audio chunk.

Gain

The following image shows the Gain parameter as found on the interface of the plug-in:



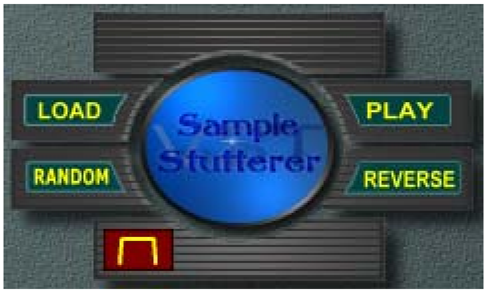
Gain values are simply between 0.00 and 1.00, and refer to how much the current sample chunk's volume should be manipulated. For example, if the current Gain value is 1.00, then the sample values throughout the current audio chunk will be multiplied by 1.00, thus resulting in no change. However, if the current Gain value is 0.5, then each sample value throughout the current audio chunk would be multiplied by 0.5, thus halving the amplitude (which in turn implies half the normal playback volume).

Gain Interpolation

Interpolation is also an option with respect to the Gain parameter. The "start" slider specifies the gain of the very first audio chunk to be played, while the "end" slider specifies the gain of the very last audio chunk to be played. The gain of intermediate audio chunks is interpolated based on the start and end values specified by the user. If no interpolation is desired, it is up to the user to ensure that both sliders are at the same position. For example, if the user wishes to play the audio stream at half the volume throughout the entire audio playback, then both the start and end Gain sliders should be set to 0.5.

CHAPTER 4: BUTTON-PARAMETERS

This section of the manual will describe the parameters and functions controlled by the buttons found on the middle of the interface. The following screenshot shows the five buttons whose functionality will be discussed:



Load

When pressed, the Load button allows the user to navigate to a desired WAV file. The file is then loaded into memory. Any playback and/or editing of other parameters are performed on the audio stream loaded by the user.

The user may load any other file at any time (by pressing the Load button again); however, only the most recently loaded file is stored in memory.

Random

The Random button allows the randomization of every parameter to be performed. This is useful for discovering unintended and diverse effects. A new set of random parameter values is computed every time the Random button is pressed.

Play

The Play button triggers playback of the loaded audio stream as manipulated by all of the parameters. This playback is just an illustration of how the current parameter settings affect the current audio stream, and is not intended for recording by any host program (see Chapter 4 for MIDI details).

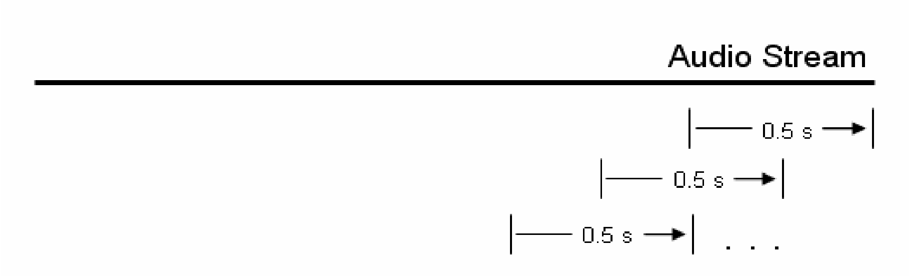
The Play button turns red in color when playback is in process. Once the playback has completed, the Play button again turns to its normal state (i.e. the background becomes green in color).

Reverse

When the Reverse button is pressed, audio playback starts at the end of the stream and works towards the beginning. In this case, the Displacement parameter represents how much the starting cursor travels towards the beginning of the stream after the playback of each sample chunk.

When pressed, the Reverse button becomes red in color. When this option is disabled, the normal (green) color becomes visible.

The following figure gives a crude representation of how playback would "look" with a chunk size of 0.5 seconds and a displacement of 0.25 seconds, if the reverse option were enabled:



Envelope

The Envelope button (shown in red in the above display) triggers enveloping for each sample chunk. When selected, a simple linear envelope is applied to each sample chunk to minimize any clicks or pops heard during playback. If such noise is desirable, then the option can be deselected.

When enveloping in enabled, the Envelope button is red in color, and becomes green when disabled.

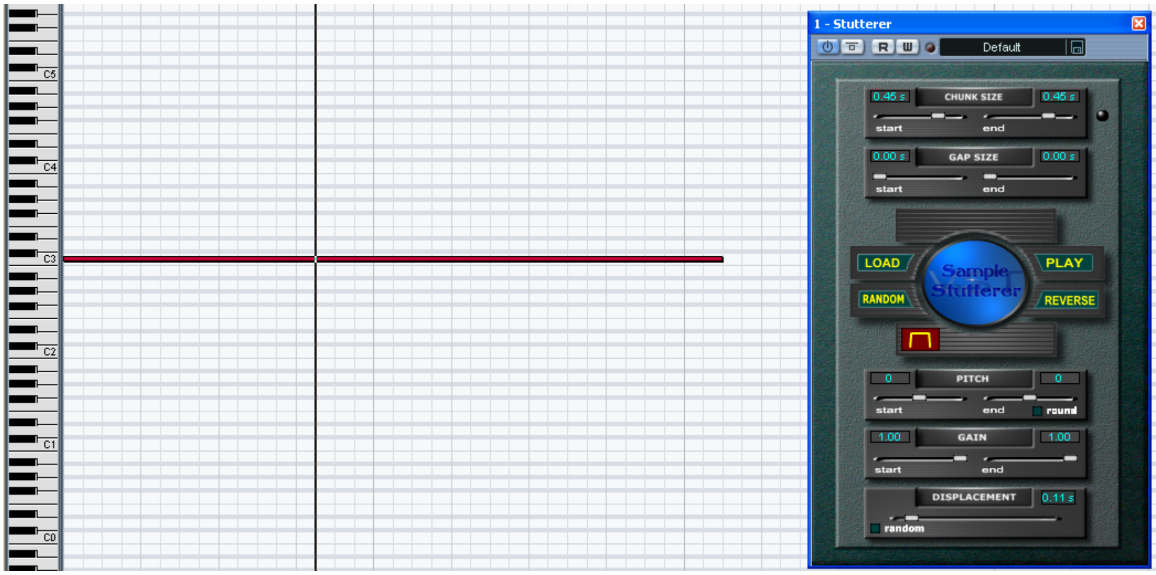
CHAPTER 5: MIDI PLAYBACK

This chapter discusses the details of using MIDI to control the playback of the Stutterer plug-in. Some general screenshots will be given; and although the general idea is the same, it is important to note that using MIDI is host-specific.

Controlling Playback

Controlling playback of the plug-in using MIDI is trivial. Any single MIDI note may be used to trigger playback. The playback of the manipulated audio stream will last the duration of the chosen note. Hence, some experimentation may be needed to determine how long a MIDI note's duration should be.

The following screenshot gives an example of how what was just described looks in Cubase:



CHUNK SIZE 0.45 s 0.45 s

GAP SIZE 0.00 s 0.00 s

PITCH 0 0 round

GAIN 1.00 1.00

DISPLACEMENT 0.11 s random

When reached, the single MIDI note shown at "C3" triggers the playback of the Stutterer plug-in. As previously mentioned, the playback will last the duration of the note.

Recording Playback

Since recording is host-specific, recording details will not be given here. However, it is sufficient to say that once you have MIDI triggering playback from the plug-in correctly, recording then becomes trivial. For example, in Cubase you simply arm an audio track and press the "Record" button to record the output of the plug-in into the audio-track.

It is recommended that you consult your host's operating manual for further details.