

# VSTForx 1.0.0 Manual

Johannes Unger

June 2, 2013

# Contents

<b>1</b>	<b>Installation</b>	<b>2</b>
1.1	System requirements . . . . .	2
1.1.1	Windows . . . . .	2
1.2	Installation . . . . .	3
1.3	Configuration . . . . .	3
1.3.1	The Scan . . . . .	3
1.3.2	Window Options . . . . .	5
<b>2</b>	<b>The Editor</b>	<b>6</b>
2.1	The Main View . . . . .	6
2.2	Moving And Connecting . . . . .	6
2.2.1	The Corona Field . . . . .	6
2.3	Main Entry/Exit . . . . .	7
<b>3</b>	<b>Parameter</b>	<b>8</b>
3.1	Special Parameter Types . . . . .	8
3.2	Operators . . . . .	8
<b>4</b>	<b>Modules</b>	<b>9</b>
4.1	Common Module Parameters . . . . .	9
4.2	Volume . . . . .	9
4.3	Pan . . . . .	10
4.4	Input / Output Switches . . . . .	10
4.5	Input/Output Step Switches . . . . .	10
4.6	Peak Transformer . . . . .	11
4.7	ADSR Trigger . . . . .	11
4.8	MIDI Receiver . . . . .	12
4.9	Appendix . . . . .	12

# Chapter 1

## Installation

### 1.1 System requirements

#### 1.1.1 Windows

- Windows XP(SP3)/Vista/7
- Pentium/Athlon Prozessor 500Mhz or better
- 512MB RAM
- about 20MB free disk space
- Windows XP(SP3)/Vista/7 compatible soundcard
- a host application with VST2.x support

## 1.2 Installation

Just extract the whole content of the .zip file into the vst-plugin folder of your host application. Open the host application and check that VSTForx was loaded successfully. You should find two VSTForx plugin variants:

- vstforx (the effect)
- vstforxInstrument (the instrument)

## 1.3 Configuration

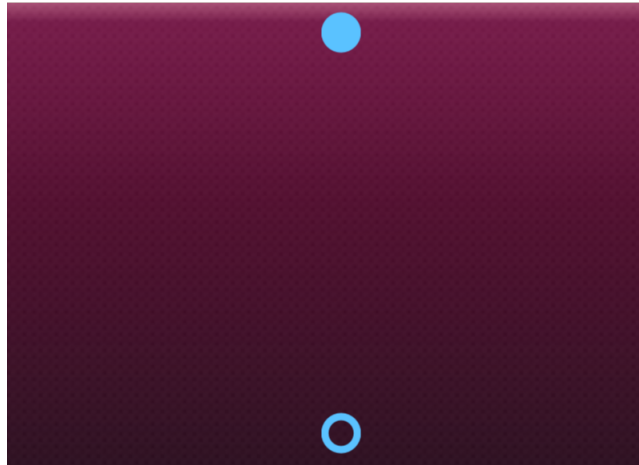


Figure 1.1: the editor

Load VSTForx into your host application. Open the editor (figure 1.1). To open the setup dialog use the *"Open Setup Dialog"* entry in the context menu of the main view.

### 1.3.1 The Scan

**Adding Pluginfolders** With the buttons *'add Directory'*, *'change Directory'* and *'remove Directory'* you are able to configure the search folders of the VSTForx database.

**The Scan Process** While the scanning process, VSTForx searches the given folders for VST plugins. Depending on the number of plugins the scan process may take a while. An extra window will show you the results of the current scan (figure 1.3).

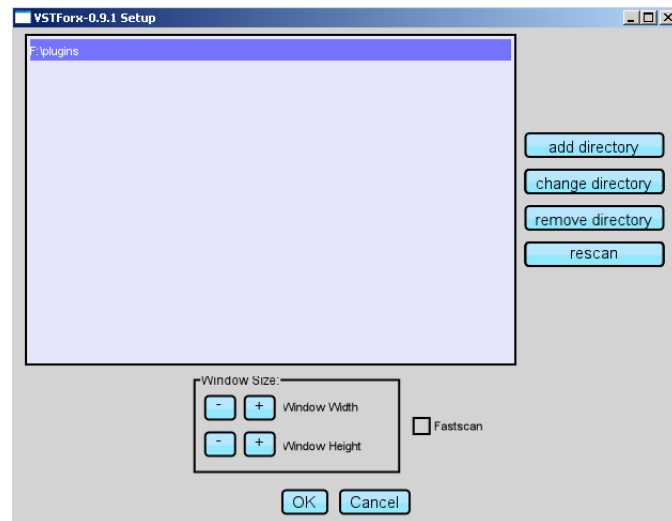


Figure 1.2: the setup dialog

**Heads Up:** don't forget to repeat this process when your plugin folder content has changed!



Figure 1.3: the scan result window

**Fast Scan** Is the "fast scan" option activated, the scan process recognizes a plugin only with the filename. The scan process is faster now but there is a lack of detailed informations about the scanned plugins (e.g. is it a instrument or not). Besides that there is a risk of recognizing a system .dll as a plugin.

### **1.3.2 Window Options**

With the "*Window Size*" buttons you are able to change the editor size of the VSTForx editor.

## Chapter 2

# The Editor

### 2.1 The Main View

The main view is the largest part of the VSTForx editor, use it to place and control plugins, modules and knobs.

### 2.2 Moving And Connecting

#### 2.2.1 The Corona Field

The problem, which appears especially when using knobs, is that we can do three actions: moving, using and connecting. But we have only one mouse button! *(The right mouse button is for translating the view. Modifier keys such as “shift” or “ctrl” are problematic because there is lack of keyboard support in some host applications.)*

The solution to this problem is the “corona” field which appears when you hover over an object. There are “two” corona types:

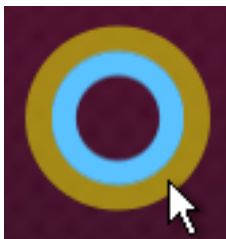


Figure 2.1: the node corona

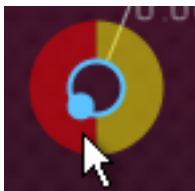


Figure 2.2: the knob corona

**The Node Corona** (figure 2.1) The node corona which is a single (yellow) field around a node. With draggin this yellow zone you are able to move the node around. Connecting will be approached by dragging the mouse over the node.

**The Knob Corona** (figure 2.2) The knob corona is a double sided field (red/yellow) around a knob. The yellow side is still for moving the knob. The red side is for connecting the knob with another knob. To manipulate (use) the knob just use the knob as expected.

## 2.3 Main Entry/Exit

**Main Entry** (●)

From here comes the audio signal of your host application.

**Main Exit** (○)

Here will the audio signal sent back to the host application.



## Chapter 3

# Parameter

Every effect and every module is controlled by parameters. All of these parameters have a knob representation which can be placed and controlled on the main view. With the knobs you are able to bind parameters together, so that every parameter changing will be transmitted to its neighbours.

### 3.1 Special Parameter Types

**Free** Free parameters are not related to any module or plugin in VSTForx. You can add as many free knobs on the view as you want.

**Host** The host parameter appears in your host application as VSTForx parameters. These parameters are the connection between the host application and VSTForx. Use these parameters e.g. for host automation.

### 3.2 Operators

Parameter Connection Operators are modifiers for parameter value transmissions. To add an operator, move the mouse over a connection and open the context menu. Select one operator from the submenu entry *add operator*. Some operators are controlled by parameters too. You can access them using the scene browser.

Available operators:

- Inverse
- Exp/Log
- Log/Exp
- Offset

# Chapter 4

## Modules

VSTForx knows three types of modules:

- signal processor
- signal switch
- signal transformer

### 4.1 Common Module Parameters

#### MIDI

Every modules/plugins which are able to receive midi data have an additional 'midi-channel' parameter:

parameter	range	function
<i>midi_channel</i>	<i>all, 0 ... 16</i>	<i>selects the receiving midi channel</i>

### 4.2 Volume

The Volume module is a signal processor. It contains a single parameter which allows to control the volume of the incoming signal.

#### Module Parameter

parameter	range	function
<i>Volume</i>	<i>0 ... 1</i>	<i>controls the volume</i>

## 4.3 Pan

Like the Volume module, the Pan module is a signal processor with a single parameter. Use it to control the panning of the incoming signal.

### Modul Parameter

parameter	range	function
<i>Pan</i>	<i>0 ... 1</i>	<i>controls the pan</i>

## 4.4 Input / Output Switches

Input / Output Switches are able to switch between several states while every state belongs to an input respectively to an output. An "Input Switch" for example can switch several inputs to one output. A "selector" parameter controls the switch state. Furthermore several parameters control the "fade-in / fade-out" behaviour.

### Adding States

To add an additional switch state you can find a respective option in the context menu of any switch object.

### Module Parameters

parameter	range	function
<i>selector switch</i>	<i>0 ... 1</i>	<i>the state selector</i>
<i>fade-in duration x*</i>	<i>0ms ... 1000ms</i>	<i>"fadein" duration x</i>
<i>fade-in curve type x*</i>	<i>type1 ... type**</i>	<i>"fadein" slope type x</i>
<i>fade-out duration x*</i>	<i>0ms ... 1000ms</i>	<i>"fadeout" duration x</i>
<i>fade-out curve type x*</i>	<i>type1 ... type**</i>	<i>"fadeout" slope type x</i>

\*: x=the related state

\*\*: see figure 4.1 on page 12

## 4.5 Input/Output Step Switches

Unlike the "I/O Switches" the "Step Switch" state is not controlled by a parameter. The states will be iterated automatically. Either with a *fixed* time per state or *synced* to the beat of the current DAW.

**Heads Up!** In sync mode the state position will be reseted on incoming "play" event of the DAW.

## Adding States

To add an additional switch state you can find a respective option in the context menu of any switch object.

## Module Parameters

parameter	range	function
<i>Step Type</i>	<i>fix / sync</i>	<i>Der Step Modus</i>
<i>Step x duration*</i>	<i>10ms...1000ms</i> or <i>1/128...1/1**</i>	<i>duration x</i>
<i>fade-in duration x*</i>	<i>0ms ... 1000ms</i>	<i>"fadein" duration x</i>
<i>fade-in curve type x*</i>	<i>type1 ... type***</i>	<i>"fadein" slope type x</i>
<i>fade-out duration x*</i>	<i>0ms ... 1000ms</i>	<i>"fadeout" duration x</i>
<i>fade-out curve type x*</i>	<i>type1 ... type***</i>	<i>"fadeout" slope type x</i>

\*: x=the related state

\*\* : see figure 4.1 on page 12

## 4.6 Peak Transformer

The peak transformer transforms incoming audio peaks into an output parameter value.

## Module Parameters

parameter	range	function
<i>out</i>	<i>0 ... 1</i>	<i>the output value</i>
<i>offset</i>	<i>0 ... 1</i>	<i>offset to the output value</i>

## 4.7 ADSR Trigger

The ADSR Trigger generates an ADSR (figure 4.2 on page 13) sequence triggered by an audio signal. This sequence will be transformed into an output parameter value. Thus it's possible to control parameters (or rather the connected units) with any audio input.

## Module Parameters

parameter	range	function
<i>out</i>	<i>0 ... 1</i>	<i>the output value</i>
<i>x level*</i>	<i>0 ... 1</i>	<i>final value of x</i>
<i>x duration*</i>	<i>0s ... 5s</i>	<i>duration of x</i>
<i>x slope type*</i>	<i>type1 ... type6**</i>	<i>slope type x</i>
<i>trigger_threshold</i>	<i>0 ... 1</i>	<i>trigger threshold</i>
<i>trigger_hold</i>	<i>0 ... 5s</i>	<i>determines the minimum retain duration of the trigger</i>

\*: x=the related state: attack, decay, sustain, release

\*\* : see figure 4.1 on page 12

## 4.8 MIDI Receiver

The MIDI Receiver is a transformer module. It transforms MIDI signals into parameter values.

### Module Parameters

127 MIDI control + pitchbend parameter.

## 4.9 Appendix

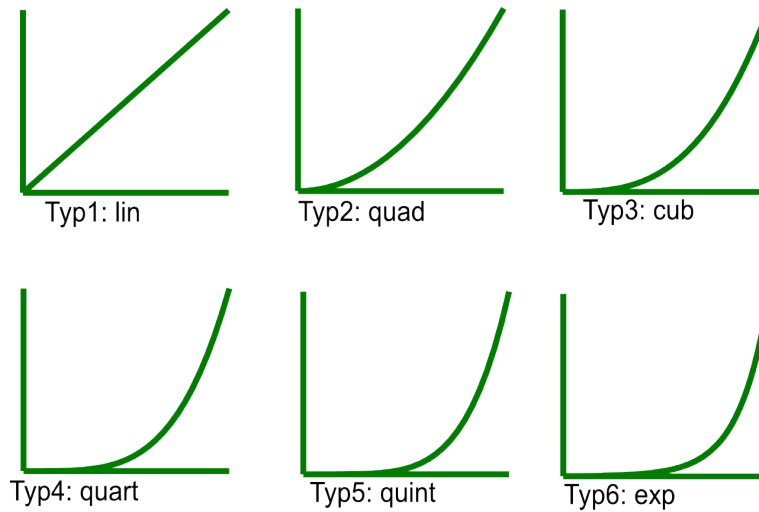


Figure 4.1: the six slope types of VSTForx

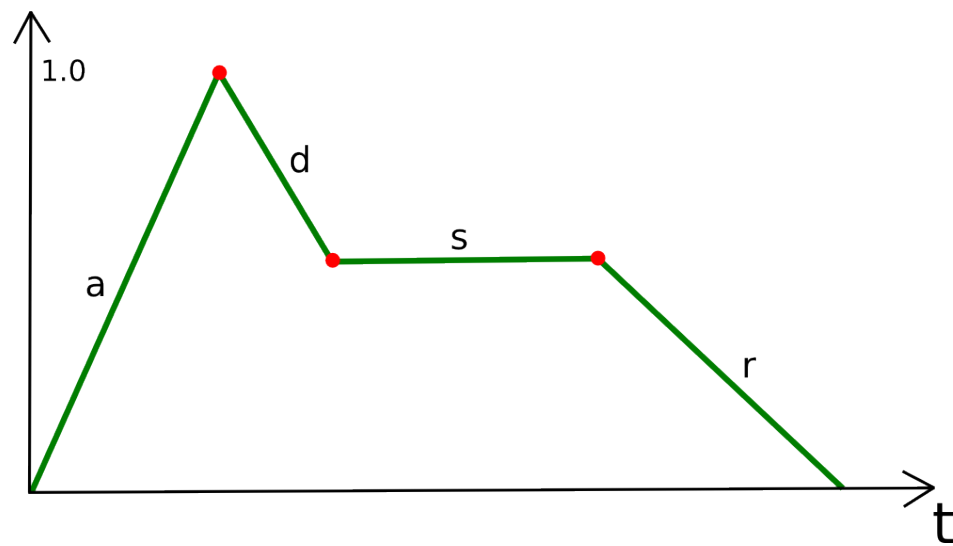


Figure 4.2: example of an ADSR curve