

# SYNTHESIZER **largo**

**User Manual**

English

Drive

Resonance

Cutoff

 **waldorf**

# Foreword

Thank you for purchasing the Waldorf Largo. You now own a synthesizer with real Waldorf DNA featuring a wide range of unique sounds of high quality.

## What to read?

The biggest problem with any manual is to find a way to cover both the needs of an absolute expert and a beginner alike. There are people who read a manual cover to cover while others don't even touch it. The latter is the worst choice, especially when the manual describes a Waldorf instrument.

If you decide to read the following manual, we promise you a lot of fun while reading about and working with the Waldorf Largo.

Your Waldorf Team

## Hint

Waldorf Music GmbH is not liable for any erroneous information contained in this manual. The contents of this manual may be updated at any time without prior notice. We made every effort to ensure the information herein is accurate and that the manual contains no contradictory information. Waldorf Music GmbH extends no liabilities in regard to this manual other than those required by local law.

This manual or any portion of it may not be reproduced in any form without the manufacturer's written consent.

Waldorf Music GmbH, Neustrasse 12, D-53498 Waldorf, Germany

## Largo Development Team

Software: Wolfram Franke, Stefan Stenzel  
Design: Axel Hartmann  
Manual: Holger Steinbrink  
Betatest/ Sounddesign: Wolfram Franke, André Geisler,  
Boele Gerkes, Daniel Gill,  
Achim Gratz, Dajan Izzo, Till  
Kopper, Dirk Krause, Jay de  
Miceli, Niels Moseley, Dr. Georg  
Müller, Kai Niggemann, Don  
Petersen, Holger Steinbrink, Dr.  
Stefan Tripler  
Revision: 1.0, May 2009



Please visit our website [www.waldorfmusic.de](http://www.waldorfmusic.de)  
Maybe you can download a newer version for your  
Largo.

## We would like to thank

Willie Eckl, Joachim Flor, Michael von Garnier, Frank  
Lauterbach, Frank Schneider, Kurt “Lu” Wangard, 吴海彬  
and anyone we have forgotten.

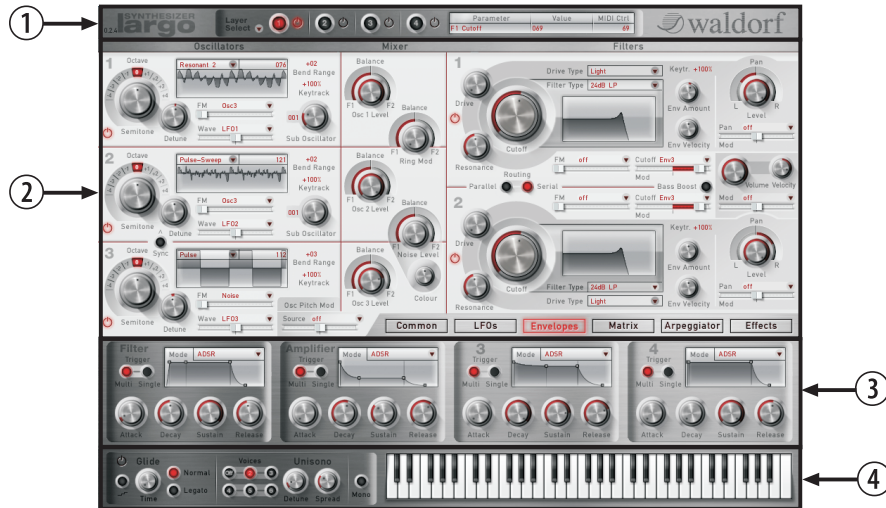
# Content

<b>Foreword</b> .....	<b>3</b>
<b>Content</b> .....	<b>5</b>
<b>Overview</b> .....	<b>6</b>
<b>Introduction</b> .....	<b>7</b>
<b>Installation</b> .....	<b>8</b>
System Requirements for Windows .....	8
Installation under Windows.....	8
System Requirements for Mac OS X .....	9
Installation under Mac OS X.....	9
Activation of Largo .....	10
<b>Basic Operation</b> .....	<b>11</b>
<b>Sound Parameter</b> .....	<b>13</b>
Overview of Functions .....	13
The Head-up Section.....	13
The Layer Menu .....	14
Oscillator Section .....	15
Mixer Section .....	22

Filter Section.....	24
Amplifier Section .....	30
Parameter Module Section .....	31
Common Area .....	31
The LFOs .....	32
The Envelopes .....	35
The Modulation Matrix .....	37
The Arpeggiator .....	40
The Effects .....	47
Sound Parameter in the Keyboard Section .....	53
<b>Appendix</b> .....	<b>55</b>
Oscillators Introduction .....	55
Filter Introduction .....	62
Envelope Types.....	69
FM Sources, Modulation Sources & Destinations .....	72
Tips & Tricks.....	74
Controller Numbers .....	76
Glossary .....	81
Product Support.....	87

# Overview

## User Interface



① Head-up section

② Main parameter section

③ Module parameter section

④ Virtual Keyboard and Play parameter

# Introduction


## About this Manual


This manual was written to help you to become familiar with the Largo synthesizer. It will also aid experienced users with routine tasks.


To avoid confusion, the terminology in this manual is based on the Largo parameter names. You will find a glossary at the end of this manual; it explains the various terms used.

We also used a uniform set of symbols to show you topics of particular interest or significance. Important terms are highlighted in bold letters.

## Symbols

 **Caution** – The comments that follow this symbol will help you avoid errors and malfunctions.

 **Info** – Additional information on a given topic.

 **Instruction** – Follow these guidelines to execute a desired function.

 **Example** – Real-world examples to try out.

## Highlighted Control Features and Parameters

All of the Largo's buttons, controls and parameters are highlighted in **bold** letters throughout the manual.

Example:

- Turn the **Cutoff** dial.

The Largo's different modes and parameter pages are illustrated in a depiction of the display.

The value range of a continuous parameter is indicated from low to high with both values shown in italic letters, separated by three dots.

Example:


**Cutoff** *0...127*


# Installation

## System Requirements for Windows

In order to be able to use the Largo, you will need at least:

- A PC with a Pentium III or AMD Athlon 1 GHz processor or better, we recommend a fast CPU.
- 64 MB free RAM.
- Windows XP or newer.
- VST 2.0 compatible host application such as Cubase SX or newer. This must be correctly installed on your computer.


 Please also observe the system requirements of your host application!

 Largo can be installed under Windows VISTA 64. Please keep in mind that Largo will run as 32 bit application.

## Installation under Windows

Proceed as follows to install the Largo:

1. Start your computer and launch your operating system.
2. Insert the Largo CD-ROM into your DVD/CD-ROM drive. If you have enabled the Autostart function in Windows, the Installer will start automatically and you can proceed with Step 6 below. If not, please proceed as follows:
3. Launch the Explorer or open the “My Computer” window.
4. Double click on the icon for the drive that holds the Largo-CD-ROM.
5. Double click on the Largo Installer icon. This launches a special installation program.
6. Follow the on-screen instructions.

 After installing Largo you will have to activate the program on your computer Please refer to the chapter „Activation of Largo“.

## System Requirements for Mac OS X

In order to be able to use the Largo, you will need at least:


- Mac with 800 MHz G4 PowerPC processor or better.
- Mac OS X 10.3.9 or newer.
- 64 MB free RAM.

or

- Mac with 1.5 GHz Intel processor or better.
- Mac OS X 10.4 or newer.
- 128 MB of free RAM.
- VST 2.0 compatible host application such as Cubase SX / SL or Nuendo. This must be correctly installed on your computer.

or


- An AudioUnit 2.0 compatible host application such as Apple Logic or GarageBand. This must be correctly installed on your computer.

 Please also observe the system requirements of your host application!

## Installation under Mac OS X

Proceed as follows to install the Largo:

1. Quit all other applications so that you return to the Finder. Disable any system activity monitoring software or extension, in particular anti-virus software. Then insert the Largo CD into your computer's DVD/CD-ROM drive.
2. If you need to, double click on the Largo icon to open the CD window.
3. Double click on the Largo Installer icon to load the installation software. Follow the instructions on the screen.

 After installing Largo you will have to activate the program on your computer. Please refer to the chapter „Activation of Largo“.



## Activation of Largo

Largo uses a software based copy protection called eLicenser. Largo can only run with an activated license.

Whether you install Largo on a Mac or Windows computer you will need to activate the software.

1. As the last step in the installation, the License Control Center is opened.
2. Open the Wizards menu and select the License Download function.
3. Follow the instructions and enter the activation code that comes with your Largo software.

**i** You require an internet connection for license activation.

**i** If you own a eLicenser Key (e.g. Steinberg Key) you can transfer the eLicense of Largo permanently to this Key to use Largo on other computers. Please read the documentation of your eLicensert Key.

**i** Please note the “Read Me!” and “Important Changes” text files on the Largo CD-ROM, which contain additional information and/or any changes.

# Basic Operation

## Worth Knowing

Largo offers a sound structure with four layers. All four layers are always saved for each sound. For each bank, 128 sounds are saved. The four layers can be controlled either together via MIDI channel 5 or separately via MIDI channels 1, 2, 3 or 4. Each layer can have its own stereo output. In the **Common** menu, you can mix all layers together to one stereo output. You can also use this architecture to generate a multimode or surround sound.



Both Effect units are available for each layer without any restrictions.

## General Operation

Largo has been optimized for a screen resolution of at least 1024x768 pixels. If your resolution is smaller, you either have to scroll your window.

Largo has various on-screen controls, the behavior of which differs according to the Control Mode settings in the **Common** menu (Auto/Linear/Circular). The faders can be moved with greater accuracy by holding down the SHIFT key on your computer keyboard while moving your mouse. Clicking the corresponding fader or knob while holding down the CTRL key resets the parameters to the stored values.

## Control Elements

Using Largo's controls is simple. There are some different types of control elements:

### Dials



To set a value, click on the dial, hold down the mouse button and drag around the dial in a circle (**Control Mode** setting: Circular) or move the mouse up and down (**Control Mode** setting: Linear).

### Parameter Values



These allow to set a parameter value. Click and hold the mouse button on the value and drag the mouse up or down. Pressing Shift while holding the mouse button allows finer adjustment.

## Pop-up Menu



Click on the corresponding parameter to open a pop up menu where you can choose the desired parameter.

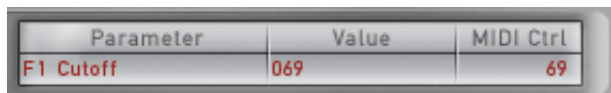
## Switches



Switches are represented by “LEDs”. If a function is active, its “LED” will be lit red and all others within the function block will be dark. To activate a function, click on its “LED”.

## Display

The information display shows parameter information and a brief description of the parameter function. In addition, MIDI controller allocations and the current numerical value of the parameter are displayed. If you hover over a parameter using the mouse, the corresponding information appears in the information display.



Parameter	Value	MIDI Ctrl
F1 Cutoff	069	69

# Sound Parameter

## Overview of Functions

The Waldorf Largo consists of numerous sound-shaping components. The following pages describe all parameters in detail. Additional information can be found in the Appendix on page 56.

You should know that the Largo consists of two different types of components for sound generation and sound shaping:

- Sound synthesis: Oscillators, Ring Modulator, Noise Generator, Filters, Amplifier, Effects. These modules represent the audio signal flow. Sound generation actually occurs within the Oscillators. They produce square, sawtooth, triangular, sine and wavetables. Noise can also be added to the mix. The Filter then shapes the sound by amplifying (boosting) or attenuating (dampening) certain frequencies. The Amplifier and the Effects are located at the end of the signal chain. They determine the overall volume of the signal and add some effects like chorus, flanger, delay etc.

- Modulators: LFOs, Envelopes, Modifiers, Modulation Matrix. These modules are called Modulators. The Modulators are designed to manipulate or modulate the sound generating components to add dynamics to sounds. The Low-frequency Oscillators (LFO) are designed for periodic or recurring modulations while Envelopes are normally used for modulations that occur once on each note. These generators are assigned to parameters through the Modulation Matrix and influence these parameters to alter a sound. In addition, the Modifier unit can process various mathematical operations and functions on the modulation signals.

## The Head-up Section



The Head-up section provides the global overview and includes the following elements:

1. The Layer menu button opens a pop-down menu containing various functions as loading, saving, copy & paste.
2. Layer Select indicates which of the four layers you are currently editing.

3. Layer On/Off indicates which of the four layers is/are currently active.

4. The information display shows parameter information and a brief description of the parameter function. In addition, MIDI controller allocations and the current numerical value of the parameter are displayed.

### The Layer Menu

Left click on the Layer menu button to open a pop-down menu with the following parameters:

- **Load Bank** opens a window to load a Largo soundbank.
- **Save Bank** opens a window to save the current Largo soundbank.
- **Load Sound** opens a window to load a single Largo sound program.
- **Save Sound** opens a window to save the current Largo sound program.
- **Copy Sound** copies the current sound program into the clipboard.
- **Paste Sound** pastes the sound program from the clipboard to the current sound program.
- **Recall Sound** recalls all sound edits and recover the original sound settings.
- **Init Sound** initializes the current sound and set all parameter to standard values.
- **Copy Layer** copies the current selected layer into the clipboard.
- **Paste Layer** pastes the layer from the clipboard to the current layer.
- **Recall Layer** recalls all layer sound edits and recover the original layer settings.
- **Init Layer** initializes the current layer and set all parameter to standard values
- **Recall Bank** recalls all sound edits and recover the original soundbank settings.
- **Save All Programs up to this Program** saves all sounds from sound program 1 to the current sound program as single sounds.
- **Save All Programs up to this Program as Audio Unit Preset (only Mac)** saves all sounds from sound program 1 to the current sound program as single AudioUnit presets.

**i** Please note that some host applications as Cubase 4 or 5 use an additional sound data management. It offers you an alternative way to load and save sounds. Please refer to the corresponding chapter in the manual of your host application.

\* **Working with Layers** – To set up a sound program with different layers please do the following: Select the desired sound program and use the **Copy Layer** function. Select the destination sound program and use the **Paste Layer** function to paste the new layer into the existing program.

## Oscillator Section

The Largo offers three oscillators that nearly use the same parameters for editing.



**i** A detailed introduction of the oscillators can be found in the Appendix.

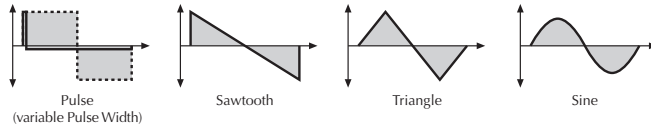
## Activating an Oscillator

Click on the Active button to switch the corresponding oscillator on or off.

**Shape** *Pulse / Saw / Triangle / Sine / Wavetables*

Sets the type of waveform to be generated by the Oscillator. The parameter is called **Shape** instead of “waveform”, because it doesn’t necessarily set only one

waveform, but sets a specific oscillator model that could produce a number of waveforms depending on other settings. A good example for this behaviour is the *Pulse* shape. However, the term “waveform” is used interchangeably throughout the manual. The following shapes are currently available:



- *Pulse* selects the pulse waveform. A pulse waveform with a pulse width of 50% has only the odd harmonics of the fundamental frequency present. This waveform produces a hollow / metallic sound. If the Pulse waveform is selected, the parameters **Wave** and **Wave Mod Source** are used to change the pulsewidth of the waveform.
- *Saw* selects the sawtooth waveform. A Sawtooth wave has all the harmonics of the fundamental frequency in descending magnitude. It doesn't have any further parameters.
- *Triangle* selects the triangular waveform. The triangle mainly consists of the odd harmonics with very low magnitudes. It doesn't have any further parameters.

- *Sine* consists of the fundamental frequency only. It has no harmonics at all. It doesn't have any further parameter.
- 68 *Wavetables* based on classic wavetables of Waldorf synthesizers can be found in oscillator 1 and 2 only. An introduction to the Wavetables can be found in the Appendix of this manual. When a wavetable is selected, the parameters **Wave** and **Wave Mod** serve to select the start point of the waves. Furthermore, the modulation sources *O1PW* and *O2PW* are active subject to which Oscillator is set to the wavetable.

- \* The *Pulse* shape can be used to create unusual synthesizer sounds and oboe like sounds.
- \* The *Saw* shape can be used for basses, leads, brass and string sounds.
- \* The *Tri* shape can be used for flute sounds or soft pads.
- \* The *Sine* shape can be used to produce basses and electric organ sounds.
- \* The *Sine* shape is also perfectly suited for ring modulation or as **FM Source**.

- \* The wavetables are perfect for very vivid sounds.
- \* The tips listed here of course are only suggestions. The variety of tones among all Waveforms in the Largo are nearly inexhaustible.

### Octave

*128'...1/2'*

Sets the basic pitch of the oscillator in steps of an octave. The **Octave** setting is shown as register value, a common measurement based on the length of organ pipes. The reference pitch for the oscillator is generated at MIDI note A3 (note no. 69) when **Octave** is set to 8', **Semitone** and **Detune** are set to 0, **Keytrack** is set to 100% and no pitch modulation is applied. In this case, the oscillator's frequency will be the same as set in the global **Master Tune** parameter (usually 440 Hz).

- \* Set **Octave** to 16' for bass sounds.
- \* Set **Octave** to 8' if you are creating typical keyboard sounds.
- \* Set **Octave** to 4' if you are programming strings or other high pitched sounds.

- \* When you use an Oscillator to modulate other Oscillators or the Filters, experiment with the Octave parameter. Low settings might result in periodic modulations while high settings might add a bell character to the timbre.
- \* Ultra-low settings like 128' can create very nice rhythmic changes when used with the ring modulator. If this is still not low enough, you can use the Standard Modulation Matrix to apply *MAXIMUM* to the respective oscillator pitch with a negative amount.

### Semitone

*-12...+12*

Sets the pitch of the oscillator in semitone steps. The standard setting for this parameter is 0, but there are cases where different values are interesting as well.

- \* Organ sounds often include a fifth, therefore one oscillator's semitone parameter must be set to +7.
- \* Lead and Solo sounds might sound interesting when you set one Oscillator to e.g. a fourth (+5 semitones).
- \* When making ring modulated or FM sounds, try to use dissonant values, e.g. +6 or +8.



## Detune

-64...+63

Fine-tunes the oscillator in steps of 1/128th of a semitone. The audible result of detuned oscillators is a Chorus or Flanger effect. Use a positive setting for one oscillator and an equivalent negative setting for another..

- \* A low value of  $\pm 1$  results in a slow and soft Flanger effect.
- \* Mid-ranged settings of  $\pm 5$  are perfect for pads and other fat sounding programs.
- \* High values of  $\pm 12$  or above will give a strong detune that can be used for accordions or effect sounds.

## FM Source

see table „FM Sources“

Selects the source for the frequency modulation of the selected oscillator.

- \* You can create nice E-Piano sounds when you use a high pitched oscillator as **FM Source** and set its **Keytrack** to a value between 000% and +050%.

- \* The use of Noise is very interesting as **FM Source** on a high pitched oscillator playing a sine or triangle waveform. With a low **FM Amount**, the oscillator starts to sound dirty or airy while higher amounts create a coloured noise similar to a filter with high resonance. A side benefit is that the filters are then still free for other purposes.

## FM Amount

0...127

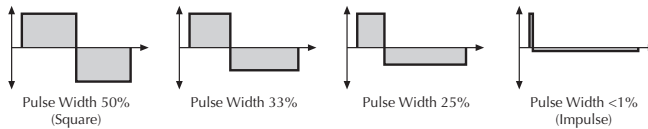
Sets the amount of frequency modulation that is applied to the oscillator by the selected **FM source**. The sound will get more metallic and sometimes even drift out of tune, especially if Oscillator 2 is used as FM Source for Oscillator 3 and **Sync** is activated. To avoid unusable detune, use a triangular or sine waveform for the FM Source.

- \* If you want to bias **FM** over the keyboard so that higher notes aren't modulated as strongly as lower notes, use the **Modulation Matrix** and apply **Keytrack** to the respective oscillator FM with a negative amount.

## Pulsewidth/Wavescan

0...127

Sets the pulse width of the Pulse waveform or changes the startpoint of the wavetable. Click in the waveform display and drag the mouse to the left or right to change the value. The value 0 is equivalent to a pulse ratio of 0%, the oscillator is producing no audible signal. The value 127 is equivalent to 50% and produces a symmetric square wave without even harmonics. If you select any waveform other than *Pulse* or the wavetables, this parameter does not have any effect. The following picture illustrates the effect of the pulsewidth parameter:



- \* A pulse width of 50% can be used for flute sounds or very hollow bass sounds.
- \* A pulse width of around 30% can be used for e-piano or fat bass sounds.
- \* A pulse width of around 10% is interesting for Clavinet sounds.

- \* Experiment with different start points for the Wavetable waveforms to get an overview of the array of available Waveforms for both wavetables.

## Wave Source

see table „Modulation Sources“

Selects the source of the pulsewidth modulation or the wavetable modulation. Common sources for pulsewidth modulation are envelopes and LFOs, but other sources like the modulation wheel or aftertouch can create nice effects as well.

- \* To create a thick oscillator sound, use a triangular *LFO* as **Wave Source** with full **Wave Amount** and a **Pulsewidth** of around 80. This basic setting is useful for very big string and lead sounds. When you play different notes on the keyboard, you might notice that bass notes sound more detuned than higher notes. To avoid this behaviour, set **Keytrack** of the used LFO to a positive setting between 50% and 100%.
- \* To create a Bass sound, use a decaying *Envelope* as **Wave Source** with negative **Wave Amount** and a **Pulsewidth** of around 80...127. This results in a fat attack phase, especially when only one oscillator is used for the Bass.

✳ When you want to create a sound with a wave sweep, you should set the startpoint roughly (**Pulsewidth**) at the desired wave, before you apply any modulation. This will help you to find the basic waveform, before the effect of modulations. Note that you can apply unipolar and bipolar modulation sources. For example, set **Pulsewidth** to 64, which is almost the middle of the wavetable and apply a slow running LFO to **Wave Source** to sweep through the whole wavetable.

**Wave Amount** -64...+63

This parameter determines the amount of modulation that is applied to the pulsewidth of the oscillator's square wave. If a wavetable is selected, **Wave Amount** determines the amount of the wavetable modulation. If you select any waveform other than *Pulse* or the wavetables, this parameter does not have any effect. The modulation source that affects the pulsewidth is selected by the source parameter described above.

**Bend Range** -24...+24

Determines the intensity of the pitchbend via MIDI Pitchbend messages in semitones for the selected oscillator.

**Keytrack** -200%...+196%

Determines how much the pitch of the selected oscillator depends on the MIDI note number. The reference note for Keytrack is E3, note number 64. For positive settings, the oscillator pitch rises on notes above the reference note, for negative settings the oscillator pitch falls by the same amount and vice versa. A setting of +100% corresponds to a 1:1 scale, e.g. when an octave is played on the keyboard the pitch changes for the same amount. Other settings than +100% make sense especially when using ring modulation, FM or oscillator synchronization. Try to use values in the range 0%...+75% or even negative settings for one oscillator while leaving the second at +100% Keytrack.

**Sub Oscillator Octave** 1...4

Available for Oscillator 1 and 2 only. **Sub Oscillator Octave** determines the relative proportion of Sub and Main oscillator frequencies. Higher values represent a lower tuning.

**Level** 0...127

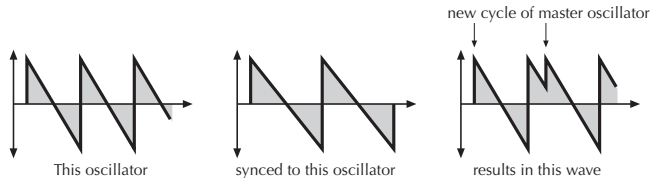
Available for Oscillator 1 and 2 only. **Sub Oscillator Level** determines the volume of the Suboscillator creating a

Square Waveform. A setting of 0 corresponds to no volume. Please note that the loudness of the Suboscillator also depends on the volume of the main oscillator

### Sync (only for Osc 2)

*off / on*

Enables or disables oscillator synchronization. When enabled, Oscillator 2 acts as a slave that is controlled by Oscillator 3, the master. Each time Oscillator 3 starts a new cycle, it sends a trigger signal to Oscillator 2, forcing it to restart its waveform cycle, too. As a result, interesting sound effects may be generated, especially when both oscillators are operating at different pitch settings. Using additional pitch modulation by envelopes, LFO, or Pitch bend will lend further movement to sync sounds. The following picture illustrates the principle of oscillator synchronization in a simplified way:



- \* Use **Sync** for Lead or Solo sounds. Set Oscillator 2 to play one octave and 7 semitones higher, apply an envelope to its pitch with positive amount and you get a screaming sync sound.
- \* **Sync** can also be very interesting on arpeggio sounds. Apply a slow clocked LFO to Oscillator 2 pitch and the arpeggio starts to move.

### Osc Pitch Mod Source *see Table “Modulation Sources”*

Selects the source of the pitch modulation for all oscillators. A common source for pitch modulation is an LFO whose strength is controlled by the modwheel or aftertouch.

- \* To create a common pitch vibrato that is controlled by the modwheel, set **Osc Pitch Mod Source** to  $LFO1 * MW$  with **Osc Pitch Mod Amount** set to around +20.
- \* To create a sound whose pitch glides in, set **Osc Pitch Mod Source** to a decaying *Envelope* with **Osc Pitch Mod Amount** set to around -25.

**i** If you want to modulate the pitch of individual oscillators, you can do so with the **Modulation Matrix**.

**Osc Pitch Mod Amount** -64...+63

Sets the amount of pitch modulation for all oscillators. Positive amounts will raise the pitch when positive modulation is applied, e.g. by pressing aftertouch on the keyboard. Negative amounts will lower the pitch when positive modulation is applied

**Mixer Section**

**Osc Level** 0...127

Controls the volume of the selected Oscillator.

**Osc Balance** F1 64...mid...F2  
63

Determines the ratio of the selected Oscillator's signal that is sent to the inputs of Filter 1 and Filter 2. If set to *F1 64*, the signal is sent to Filter 1 only. Higher values will increase the amount



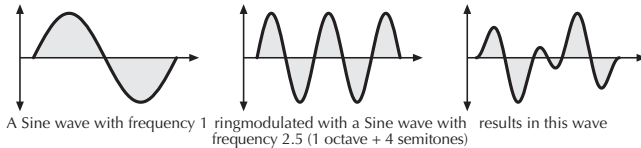
of signal that feeds Filter 2 and decrease the amount of signal that feeds Filter 1. If set to *mid*, both filters will receive the same signal level. If set to *F2 63*, the signal is sent to Filter 2 only.

**Ring Modulation**

The following parameters refer to the ring modulator.

**Ring Mod Level** 0...127

Volume of the ring modulation between Oscillator 1 and 2. From a technical point of view ring modulation is the multiplication of two oscillators' signals. The result of this operation is a waveform that contains the sums and the differences of the source frequency components. Since the ring modulation generates disharmonic components, it can be used to add metallic distorted sound characteristics. This is useful e.g. when generating synth percussion. Please note that in a complex waveform all harmonic components behave like interacting sine waves, resulting in a wide spectral range of the ring modulated sound. The following pictures show the results of two ringmodulated sine waves:



**i** Ring Modulation can result in unwanted low frequencies when the pitches of oscillator 1 and 2 don't differ very much. This is logical because when you use i.e. one oscillator set to 100 Hz and the second set to 101 Hz, the resulting ring modulation is 201 Hz and 1 Hz, and 1 Hz is very low.

- \* Ring Modulation can be very interesting when a slow pitch modulation is applied to one oscillator, i.e. a decaying *Envelope*. This creates spacy effect sounds.
- \* For an E-Piano sound, you might apply Ring Modulation when one high pitched oscillator's **Keytrack** is lowered to i.e. 50%.
- \* If you turn down the pitch of one oscillator markedly, you can get an effect very similar to Amplitude modulation. Use this for sounds with a periodic element if you wish.

## RingMod Balance

*F1 64...mid...F2 63*

Determines the ratio of the ring modulator's signal that is sent to the inputs of Filter 1 and/or Filter 2. If set to *F1 64*, the signal is sent to Filter 1 only. Higher values will increase the amount of signal that feeds Filter 2 and decrease the amount of signal that feeds Filter 1. If set to *mid*, both filters will receive the same signal level. If set to *F2 63*, the signal is sent to Filter 2 only.

## Noise

The following parameters refer to the noise generator.

### Noise Level

*0...127*

Volume of the noise generator Noise is a fundamental source for any kind of analog-type percussion. Also, wind, beaches and other sound effects can be created by using the noise generator.

### Noise Balance

*F1 64...mid...F2 63*

Determines how loud the noise signal is sent to the inputs of Filter 1 and Filter 2. If set to *F1 64*, the signal is sent to Filter 1 only. Higher values will increase the amount of signal that feeds Filter 2 and decrease the amount of

signal that feeds Filter 1. If set to *mid*, both filters will receive the same signal level. If set to *F2 63*, the signal is sent to Filter 2 only.


### Noise Colour -64...+63

Colourizes the noise signal. A value of 0 produces White Noise while positive values lower the bass area. Negative values dampens the higher noise frequencies.

### The Filter Section

The Largo offers two filter that use the same parameters for editing.



 A detailed introduction of the filters can be found in the Appendix of this manual.

### Routing *parallel / serial*

The Routing function is one of the advanced features of the Largo. Its purpose is to control the signal flow of the filters. In comparison with many other synthesizers where signal flow is static, the Largo offers a more flexible configuration.

The Largo offers two independent filters and panning units. In fact, the panning units are part of the filters in the Largo. The routing control makes it possible to change the signal flow from a parallel to a serial filter configuration and vice versa.

This is how the Routing section works in detail:

- The Oscillator section uses two separate outputs – one connected to the input of Filter 1, the other connected to the input of Filter 2. Each sound source, i.e. the oscillators, the ring modulator, and the noise generator has an individual **Balance** control. By means of these controls you can determine what portion of each source's signal is routed to the inputs of Filter 1 and Filter 2. E.g. this makes it possible to send the signal of Oscillator 1

and 2 to Filter 1 and the ring modulation signal to Filter 2.

- When the **Routing** parameter is set to *serial*, the whole output signal of Filter 1 is routed to the input of Filter 2, and added to the amount of signal that is already coming from the Oscillators through its dedicated output. This setting is equivalent to a serial connection of the two filters.
- When the **Routing** parameter is set to *parallel*, the whole output signal of Filter 1 is routed to the filter's panning unit. This setting is equivalent to a parallel routing of two filters, each filtering the input signals independently.

After passing the panning units, both signals are summed up again and sent to the Amplifier and Effect section.

## Cutoff

0...127

Controls the cutoff frequency for the low pass and high pass filter types, the center frequency for the band pass and notch filter types, and the delay length of the comb filter types.

- When a low pass type is selected via the **Type** parameter, all frequencies above the cutoff frequency are damped.

- When a high pass type is selected, all frequencies below the cutoff frequency are damped.
- When a band pass type is selected, only frequencies near the cutoff setting will be passed through.
- When a notch type is selected, the frequencies around the cutoff frequency are damped.
- When a comb type is selected, the frequencies near the cutoff frequency are emphasized (comb+) or attenuated (comb-).

You can bring more movement into the sound by modulating the cutoff frequency via the LFOs, the envelopes or the **Keytrack** parameter of the filter. At a value of 64 and a **Resonance** value of 114, the filter oscillates with 440Hz, which is equal to A3 (the Comb+ type oscillates one octave higher). Tuning is scaled in semitone steps. When **Keytrack** is set to +100%, the filter can be played in a tempered scale.

## Resonance

0...127

Controls the emphasis of the frequencies around the cutoff point. Use lower values in the range of 0...80 to give more brilliance to the sound. At higher values of 80...113 the sound gets the typical filter character with a strong



boost around the cutoff frequency. When the setting is raised to values above *113*, the filter starts to self-oscillate, generating a pure sine wave. This feature can be used to create analog-style effects and percussion-like electronic toms, kicks, zaps etc.

**Drive** *0...127*

Determines the amount of saturation that is added to the signal. If set to *0*, no saturation will be added or, in other words, the signal will remain clean. Lower values will add some harmonics to the signal, resulting in a warm character. Increasing the value will bring in more and more distortion, suitable for harder lead sounds and effects.

**Drive Types** *miscellaneous*

Determines the character of the drive. The following drive curves are available:

- **Light** produces a symmetric light distortion.
- **Tube** simulates the asymmetric distortion of a tube circuit.
- **Medium** produces a symmetric intermediate distortion.

- **Hard** cuts the signal at a certain level.
- **Shaper** modulates the level of the signal with a sine waver. High levels will be turned down while medium levels will be raised up. Same results with negative levels. A sawtooth wave for example changes to a sine wave. This drive type is very effective in combination with amplitude modulation, ring modulation or noise.
- **Fuzz** distorts nearly every input signal to a square wave. Use this drive type for outlandish effects.
- **Pickup 1** simulates an electrostatic pickup. Use amplitude modulations for more absorbing sound effects.
- **Pickup 2** simulates an electromagnetic pickup. As for Pickup 1 use amplitude modulations for more absorbing sound effects. Think about a guitar or an E-piano. The level of a string or a steel reed influences the electrical conversion done by the pickup.

**Filter Type** *miscellaneous*

Selects the filter type.


- *Bypass* deactivates the filter. This is useful if you want to disable the filter temporarily and listen to the oscillators' pure signals.
- *LP 24dB / LP 12dB Low Pass* are suitable for most normal applications. Use the 24dB slope if you want to create sounds with a typical audible filtered character; use the 12dB slope if you want to get softer results.
- *BP 24dB / BP 12dB Band Pass* remove frequencies both below and above the cutoff point. As a result, the sound character gets thinner. Use these filter types when programming effect and percussion-like sounds.
- *HP 24dB / HP 12dB High Pass* are useful to thin out a sound's bass frequencies. This may also give interesting results in conjunction with cutoff frequency modulation. By doing this you can e.g. "fly-in" a sound starting at its high harmonics and then coming up to its full frequency range. Use the 24dB slope if you want to create sounds with a typical audible filtered character; use the 12dB slope if you want to get softer results.
- *Notch 24dB / Notch 12dB* are the opposites of the band pass types. They dampen frequencies around

the cutoff point. Frequencies below or above the cutoff point are passed through. Use these filter types for programming effect sounds. On Notch filter types, the Resonance parameter is almost useless by definition because the resonance frequency is exactly the frequency that is blocked by the filter. However, you will still be able to hear slight differences when you change the Resonance because of phase changes. Anyway, the effect isn't that spectacular.

- *Comb+ / Comb- Filter* differ from the other filter types greatly, because they don't actually damp any part of the signal, but instead add a delayed version of the input signal to the output.



**What exactly is a Comb filter?** A Comb filter is basically a very short delay that can be controlled in length and feedback. The delay time is so short that you can't hear its individual taps but a colorization of the original signal created by peaks or holes in the frequency spectrum. The frequency of the colorization is set by the delay length, which is controlled in the Largo through **Cutoff**, while the amount of colorization is set by the Comb filter feedback, which is controlled in the Largo through **Resonance**.

 To learn more about the comb filters please refer to the corresponding paragraph in the Appendix of this manual.

### **Env Amount** -64...+63

Determines the amount of influence the filter envelope has on the cutoff frequency. For positive settings, the filter cutoff frequency is increased by the modulation of the envelope, for negative settings, the cutoff frequency is decreased. Use this parameter to change the timbre of the sound over time. Sounds with a hard attack usually have a positive envelope amount that makes the start phase bright and then closes the filter to get a darker sustain phase. String sounds, on the other hand, usually use a negative envelope amount that gives a slow and dark attack before the cutoff rises in the sustain phase.


Since there are two filters, you could use the Filter Envelope on one of them, and another envelope on the second filter, but setting this parameter for that filter to zero and use the Modulation Matrix for the other used envelope. Specially with the two filters placed serial this can bring nice effects.

### **Keytrack** -200%...+196%

Determines how much the cutoff frequency depends on the MIDI note number. The reference note for Keytrack is E3, note number 64. For positive settings, the cutoff frequency rises on notes above the reference note, for negative settings the cutoff frequency falls by the same amount, and vice versa. A setting of +100% corresponds to a 1:1 scale, so e.g. when an octave is played on a keyboard the cutoff frequency changes by the same amount. If you want to play the filter in a tempered scale, e.g. for a solo sound with self-oscillation, set the value to +100%. On most bass sounds lower settings in the range +50...+75% are optimal to keep the sound smooth at higher notes.

### **Env Velocity** -64...+63

Determines the amount of influence the filter envelope has on the cutoff frequency, based on key velocity. This parameter works similarly to the **Env Amount** parameter with the difference that its intensity is velocity based. Use this feature to give a more expressive character to the sound. When you hit the keys smoothly, only minimal modulation is applied. When you hit them harder, the modulation amount also gets stronger.

 The overall modulation applied to the filter's cutoff frequency is calculated as the sum of both the **Env Amount** and **Env Velocity** parameters. Therefore you should always bear this total in mind, especially when the filter does not behave as you expect. You can also create interesting effects by setting one parameter to a positive and the other to a negative amount.

**FM Source** *see table „FM Sources“*

Selects the source of the frequency modulation.

**FM Amount** *off, 1...127*

Sets the amount of frequency modulation that is applied to the filter by the selected source.

**Cutoff Mod Source** *see table „ModulationSources“*

Selects the source of the cutoff modulation.

**Cutoff Mod Amount** *-64...+63*

Controls the amount of cutoff modulation. Positive amounts will increase the cutoff frequency when positive modulation is applied, e.g. by pressing the aftertouch on

the keyboard. Negative amounts will decrease the cutoff frequency when positive modulation is applied.

**Level** *0...127*

Controls the volume of the corresponding filter output. This is very useful when unwanted distortions occur.

**Pan** *left 64...center...right 63*

Determines the position in the stereo panorama. When the setting is *left 64*, the sound is panned far left; when the setting is *right 63*, it is panned far right. If you want to situate the sound in the middle of the stereo panorama, use the *center* setting. To give further movement to the sound, set this parameter to a basic value and apply some modulation to it via the **Pan Source** parameter.

**Pan Mod Source** *see table „ModulationSources“*

Selects the source of the panorama modulation.

**Pan Mod Amount** *-64...+63*

Determines the amount of panorama modulation.

## Bass Boost

*On / Off*

This button activates the typical Waldorf bass emphasis which is located between filter and amplifier.

## The Amplifier

### Level

Determines the master volume of the sound program. To understand the operation of this unit, it is important to know that the Amplifier Envelope is always acting as a modulation source for the volume level. This means that an audio signal can only pass through if the Amplifier Envelope is triggered and opened.

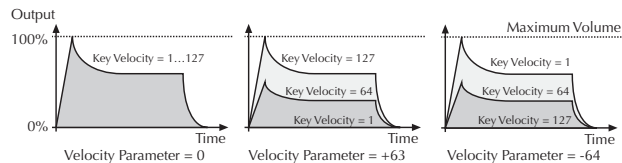


### Velocity

*-64...+63*

Specifies how much volume will be affected by keyboard velocity. Use this feature to give more expression to the sound. With a setting of 0, velocity will have no effect on the volume. Classic organs work in this way because they do not have dynamic response. For positive settings, the volume rises with higher velocities. This is the most

commonly used setting, which gives a piano-like character. With negative settings, the volume decreases at higher velocities. This gives an atypical character suitable for effect sounds. As the Amplifier always works in conjunction with the Amplifier Envelope, this parameter actually determines the envelope velocity amount. The following picture illustrates this functionality:



### Amp Mod Source

*see table Modulation Sources*

Selects the source of the volume modulation.

### Amp Mod Amount

*0...127*

Determines the amount of volume modulation.

## Parameter Module Section

The controls for e.g. envelopes, the modulation matrix and the three LFOs can be found in the **Parameter Module Section**.



### Selection of the Modules

Click on the desired Module to open the corresponding area.

### Common Area



### Sound

The Sound display shows the name and category of the current sound program. Click on the left arrow buttons to step through the sound programs of the current loaded sound bank. Click on the right arrow button to open a pop-down menu for a faster sound selection.

Doubleclick on the soundname or the category to change it by entering a new indication.

### Equalizer

Largo offers a parametric equalizer to adjust the sound frequencies to your needs. The equalizer contains four bands with the following parameters:

- **Gain** raises or lowers the volume of the selected frequency.
- **Freq (Frequency)** sets the frequency at which the tones will be affected.
- **Q** widens or shortens the frequency range of the current band. Low and High band offer additional shelving characteristics with overshots and undershots.

### Control Mode

Determines the mouse control for editing knobs in Largo:

- *Auto* uses the settings of your host application. Please refer to the corresponding manual of your host application.
- *Circular* lets the knob control behave like a real knob. Click on the knob, hold down the left mouse button

and drag the mouse around the dial in a circle to change values.

- *Linear* is an alternative knob control. Click on the knob, hold down the left mouse button and drag the mouse up or down to change the value. The knob reacts like a fader.

**i** We recommend the settings *Auto* or *Linear*.

## Voices

1...256

Largo offers up to 256 voices. The number of available voices depends on the available processor power.

## Outputs

*Stereo / Multi*

Largo offers up to four independent stereo outputs.

- *Stereo* mixes down all four layer signals to one single stereo output.
- *Multi* offers four separate stereo outputs - one output for each layer.

**i** Please keep in mind that some host applications (e.g. Cubase 4 or 5) need to activate additional outputs within the application. Please refer to the corresponding manual chapter of your host application.

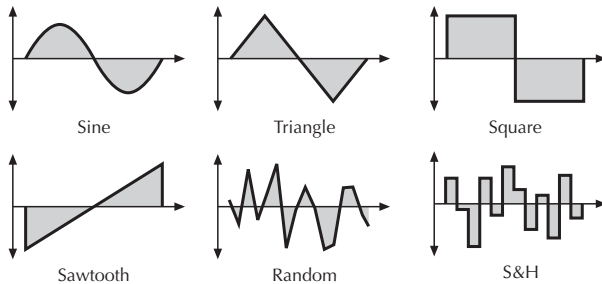
## The LFOs



In addition to the main oscillators, the Largo is equipped with three low frequency oscillators (LFO) that can be used for modulation purposes. Each LFO generates a periodic waveform with adjustable frequency and shape. LFO3 offers free definable steps.

**Shape 1** *Sine/ Triangle/ Square/ Saw/ Random/ S&H / Step*

Sets the type of waveform. The following picture shows the available shapes:



- The *Sine* shape is best suited for oscillator FM or pan modulations.
- The *Triangle* shape is perfect for smooth pitch, filter or volume modulations.
- The *Square* shape can be interesting for hard pan modulations or special effects.
- The *Sawtooth* shape can generate interesting filter or volume changes. If you need a modulation with inverted slope, just apply the Sawtooth shape with negative amount.
- The *Random* shape generates random values and glides to them linearly.
- *S&H* (Sample & Hold) samples a random value and holds it until the next value is generated. If

**Speed** is set to 0, a random value is generated on each new note.

- *Step* is only available for LFO3 and offers a user definable step shape for rhythmical modulation.

### Start Phase

*free, 0°...360*

Controls the initial phase of the LFO when a new note is started. Click in the waveform display of the LFO and drag it to change the start phase of the LFO. *Free* means that the LFO isn't restarted on a new note but runs freely while other values set the LFO phase to the respective offset in degrees.

### Speed

*0...127 or 1280 bars...1/96*

Determines the frequency of the corresponding LFO. At low values, it might take several minutes for the LFO to perform a complete cycle while higher values are in the audible range. Very high values are scaled in semitone steps. With **LFO Keytrack** set to 100%, a **Speed** setting of 122 delivers an 8' LFO oscillation. 16' oscillation can therefore be generated with a **Speed** setting of 110, and so on.

When the LFO **Clock** parameter is set to *On*, you can adjust the **Speed** in musical values. The lowest possible



value is *1280 bars*, meaning that a complete LFO cycle would need 1280 bars.

**Sync** *off, on*

When **Sync** is set to *on*, the LFO phases of all voices are synced so that they sound as one LFO. This can be interesting when the LFO is applied to modulate **Filter Cutoff** or **Pan**.

When **Sync** is set to *off*, the LFOs run independently; this is better suited for pitch modulation to obtain thicker sounds.

**i** **Sync** does not mean that the LFO is synced to song tempo or to note start. This is done with the **Clock** parameter or the **Phase** parameter.

**Clock** *off / on*

When **Clock** is activated, the LFO is synced to the song tempo of the host application. The **LFO Speed** setting changed to offer musically meaningful values.

**Keytrack** *-200%...+196%*

Determines how much the speed of the LFO depends on MIDI note number. The reference note for Keytrack is E3, note number 64. For positive settings, the LFO speeds up

on notes above the reference note, for negative settings the LFO slows down when higher notes are played and vice versa. A setting of *+100%* corresponds to a 1:1 scale, e.g. when the keyboard is played an octave higher, the LFO speed is doubled.

**Delay** *0...127*

The **Delay** parameter works in different ways depending on the setting of the **Fade** parameter:

- When **Fade** is set to *+00...+63*, the LFO signal output is zero for the time set with the **Delay** parameter. After this time, the LFO is faded in and then runs with full magnitude.
- When **Fade** is set to *-64...-01*, the LFO runs with full magnitude for the time set with the **Delay** parameter. After this time, the LFO is faded out to zero.

**Fade** *-64...+63*

Controls the speed with which the LFO is faded in or out. With this parameter you can create slowly rising or falling modulations that might create interest when routed to pitch or volume.



### Contour (LFO 3 only)

0...127

This parameter smooths the envelope curve of the step waveform with falling edges when the Step shape is chosen for LFO3.

### Step Shape (LFO 3 only)

Use your computer mouse to change the amplitude of the LFO steps. While holding the left mouse button you can change more than one step at the time.

### Step Count (LFO 3 only)

1...16


Determines the number of the LFO steps.


## The Envelopes



The Largo envelopes allow you to manipulate sound parameters via rate or timed modulations. The Largo offers four independent programmable envelopes for every sound program:

- A Filter Envelope (Filter Env). This envelope is fixed to control the filter but can also be used for other modulations.
- An Amplifier Envelope (Amp Env). This envelope is fixed to control the sound volume, but can also be used for other modulations.
- Two additional Envelopes, Env 3 and Env 4. These envelopes can be used freely to perform additional modulations on any module.

 The Parameter Controls of all envelopes are nearly similar.

 A detailed introduction of the envelopes can be found in the Appendix of this manual.

## The ADSR Envelope

Click on one of the handles in the envelope graphics to continuously and smoothly change the envelope parameters or click into any envelope phase to let its value jump here.

**Attack** 0...127

Determines the attack rate or amount of time it takes for a signal to go from zero to maximum level.

**Decay** 0...127

Determines the decay rate or amount of time it takes for a signal to reach the **Sustain** level.

**Sustain** 0...127

Determines the sustain level which is held until a note ends.

**Release** 0...127

Once the note has ended, the release phase begins. During this phase, the envelope fades to zero at the rate determined by the Release value.

**Trigger** *Multi / single*

Determines the triggering of the Filter Envelope.

- *Multi* means that every Note starts the filter envelope of its own voice.
- *Single* means that the envelopes of all voices of a selected program behave like a single envelope. This common envelope starts as soon as the first note is played. The sustain level remains until the last note is released. Afterwards the release phase is active.

**Mode** *ADSR / ADS1DS2R / One Shot / Loop S1S2 / Loop All*

Pop-up menu to switch between the various envelope types. Please refer to the Appendix for detailed explanation of all the types.

## The additional Envelope Parameter



### Attack Level 0...127

Controls the level at each the **Attack** phase ends and the **Decay** phase starts. This parameter affects the envelope types *ADS1DS2R*, *One Shot*, *Loop S1S2* and *Loop All* only.

### Decay 2 0...127

Determines the decay rate or amount of time it takes for a signal to reach the **Sustain 2** level. This parameter affects *ADS1DS2R*, *One Shot*, *Loop S1S2* and *Loop All* only.

### Sustain 2 0...127

Sets the second **Sustain** level. As soon as this level is reached, the envelope goes into the **Release** phase. This parameter affects the envelope types *ADS1DS2R*, *One Shot*, *Loop S1S2* and *Loop All* only.

## The Modulation Matrix

A modulation can be described as a signal-generating unit's influence upon a sound parameter. The terms used in this context are "Source" (**Src**) and "Destination" (**Dest**). The Largo offers 16 independent modulation assignments (slots) each with individual settings of source, destination and amount. The Modulation Matrix (Mod Matrix) is the key of the power of each Waldorf synthesizer, so start experimenting with it *right now*.



**i** A complete table of all available sources and destinations can be found in the Appendix.

### Src (Source) *see table Modulation Sources*

This pop-up menu defines the modulation source.

### Amount -64...+63

Determines the amount of modulation applied to the destination. Since the modulation is in fact a multiplication of the source signal and this parameter, the

resulting amplitude depends on the type of modulation source you select:

- For the so-called unipolar modulation sources, the resulting amplitude lies within the range of 0...+1, if Amount is positive or 0...-1, if Amount is negative. These sources are: all envelopes, all MIDI controllers including Modwheel, Foot control etc., Velocity, Release Velocity, Aftertouch (Pressure) and Polyphonic Pressure.
- For the so-called bipolar modulation sources, the resulting amplitude lies within the range of -1...0...+1. These sources are: all LFOs, Keytrack, Pitchbend and the Modifiers.

**Dest (Destination)** *see table Modulation Destinations*

This pop-up menu defines the modulation destination.

### Modifier

Modifiers allow you to apply mathematical functions on modulation signals. Depending on the function type selected, calculation will affect two source



signals or a source signal and a constant parameter. You can use up to four independent modifier units. The result of each operation is not directly audible, but is used as input source for the Modulation Matrix. Moreover, you can use it as source for yet another modifying process.

The parameter for the modifiers can be found after the pages of the Modulation slots. Please select these pages with the **Selection Dial** an.

**Src A (Source A)** *see table Modulation Sources*

Selects the first source signal used for the calculation.

**Src B (Source B)** *Const / see table Modulation Sources*

Selects the second source signal when two sources are required for the calculation. See description of modifier functions for further details. The possible settings are the same as for **Source A** with one exception: "Off" is replaced by "constant" meaning that the calculation is performed with a constant value that you can set up with the **Constant** parameter.

**Op (Operation)***see table*

Determines which kind of operation will be performed on the selected input sources. The following types are available:

Setting	Description
+	Addition
-	Subtraction
*	Multiplication
XOR	Exclusive OR function
OR	OR function
AND	AND function
Min	Minimum value
Max	Maximum value

*Modifier functions*

The result of a modifier operation always lies within the range -max...0...+max. When it is assigned to a parameter in the Modulation Matrix, it is scaled to the range of the selected parameter.

The following paragraph describes the function and the result of each modifier function in detail:

- + Returns the sum of **Source A** and **Source B**.

- - Returns the difference of **Source A** and **Source B**.
- \* Returns the product of **Source A** and **Source B**.
- AND Returns the binary „and“ operation of **Source A** and **Source B**.
- OR Returns the binary „or“ operation of **Source A** and **Source B**.
- XOR Returns the binary „exclusive-or“ operation of **Source A** and **Source B**.
- min Returns the minimum value of either **Source A** or **Source B**. If **Source A** is smaller than **Source B**, the value of **Source A** is returned and vice versa.
- MAX Returns the maximum value of either **Source A** or **Source B**. If **Source A** is greater than **Source B**, the value of **Source A** is returned and vice versa.

**Constant***0...127*

Defines a value for modifier functions that require a constant parameter. See the **Operation** parameter described above for further details.

## The Arpeggiator



An Arpeggiator is a device that splits an incoming MIDI chord into its single notes and repeats them rhythmically. Different sequence modes can be defined for the Arpeggiator to cover a wide range of applications. In addition to the synthesis features, the Largo offers a deeply programmable Arpeggiator for every sound program. It can play a wide range of different rhythm patterns including accents and different timings, and allows creating sophisticated custom rhythm patterns.

The arpeggiator uses a so-called note list that can store up to 16 notes. This list is set up depending on the Arpeggiator parameter settings. Refer to the individual parameter descriptions to see if and how a parameter alters the list.

**Arp Active** off / on

Switches the Arpeggiator on or off.

**Mode** Normal / One Shot / Hold

This parameter sets the way the Arpeggiator works.

- If *Normal* is selected, the arpeggiator is active. When you press a note or a chord on the keyboard, it is split up and repeated rhythmically. As soon as you release a note, it is removed from the arpeggio rhythm. Conversely, as soon as you add another note to the existing chord, it is inserted into the arpeggio. When you release all notes, the arpeggiator stops.
- If *One Shot* is selected, the arpeggiator splits up all played notes and plays back an arpeggio. The actual length of this arpeggio is set by the **Pattern Length** parameter. After the arpeggio rhythm is played once, it is stopped automatically unless you hit a new chord. This mode is especially useful in a live performance where you might have to “synchronize” yourself, for example, to a drummer. Just hit a chord at each new bar.
- If *Hold* is selected, the arpeggiator splits up all played notes and generates a continuous arpeggio even when the chord is released. This gives you two ways of entering a chord:
  - Press all keys of the chord simultaneously. This is the normal procedure you would follow with the other Arpeggiator modes, too.  
or

- Press and hold the first key of the chord. While holding this key, enter the other keys sequentially. After playing all keys, you can release the first key. This method is practical for playing difficult chords. It allows you to create arpeggios in the sequence of played notes. You can even hit the same note several times and it will appear in the note list accordingly.



You can stop playback of the Arpeggiator by sending an All Notes Off message from your sequencer.

### Clock

*1/64t... 1280 bars*

Sets the note value for the steps of the rhythm pattern in a range from whole notes to thirty-second triplet notes. Triplets (e.g. 1/8T) and dotted notes (e.g. 1/16.) are available for every note value.

### Length

*1/64t... 1024 bars / legato*

Sets the length of the generated arpeggio notes. However, when **Length** is set to *Legato*, all arpeggio notes are played without pauses between each step and **Arpeggiator Length** therefore has no effect.

### Sort Order

*as played / reversed / Key Lo>Hi / Key Hi>Lo / Vel Lo>Hi / Vel Hi>Lo*

Sets the order in which the note list is arranged. With this parameter, you can determine how the notes you originally play are split up for the arpeggio.

- If *as played* is selected, the notes are sorted in the exact order you played them. If you e.g. press E1, G1 and C1, the note list looks exactly like that.
- If *reversed* is selected, the notes are sorted to the reverse order in which you played them. If you e.g. press E1, G1 and C1, the note list is sorted to C1, G1 and E1.
- If *Key Lo>Hi* is selected, the notes are sorted from the lowest note to the highest note. If you e.g. press E1, G1 and C1, the note list is sorted to C1, E1 and G1.
- *Key Hi>Lo* is the opposite of *Key Lo>Hi*. The example would be sorted as G1, E1 and C1.
- If *Vel Lo>Hi* is selected, the notes are sorted from the softest to the loudest velocity. If you press notes with velocities 64, 120 and 96, the note list will be sorted to 64, 96 and 120.



- *Vel Hi>Lo* is the opposite of *Vel Lo>Hi*. The above notes would be sorted as 120, 96 and 64.

**Vel Mode** *Each Note / First Note / Last Note*

Determines how velocity is interpreted in the arpeggio. Note that each arpeggio step might have an additional positive or negative offset set by the **Arp Accent** parameter.

- If *Each Note* is selected, each note of the arpeggio is played back with the velocity that you originally played.
- If *First Note* is selected, the first note you played sets the velocity for all arpeggio steps.
- If *Last Note* is selected, the last note you played sets the velocity for all arpeggio steps.

**Direction** *Up / Down / Alt Up / Alt Down*

Sets the direction that is used to play back the arpeggio.

- If *Up* is selected, the note list is played forward and the octaves are transposed upward. The arpeggio starts in the original octave and goes up to the highest octave. Then the arpeggio is repeated.

- If *Down* is selected, the note list is played backward and the octaves are transposed downward. The arpeggio starts in the highest octave and goes down to the original octave. Then the arpeggio is repeated.
- If *Alt Up* is selected, the note list is first played forward and the octaves are transposed upward. After reaching the last note of the note list in the highest octave to play, the note list is played backward and the octaves are transposed downward down to the first note of the note list in the original octave. Then the arpeggio is repeated.
- If *Alt Down* is selected, the note list is first played backward and the octaves are transposed downward. The arpeggio starts in the highest octave. After reaching the first note of the note list in the original octave, the note list is played forward and the octaves are transposed upward up to the last note of the note list in the highest octave to play. Then the arpeggio is repeated.

**Octave Range** *1...10*

Determines the range of the single notes in octaves. When it is set to *1 Oct*, the note list will be played back in the same octave as originally entered. Greater values mean

that the note list is repeated in higher or lower octaves. The octave in which the arpeggio starts is determined by the **Direction** parameter. If you play notes that span more than one octave, they are still kept in the note list and played back before the note list is transposed. The following table shows some possible arpeggios:

Note input	Range	Direction	Resulting Arpeggio
C1 E1 G1	1 Oct	Up	C1 E1 G1   C1 E1 G1
C1 E1 G1	2 Oct	Up	C1 E1 G1   C2 E2 G2   C1 E1 G1   C2
E1 G1 C1	3 Oct	Up	E1 G1 C1   E2 G2 C2   E3 G3 C3   E1
C1 G1 E2	3 Oct	Up	C1 G1 E2   C2 G2 E3   C3 G3 E4   C1
C1 E1 G1	3 Oct	Down	G3 E3 C3   G2 E2 C2   G1 E1 C1   G3
C1 E1 G1	2 Oct	Alt Down	G2 E2 C2   G1 E1   C1 E1 G1   C2 E2

*Arpeggio results depending on note input*

**Max Notes**

0...16

Sets the maximum number of notes that are used for the note list. When you press more notes than set up here, the

oldest notes are removed from the list. When you set it to 1 on a Bass sound, for example, the Arpeggiator only stores the last entered note and plays back sequentially. All other notes might still be used by other sounds, but the Bass sound plays back only this one particular note.

**Pattern Reset**

*off / on*

When all steps of an arpeggio pattern are played back, the pattern is repeated from the beginning so that the arpeggio is looped. With **Pattern Reset**, you can decide if the note list is also restarted from the beginning when the rhythm pattern is reset.

- If *off* is selected, the note list is not restarted, so that there is no synchronization between rhythm and note list. E.g., when you have a pattern where four steps are set and you play three notes, the pattern and the note list are repeated differently. The pattern restarts after the fourth step while the note list restarts after the third step. The arpeggio might look like this:

Pattern Step	1	2	3	4	1	2	3	4
Note	C1	E1	G1	C1	E1	G1	C1	E1

<b>Pat. Step</b>	1	2	3	4	1	2	3	4
<b>Note</b>	G1	C1	E1	G1	C1	E1	G1	C1

*Arpeggio with Pattern Reset set to „on“*

- If *on* is selected, the note list will be restarted as soon as the rhythm pattern is restarted. The same arpeggio might now look like this (note the two C1s in sequence):

<b>Pattern Step</b>	1	2	3	4	1	2	3	4
<b>Note</b>	C1	E1	G1	C1	C1	E1	G1	C1

<b>Pat. Step</b>	1	2	3	4	1	2	3	4
<b>Note</b>	C1	E1	G1	C1	C1	E1	G1	C1

*Arpeggio with Pattern Reset set to „Off“*

**Pattern** *off / User / 1...15*

Sets the rhythm pattern that is used for generating the arpeggio. **Pattern** can set to *off*, *User* or to one of the 15 ROM patterns.

- If *off* is selected, the arpeggiator plays back a continuous sequence of notes with the current **Clock** setting.

- *User* gives you the ability to create your own custom rhythm pattern. This pattern is stored in in the sound. See the section “Arpeggiator Step Editor” below about the pattern settings you can create.
- *1...15* selects one of the 15 internal ROM rhythm patterns. See the table below for an overview of each ROM rhythm pattern:

Pattern	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	●		●	●	●		●	●	●		●	●	●		●	●
2	●		●		●			●		●		●		●		●
3	●		●		●		●		●		●		●		●	
4	●		●	●	●		●		●		●	●	●		●	
5	●		●		●	●		●		●		●		●	●	●
6		●		●		●		●		●		●		●		●
7	●		●		●		●		●		●		●		●	
8	●		●		●		●		●		●		●		●	
9	●	●	●		●	●	●		●	●	●		●	●	●	
10	●	●		●	●		●	●		●	●		●	●	●	
11	●	●		●	●		●	●		●	●		●	●	●	
12	●	●		●	●		●	●		●	●		●	●	●	
13	●		●		●		●		●		●		●	●	●	●
14	●		●		●		●		●		●		●	●	●	●
15	●		●		●		●		●		●		●	●	●	●

## Timing Factor

0...127

Determines how much the **Arp Timing** parameter affects an arpeggio step. If **Timing Factor** is set to 0, the settings in **Arp Timing** are completely ignored and the arpeggio is played back without any shuffled timing. Settings from 1 to 127 increase the shuffling of the notes depending on the setting in the **Arp Timing** parameter. **Timing Factor** also works on ROM patterns; these are set up with standard swing rhythm.

## Same Note Overlap

on / off

Due to the various **Length** parameters, it might happen that two notes having equal pitch overlap, e.g. when you play only one single note on a 16 beat rhythm with lengths set to 1/8.

Note that this parameter only affects notes of the same pitches. If you want to control the lengths of notes of different pitches, use the **Length** or **Arpeggiator Length** parameters.

- If *on* is selected, the Note Off message of the previous note is allowed to be sent after a Note On message of the next note with the same pitch. This setting can be nice with arpeggios triggering sounds with slow attack.

- If *off* is selected, the Note Off message of the previous note is sent right before the next note with the same pitch is started.

## Arpeggiator Step Editor

The following Arpeggiator parameters can be edited with your computer mouse easily.



You can only select the steps that are in the range set by **Pattern Length**. E.g., when you have set **Pattern Length** to 8, you can't change a value of step 9 or above.

## Pattern Length

1...16

Sets the length of the rhythm pattern. This parameter is also valid when **Pattern** is set to *Off* or if one of the ROM patterns is selected. Note that you can only edit a particular step in the **Step Editor** when its position is within the range of the **Pattern Length** parameter.

## Step Editing

Click and hold the left mouse button on the desired step to edit it. Depending on whether you click on the step, you can change different parameter:

- Click in the left area of a step and move the mouse horizontally to change the timing of the corresponding step.
- Click in the right area of a step and move the mouse horizontally to change the step length.
- Click outside of the described areas (see above) and move the mouse vertically to change the accent of the corresponding step.

### Arp Step

*on / off*

Click on the corresponding button 1 to 16 to switch an Arp Step on or off.

### Accent

*miscellaneous step data*

Sets the accent of a particular step. This accent is interpreted as a velocity offset that is added or subtracted from the original velocity stored in the note list. However, the generated velocities can never exceed the maximum MIDI velocity 127 or minimum MIDI velocity 1. This means when you have already played notes with a high velocity, **Accent** might not be able to further offset them positively, so you will only hear differences with negative accent offsets. Conversely, low velocities might not be able to be lowered by **Accent**. The only exception is *silent*. See the following description:

- If *silent* is selected, the current step is actually played, but inaudible. This means that the note list is advanced by one note, but you are not able to hear it. This feature is the opposite to **Step** set to *o pause*, which doesn't generate any note and therefore doesn't advance the note list.

### Timing

*random / -3...+3*

Moves the playback time of a step forward or backward. Forward means that a step is played later while backward means that it is played earlier. The overall strength of this parameter is set by **Timing Factor**. If **Timing Factor** is set to 0, **Arp Timing** has no effect on the rhythm at all. If **Timing Factor** is set to maximum, **Arp Timing** can move the step by a maximum of half the clock division. This means that it can move the step by 1/32 forward or backward when clock is set to 1/16.

- Negative values (-3, -2 oder -1) move the step backward so that it is played earlier.
- If 0 is selected, the step isn't moved at all.
- Positive values (+1, +2, +3) move the step forward so that it is played later.

## Length

*legato / -3...+3*

Changes the length of the note of a particular step. The overall length of the arpeggio depends on the **Length** setting. If **Length** is set to *legato*, **Arp Length** doesn't have any effect at all. Also, if **Length** is set to a very small value, **Arp Length** might not have an audible effect when you set it to a negative value. You can create very nice staccato and legato effects with this parameter.

- If *legato* is selected, the notes of this step are held until the next step is played. Empty steps force any step notes set to legato to remain held.
- Negative values (-3, -2 oder -1) shorten the length of the notes of this step.
- If 0 is selected, the step is held for the time set in the **Length** parameter.
- Positive values (+1, +2, +3) extend the note duration of this step.

## The Effects



The Largo has two effect units which are editable separately for each layer.

## FX Active

Switches the corresponding FX unit on or off.

## Mix

*0...127*


The only parameter that is common to all types of effects is the **Mix** parameter. This parameter controls the volume ratio between the original signal and the effect output. If set to 0, the dry signal is sent to the outputs only so that no effects can be heard. Higher values will increase the effect signal. At maximum setting, the pure effect signal will be heard.

**Type** *Bypass / Chorus / Flanger / Phaser / Overdrive / Delay / Reverb*

Sets the type of effect produced by the first effect unit. All further parameters depend on the selected effect type.

## Chorus

A Chorus effect is generated by using Comb filters that generate slightly detuned copies of the input signal and mix it into the output signal. The result sounds like an ensemble of several simultaneous sounds, like a choir as opposed to a single voice; hence the name Chorus. The detuning is generated by an internal LFO that can be controlled in speed and depth. The Chorus' high frequency output can be dampened with the Cutoff parameter.

 A Mix setting of 48 to 96 produces the strongest effect because both the unaffected signal and the processed signal are mixed together.

**Speed** *0...127*

Sets the LFO speed of the Chorus effect.

**Depth** *0...127*


Sets the modulation depth of the Chorus effect.

**Delay** *0...127*

Sets the initial delay that is added to the Chorus delay line.

## Flanger

The Flanger effect is very similar to the Chorus effect, but features feedback circuitry to feed the generated signal back into the comb filter. This generates a deeper detuning and colorizes the signal. With extreme settings you can hear a whistling sound which is very characteristic of a Flanger effect.

 A Mix setting of 48 to 96 produces the strongest effect because both the unaffected signal and the processed signal are mixed together.

**Speed** *0...127*

Sets the LFO speed of the Flanger effect.

**Depth** *0...127*

Sets the modulation depth of the Flanger effect.

**Delay** *0...127*

Sets the initial delay that is added to the Flanger delay line.

**Feedback** *0...127*

Controls the feedback amount of the Flanger signal.

**Polarity** *positive / negative*

Determines whether the feedback signal is fed back into the Flanger as is, or is inverted beforehand.

**Phaser**

A Phaser is a combination of several "allpass" filters working in parallel. This generates an effect with equally spaced frequency peaks or troughs. The result is a strongly colored signal with a "spacy" character.



A Mix setting of 48 to 96 produces the strongest effect because both the unaffected signal and the processed signal are mixed together.

**Speed** *0...127*

Sets the LFO speed of the Phaser effect.

**Depth** *0...127*

Sets the modulation depth of the Phaser effect.

**Center** *0...127*

Controls the basic delay length of the allpass filters. Lower settings produce a high pitched Phaser effect while higher settings enable the Phaser to cover deeper frequencies.

**Spacing** *0...127*

Controls the relative frequencies of the allpass filters. A setting of 0 produces a classic phaser, while higher settings spread out the frequencies of the allpass filters.

**Feedback** *0...127*

Controls the feedback amount of the delay signal.



**Polarity** *positive / negative*

Determines whether the feedback signal is fed back into the Phaser as is, or is inverted beforehand.

**Overdrive**

The Overdrive effect distorts the input signal by amplifying it drastically and clipping the resulting signal to a certain output level. The difference between this Overdrive effect and the Drive parameter of the Filter sections is that Drive affects one single voice, while this effect type distorts the summed output of the whole instrument. Therefore, the resulting effect is different as soon as you play more than one note and you should consider which is best suited to a particular sound program. The Overdrive effect, for example, works great on Organ or E-Piano sounds.

**Drive** *0...127*

Controls the amount of distortion the effect produces. Low values create no or only slight distortion while high values create heavier distortion.

**Post Gain** *0...127*

Controls the output level of the distorted signal.

**Cutoff** *0...127*

Dampens the high frequency output of the Overdrive effect.

**i** Please note that the setting of the **Mix** parameter doesn't affect the strength of the overdrive effect but only the volume of it. Therefore, you can get a very strong overdrive with a low volume when you turn up **Drive** and turn down **Mix**.

**Delay**

**i** This effect type is only available for the **Effect 2** unit.

A Delay is an effect that produces echoes of the input signal.

**Clock** *on / off*

Synchronizes the Delay to the song tempo of your host application.

**Delay** *0...127 or 1/512t...12 bars*

Sets the length of the Delay tap up to three seconds.

**Feedback***0...127*

Controls the amount of signal that is routed back into the Delay line. Lower values therefore produce fewer echoes than higher values.

**Polarity***positive / negative*

Determines whether the feedback signal is fed back into the delay line as is, or is inverted beforehand.

**Cutoff***0...127*

Dampens the signal produced by the Delay effect. This filter is routed before the feedback circuitry meaning that adjacent taps of the Delay will be dampened further. This creates the typical “high frequency loss” that often happens in natural echoes. A setting of 127 means that the signal isn’t filtered, while lower settings filter the high frequencies of the feedback signal.

**Ping Pong***-64...+63*

Spreads the left and right delay output to half of the delay time maximum. Settings from -64 or +63 create a typical ping pong delay.

**Reverb**

**i** This effect type is only available for the **Effect 2** unit.

The Reverb effect is probably the most widely used effect in music production. It is used to add a realistic ambience to clean and dry audio recorded in a studio. Very complicated mathematical algorithms are needed to simulate the complexity of a natural reverb. As a result, good reverb processors are very expensive. The Largo’s reverb effects don’t intend to simulate the perfect natural room, rather they are an addition to the Largo’s sound synthesis to make it more 3 dimensional and expressive.

**Size***0...127*

Determines the length of the longest room side in meters. Lower settings simulate a normal room while higher settings simulate a big hall or church.

**Shape***0...127*

Changes the characteristic of the reverb. In most effect units you can select different rooms and reverb types like plate or echo chamber. The **Shape** parameter enables stepless changes of these reverb types. Lower settings

simulate a more natural reverb while higher settings simulate a plate reverb.

**Decay** *0...127*

Determines the length of the reverb reflections. To simulate a big room choose higher Decay settings, to simulate a smaller room choose lower settings.

**Predelay** *0...127*

Determines the delay in milliseconds between the direct sound and the reverb effect output. A natural reverb contains a reflexion which is heard later than the original signal. The predelay determines this delay so that the reverb is separated from the original sound. Lower settings connect the reverb more to the original signal while higher settings separate the effect signal to produce a more spacious sound.

**HP (High Pass)** *0...127*

Determines the frequency where the lower frequency parts are divided from the effect signal. This parameter works in the same way as the highpass filter in the Largo's sound synthesis. A natural reverb doesn't produce all

frequencies equally so the highpass cuts the lower frequencies to produce a more realistic reverb effect.

**LP (Low Pass)** *0...127*

Determines the frequency where the higher frequency parts are divided from the effect signal. This parameter works in the same way as the lowpass filter in the Largo's sound synthesis. A natural reverb doesn't produce all frequencies equally so the lowpass cuts the higher frequencies to produce a more realistic reverb effect.

**Diffusion** *0...127*

Determines the consistency of different room materials. Lower settings produce a cold strong reflection like a metal or tile wall. Higher settings make a warm and thick reverb as if the signal were reflected by a uneven surface. Settings above *100* change the room characteristic even further.

**Damping** *0...127*

Determines how fast the room simulation dampens the higher reverb frequencies. Higher frequencies are not dampened not as much as lower frequencies. This phenomenon makes a natural echo sound somewhat

hollow. Higher values of the Damping parameter mean more high frequencies will be dampened.

## Sound Parameter in the Keyboard Section



### Glide *off / on*

Enables or disables the Glide effect. "Glide" or "Portamento" describes the continuous gliding from one note to another. This effect can be created on fretless stringed instruments or some brass instruments (e.g. trombone). It is very common on synthesizers and used throughout all music styles. Please note that Glide affects the pitch of all oscillators.

### Time *0...127*

Determines the glide time. Low values will give a short glide time in a range of milliseconds that gives a special character to the sound. High values will result in a long glide time of up to several seconds which can be useful for solo and effect sounds.

### Glide Steps *on / off*

Activates quantizing on the glide steps to allow a glissando effect.

### Glide Mode *Normal / Legato*

Determines the way the Glide effect works.


- *Normal* means that a continuous glide is performed on all new notes.
- *Legato* means that a continuous glide is performed only when notes are played legato. Staccato played notes start on the exact pitch of their note.

### Unisono Voices *Off / Dual / 3...6*

Controls how many voices are triggered when a note is played.


- *Off* means that a note triggers one voice. This is the standard mode.
- *Dual* means that a note triggers two voices. Both voices have high priority so they can cut off other voices that are played.
- *3...6* means that this number of voices is triggered when a note is played. Only the first voice has high

priority, meaning that it can cut off other played notes. The other voices can only sound if any voices are free or if there are other unison voices with lower priority that could be cut off. This ensures that older notes play at least one voice as long as the voice allocation isn't forced to steal even this voice for a new note.

 **Please note that the chosen number of unison voices will respectively lower the number of available voices.**


### Unisono Detune 0...127

Controls the detuning of the Unison voices. Each voice is detuned differently; with **Unisono Detune**, you control the overall amount.

 **Unisono Detune** is perfectly suited to thickening the tone. Arpeggios benefit too from the detune function.

### Unisono Spread 0...127


Spreads all unisono voices automatically within the stereo panorama. Higher values will alternate the voices between a left and right panorama position.

 **Unisono Spread** is perfect to add more width to your sound.

### Mono *On / Off*

Controls if the Sound can be played polyphonically or monophonically.

- *Off* means that each note triggers its own voice or voices, as on a piano.
- *On* means that only the last played note sounds. All other notes are stored in an internal list but aren't played. As soon as you release the note that is currently played, the second last note is played and so on. When you play legato, only the first note that was played triggers the envelopes. All later notes use these envelopes, but sound in the pitch you've played. This mode is for sustained sounds like typical 70's solo sounds.

 When *Mono* is selected and you have set up a decaying volume envelope for the selected Sound, you might not hear anything after playing several notes because of the envelopes decaying to 0.

# Appendix

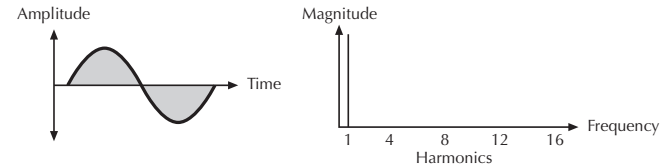
## Oscillators Introduction

The oscillator is the first building block of a synthesizer. It delivers the signal that is transformed by all other components of the synthesizer. In the early days of electronic synthesis, Robert A. Moog found out that most real acoustic instrument waveforms could be reproduced by using abstracted electronic versions of these waveforms. He wasn't the first who came to that conclusion, but he was the first in recreating them electronically and building them into a machine that could be used commercially. What he implemented into his synthesizer were the still well-known waveforms sawtooth, square and triangle. For sure, this is only a minimal selection of the endless variety of waveforms, but the Largo gives you exactly these waveforms at hand, plus other classic waveforms like pulse (which is the father of the square waveform) and the sine wave (also part of every other waveform).

Now, you probably know how these waveforms look and sound, but the following chapter gives you a small introduction into the deeper structure of these waveforms. Let's start with the most basic one.

## The Sine Wave

The Sine Wave is the purest tone that can be generated. It consists only of one harmonic, the fundamental, and has no overtones. The following picture shows the sine wave and its frequency representation:



*The Sine Wave*

There is no acoustic music instrument that generates a pure sine wave; the only instrument that comes close to it is the pitch fork. Therefore, the sine wave sounds a little artificial to the ear. However, the sine wave can be an interesting oscillator waveform to emphasize a certain harmonic while other oscillators are playing more complex waveforms, or as FM source for frequency modulation.

The sine wave is the most basic building block of each waveform. Any waveform can be broken down to several or many sine waves that are arranged with different frequencies and magnitudes. These sine waves are called

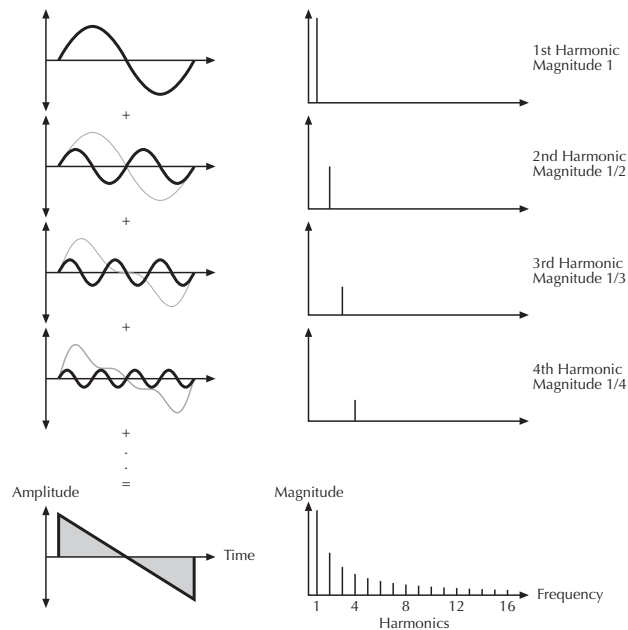
*partials*. In most waveforms, the partial with the lowest frequency is dominant, meaning that this partial is used by the ear to determine the pitch of the tone. This partial is called *fundamental*. All other partials are called *overtones*. So, the second partial is the first overtone.

Cyclic waveforms such as the waveforms in the Largo only feature sine waves in integer frequency ratios to the fundamental like double frequency, triple frequency and so on. Those partials are called *harmonics* because their frequency is a harmonic multiple of the fundamental.

Confusing? Let's generalize it to cyclic waveforms: a cyclic waveform like sawtooth, square etc. only consists of harmonic partials. The harmonic with the lowest frequency is dominant and therefore called fundamental. All other harmonics are called overtones.

### The Sawtooth Wave

The Sawtooth Wave is the most popular synthesizer waveform. It consists of all harmonics in which the magnitude of each harmonic descends by the factor of its position. This means that the first harmonic (the fundamental) has full magnitude, the second harmonic has half magnitude, the third harmonic has a third magnitude and so on. The following picture shows how the individual harmonics build up the sawtooth wave:



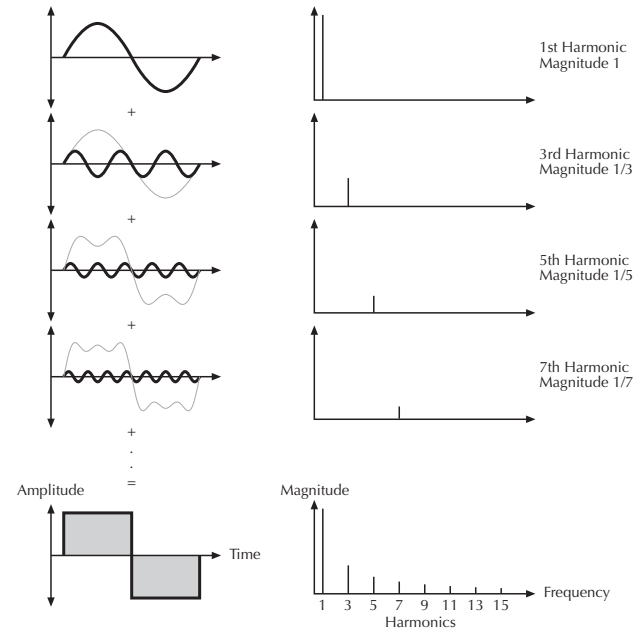
*Additive components of the Sawtooth wave*

The sawtooth wave was thought as an abstraction of the timbre of string and brass instruments. You can easily understand that when you think of a violin. Imagine a

bow pulling the string slightly into one direction. At one point, the string abruptly comes off the bow and swings back to its original position. The bow is still moved and so it catches the string again and the procedure is repeated. The result is a waveform that looks like a sawtooth. The same is true for a brass instrument. The string in this case are the lips while the bow is the air. The lips are moved by the air to a certain extent and abruptly move back to their original position.

### The Square Wave

The Square Wave is a special waveform generated by a pulse waveform with 50% pulse width. This means that the positive part of the waveform has equal length to the negative part. The pulse waveform can have other pulse widths as you will read later. For now, we'll talk about the square wave as a unique waveform. The square wave consists of all odd harmonics in which the magnitude of each harmonic descends by the factor of its position. This means that the first harmonic has full magnitude, the third harmonic has a third magnitude, the fifth harmonic has a fifth magnitude and so on. The following picture shows how the individual harmonics build up the sawtooth wave:



*Additive components of the square wave with 50% pulse width*

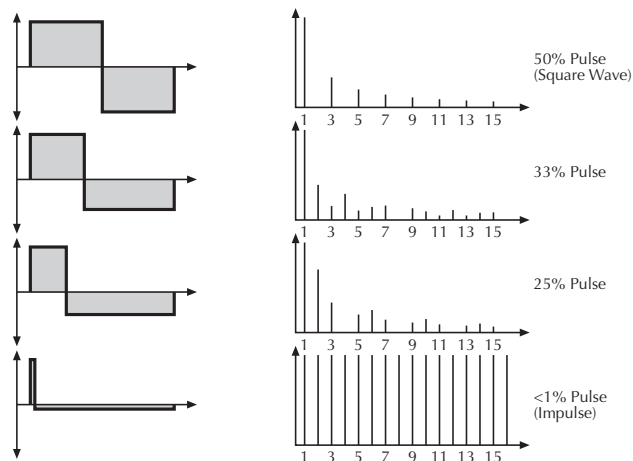
The square wave was thought as an abstraction of wind instruments like a panflute or a recorder. They consist of a



tube that can contain a certain amount of air. The player blows air through the tube in a way such that the air starts to vibrate. This vibration is almost completely symmetrical, resulting in a hollow timbre.

### The Pulse Wave

The Pulse Wave is the most versatile wave in a classic synthesizer because its shape and therefore its harmonic content can be changed in real time. This is done by changing the width of the upper and lower portion of the waveform cycle. These portions are called pulses, hence the name pulse width. The width of the first pulse is used to distinguish between different pulse waves and it is measured in percent. The following picture shows several pulse waves with different pulse widths:



*Additive components of Pulse wave with different pulse widths*

The first thing you probably observe is that the lower part of the wave has a narrower excursion. This is because the energy of the wider pulse is higher than the one of the narrower pulse. If this were not compensated, the overall signal would have an unwanted DC offset.

As you have read in the previous chapter, the harmonic content of a 50% pulse wave is a special case. It has a

---

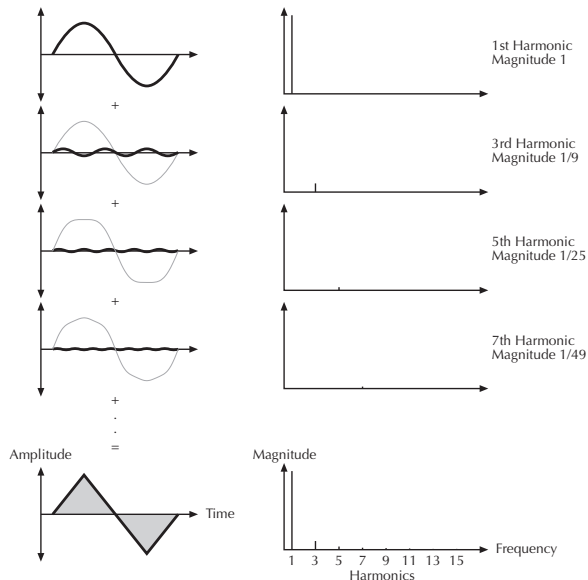
very symmetrical harmonic content, while all other pulse widths create peaks or troughs at certain frequencies. Another special case is a pulse wave with a very narrow pulse width, in the above picture labelled as <1%. An infinitely thin pulse creates a spectrum that has all harmonics with equal magnitudes. In a digital synthesizer, “infinitely” necessarily means one sample.

The pulse wave is an artificial wave, which means that it doesn't occur in nature. It was built into synthesizers because it could create a lot of different timbres with a minimum of technical effort. However, certain pulse widths sound very close to the timbres of acoustic (or semi-acoustic) instruments, i.e. a guitar or bass guitar, an e-piano or even a flute.

The most powerful feature of the pulse wave is the ability to change its width while sounding. This is called pulse width modulation. When the pulse width is changed, the waveform starts to sound thicker. This happens because the effect is very similar to what you hear when you have two oscillators running with slightly different frequencies. They interfere and create irregular troughs in the resulting waveforms.

## The Triangle Wave

The Triangle Wave is very similar to the square wave. It is composed of the same harmonics as the square wave, but with different magnitude ratios. The magnitude of each harmonic is divided by the power of its number. This means that the third harmonic's magnitude is a ninth, the fifth harmonic is a twenty-fifth and so on. The following illustration shows the harmonic content:



*Additive components of the Triangle wave*

The reason why the triangle wave is so popular in classic synthesizers: It could act as a suboscillator wave, to emphasize certain frequencies or to frequency modulate other oscillators.

The triangle wave sounds like a woodwind instrument, i.e. a clarinet. It can also be used for mallet instruments like vibraphone, xylophone etc.

### Wavetable Synthesis in the Largo

The sound generation of the wavetables in the Largo is based on wavetable synthesis. Although wavetable synthesis is not a form of "sample playback", you should avoid this term because functionality, operation and results are totally different. Each of the wavetables contain up to single 128 waves.

**⚠ Please keep the terms "wavetable" and "wave" in mind. Don't confuse the two.**

The following overview explains how the Wavetable Synthesis works: A wavetable into the Largo is a table consisting of up to 128 waveforms. Each waveform is classified by its own special sound character. The main difference of wavetable synthesis in comparison with other sound generation principles is the facility not only to play one waveform per oscillator, but also to walk through the wavetable via different modulations. Therefore you can create wavetable sweeps. The results can be dramatic - much more so than those any sample playback based system could ever produce. This is a unique feature of wavetable synthesis.

This principle offers powerful capabilities. To give some examples:

- Each note on a keyboard can access a different wave of a wavetable.
- Different waves can be played depending on key velocity.
- A LFO can modulate the position within the wavetable. You can create subtle to drastic sound changes.
- User-selected controllers e.g. the mod wheel, can change the position within the wavetable. When you turn the wheel while playing a chord, each note's wave will be modified instantly.
- Wavetables are perfect as FM sources when you want to create very strange sounds.

You should keep the following sentence in mind:

**A wavetable is a list with up to 128 waves, among which you can move at will.**

**i** If you like the sound and possibilities of wavetable synthesis, you should try out the virtual instrument Waldorf PPG Wave 2.V which is part of the Waldorf Edition plug-in bundle.

## List of the Largo Wavetables

No.	Wavetable	No.	Wavetable
1	Alt 1	35	SawSync1
2	Alt 2	36	SawSync2
3	Resonant	37	SawSync3
4	Resonant2	38	PulSync1
5	MalletSyn	39	PulSync2
6	Sqr-Sweep	40	PulSync3
7	Bellish	41	SinSync1
8	Pul-Sweep	42	SinSync2
9	Saw-Sweep	43	SinSync3
10	MellowSaw	44	PWM Pulse
11	Feedback	45	PWM Saw
12	Add Harm	46	Fuzz Wave
13	Reso 3 HP	47	Distorted
14	Wind Syn	48	HeavyFuzz
15	HighHarm	49	Fuzz Sync
16	Clipper	50	K+Strong1
17	OrganSyn	51	K+Strong2
18	SquareSaw	52	K+Strong3
19	Formant1	53	1-2-3-4-5

20	Polated	54	19/twenty
21	Transient	55	Wavetrip1
22	ElectricP	56	Wavetrip2
23	Robotic	57	Wavetrip3
24	StrongHrm	58	Wavetrip4
25	PercOrgan	59	MaleVoice
26	ClipSweep	60	Low Piano
27	ResoHarms	61	ResoSweep
28	2 Echoes	62	Xmas Bell
29	Formant2	63	FM Piano
30	FmntVocal	64	Fat Organ
31	MicroSync	65	Vibes
32	MicroPWM	66	Chorus 2
33	Glassy	67	True PWM
34	SquareHP	68	UpperWaves

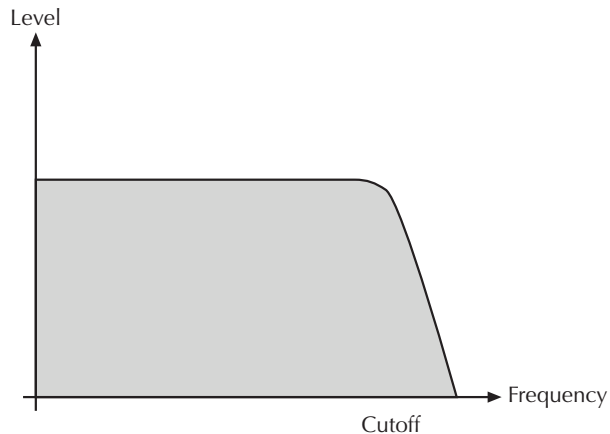
## Filter Introduction

Once the audio signal leaves the mixer, it is sent to the filters. The Largo has two independent filter units, each with its own individual settings. The signal flow in the filters can be controlled via the Routing function. The

filters are components that have significant influence on the Largo’s sound characteristics.

For a detailed description of the different filter types that are available in the Largo, see the section “Filter Types”. For now, we’ll explain the basic function of a filter discussing the type used most commonly in synthesizers: the low pass filter.

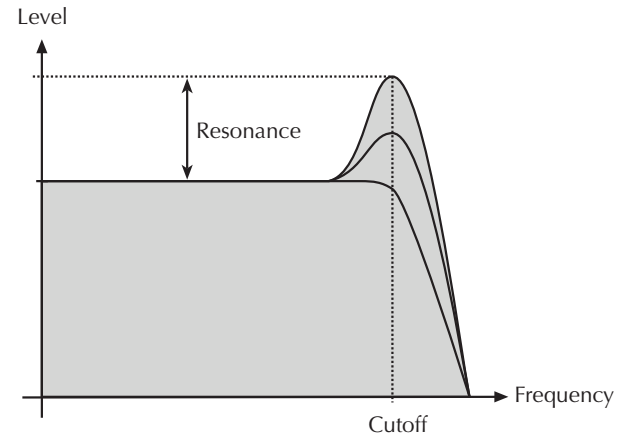
The low pass filter type dampens frequencies that lie above a specified cutoff frequency. Frequencies below this threshold are hardly affected. The frequency below the cutoff point is called the pass band range, the frequencies above are called the stop band range. The Largo’s filter dampens frequencies in the stop band with a certain slope. The slope can be 12dB or 24dB per octave. This means that the level of a frequency that lies an octave above the cutoff point will be 12dB or 24dB less than those frequencies of the signal that fall into the pass band. The following picture shows the basic principle of a low pass filter:



To give you an idea of the extent of damping, consider this example of a low pass filter: A reduction of 24dB reduces the original level one octave above the cutoff point by approx. 94%. The damping factor two octaves above the cutoff point reduces the original level by more than 99%, which in most cases means this portion of the signal is no longer audible.

The Largo's filter also features a resonance parameter. Resonance in the context of a low, band or high pass filter means that a narrow frequency band around the cutoff

point is emphasized. The following picture shows the effect of the resonance parameter on the filter's frequency curve:

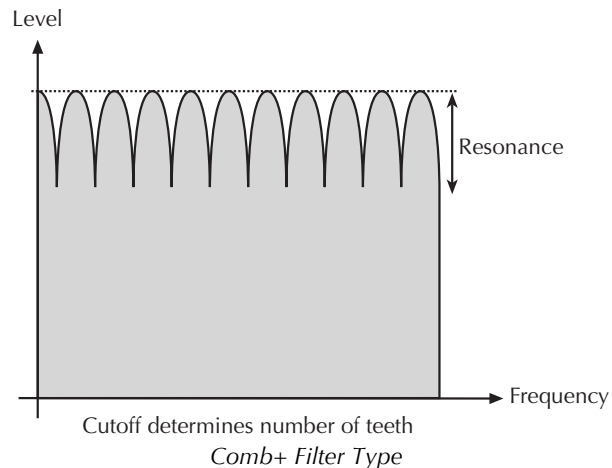


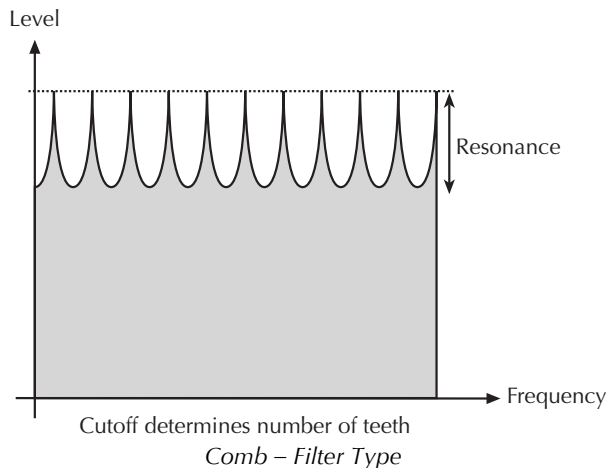
If the resonance is raised to a great extent, then the filter will begin self-oscillation, i.e. the filter generates an audible sine wave even when it does not receive an incoming signal.

## The Comb Filter Types

The Comb filter types differ from the other filter types greatly, because they don't actually damp any part of the signal, but instead add a delayed version of the input signal to the output. What exactly is a Comb filter?

A Comb filter is basically a very short delay that can be controlled in length and feedback. The delay time is so short that you can't hear its individual taps but a colorization of the original signal created by peaks or holes in the frequency spectrum. The frequency of the colorization is set by the delay length, which is controlled in the Largo through **Cutoff**, while the amount of colorization is set by the Comb filter feedback, which is controlled in the Largo through **Resonance**.





So, as with all other filter types, the Comb filter is controlled with two parameters:

- **Cutoff** controls the delay length.
- **Resonance** controls the feedback depth.

Now, what sound can a comb filter produce? Here are some examples:

## The Comb filter as a Chorus

This is the most basic way to use the Comb filter. A Chorus in fact is made of one or several Comb filters, so this is an easy task to set up manually.

### 👉 To set up the Comb filter to produce a Chorus:

1. Select the Comb+ filter type for the desired filter.
2. Set **Cutoff** to a medium value.
3. Set **Resonance**, **Drive**, Filter **Keytrack** and any other modulation to 0.
4. Select an *LFO* as **Mod Source**.
5. Set **Mod** to a medium positive or negative amount.
6. Adjust the **LFO** to generate a slow triangle oscillation.
7. Fine tune **Cutoff**, **Mod** and **LFO Speed** until you get the desired Chorus effect.

\* You can also use both filters to create a Chorus and pan them to different stereo positions.



✳ You can also pan the original, i.e. low pass filtered signal, to one stereo side and route 50% into Filter 2, producing a Chorus effect that is panned to the other stereo side.

You might ask why you should use a Comb filter type to produce Chorus instead of using the FX section of the Largo. Here are some reasons:

- The FX sections remain free for other purposes. Also, in Multi Mode, the Comb filter works no matter on which Instrument you use it.
- The Comb filter works *per voice* rather than per Instrument. This means that each voice has its own Chorus and therefore could generate a different Chorus effect. You can easily hear this when you change **LFO Keytrack** or **Filter Keytrack**. Now, each voice sounds with a different Chorus speed or frequency.
- You have full control over MIDI of all parameters that make up the Chorus effect, by using Control messages.

## The Comb filter as a Flanger

A Flanger is almost the same as a Chorus. The only difference is that it has feedback circuitry to change the depth of the Flanger effect.

### To set up the Comb filter to produce a Flanger:

1. Adjust the filter as described in the Chorus paragraph above.
2. Turn up **Resonance** to create the typical Flanger sound.
3. Gradually lower **LFO Speed** to create slow Flanger sweeps.
4. Select either *Comb+* or *Comb-* to change the character of the Flanger.

✳ Set up **Mod Source** to *Modwheel* to change the Flanger effect manually.

✳ Use the **Filter Envelope** with positive or negative amount to create a “one time sweep” that is triggered at Note On.

✳ Use **Filter Keytrack** with an amount of 100% to create really different flangings depending on the Note you play.

## The Comb filter used to simulate a string or a tube

The Comb filter is one of the most basic building blocks of “Physical Modeling”, which is the term for an algorithmic description of an acoustic instrument. The Largo is definitely not meant to replace a good physical modeling synthesizer, so you shouldn’t expect to get a perfect emulation of an acoustic instrument like a trumpet or a violin out of the Largo. However, the Comb filter enables you to create abstractions of those sounds that can be played expressively.

To do so, you have to understand that the Comb filter is the part that actually sets the pitch of the resulting sound; the oscillators or the noise generator are only used to trigger the Comb filter somehow. When you play with the Comb filter, you can hear that it emphasizes or attenuates certain frequencies depending on the setting of Cutoff and Resonance.

When you set **Resonance** to a very high value, you will notice that the Comb filter oscillates heavily, and that’s the key to using it as a tone generator. When you set **Filter Keytrack** to +100%, you can play the Comb filter musically.

Now you need to know how to set up Cutoff. Filter Cutoff is scaled in semitone steps, and you only have to find the Cutoff value where the filter oscillates with standard pitch. The following table gives you these settings:

Oscillator	Comb+	Comb-
64’	11	23
32’	23	35
16’	35	47
8’	47	59
4’	59	71
2’	71	83

*Comb Filter Cutoff settings*

If you need other tunings, just keep in mind that **Cutoff** is changed in semitones, so if you increment **Cutoff** by 12, the filter oscillates one octave higher, and so forth.



### To set up the Comb filter to produce a tone:

1. Select either *Comb+* to produce a string-like tone or *Comb-* to produce a tube-like tone.
2. Turn up **Resonance** to around 114 to 127.

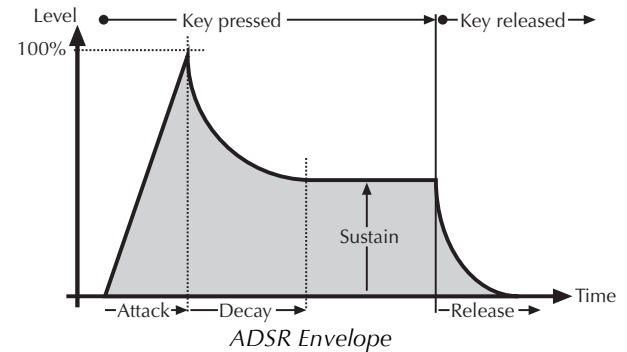
3. Turn up Filter **Keytrack** to +100% to have Filter Cutoff change with equal-tempered scaling.
4. Set **Cutoff** to 23, 35, 47 or 59 to have it oscillating in the same pitch as the oscillator's usually tuning. Please note that *Comb+* oscillates one octave higher than *Comb-*.
5. Feed the Comb filter with a signal of any kind. The incoming signal's pitch isn't important for the resulting tuning, but both pitch and timbre of the input signal affect the timbre of the Comb filter output sound greatly.

- \* Try to experiment with all kinds of input signals, i.e. different oscillator settings, noise, combinations of both and the like.
- \* Also try to set up envelopes to control the levels of the input signal, i.e., use a short percussive signal that only triggers the comb filter briefly. The filter then oscillates to produce the resulting tone.

## Envelope Types in the Largo

### ADSR Envelope

Most traditional synthesizers feature ADSR envelopes. These envelopes are made up of four parameters that determine their response: **Attack**, **Decay**, **Sustain** and **Release**. The other parameters of the Envelope section have no function, so they can not be edited. The following picture illustrates the structure of an ADSR envelope:

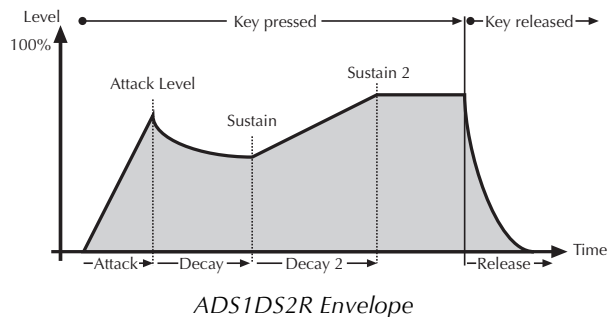


The envelope is started by pressing a key. It ascends to its maximum value at the rate determined by the **Attack** parameter. It then descends at the rate determined by the **Decay** value until it reaches the predetermined **Sustain** value. It remains at this value until the key is released. The envelope then descends to zero at the rate determined by the **Release** parameter.

### ADS1DS2R Envelope

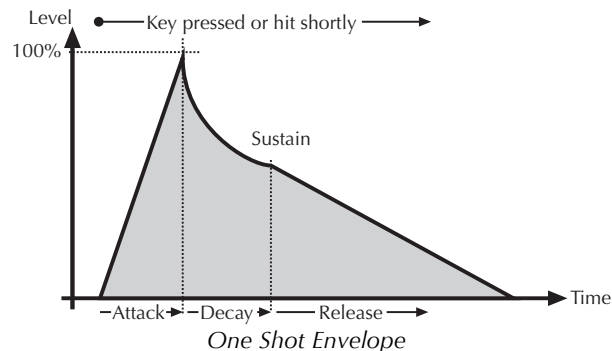
A difficult name for an envelope that is quite easy to understand. Besides the parameters an ADSR envelope features, it offers an adjustable attack level and a second

Decay and Sustain pair. With these additional parameters, you can create much more complex envelopes.



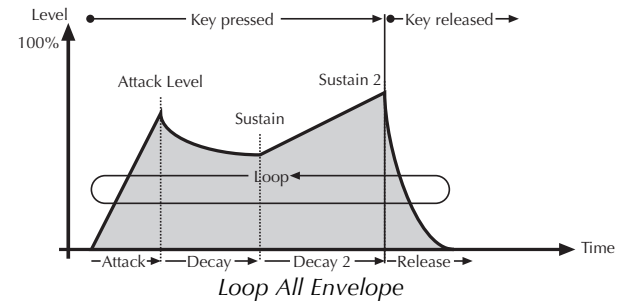
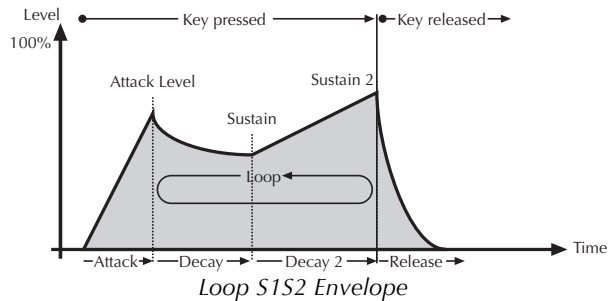
### One Shot Envelope

The One Shot envelope type is designed for percussive sounds that don't need a stage maintained as long as a note is played. In other words: the envelope goes through all its stages, no matter how long a key is pressed. This includes even the Attack phase. It uses the parameter set of the ADSR envelope type in which the **Sustain** parameter is used to set a level breakpoint. This allows the creation of One Shot envelopes with a very percussive attack or with a "Gate" effect. Some parameters of the Envelope section have no function so they can not be edited.



### Loop S1S2 Envelope

The Loop S1S2 envelope type loops the envelope between **Sustain 1** and **Sustain 2** as long as a note is held, after has been through the **Attack** phase once. This means that when **Sustain 2** is reached, **Decay 1** is used to reach **Sustain 1** again, then **Decay 2** is used to reach **Sustain 2**, and so on. As soon as the note is released, the envelope proceeds with the **Release** phase. It uses the parameter set of the ADS1DS2R envelope.



### Loop All Envelope

The Loop All envelope type is similar to the Loop S1S2 envelope type but it loops through all envelope stages as long as a note is held. This means that it goes through all envelope stages first, and if it ended with the **Release** phase, the envelope restarts from zero and goes through all its phases again. As soon as the note is released, the looping stops and the envelope goes into its **Release** phase.

## FM Sources

FM Source:	Description:
off	No modulation
Osc1	Oscillator 1 Signal
Osc2	Oscillator 2 Signal
Osc3	Oscillator 3 Signal
Noise	Noise Generator Signal
LFO1	LFO 1 Signal
LFO2	LFO 2 Signal
LFO3	LFO 3 Signal
FilterEnv	Filter Envelope Signal
AmpEnv	Amplifier Envelope Signal
Env3	Envelope 3 Signal
Env4	Envelope 4 Signal

## Modulation Sources

Modulation Source:	Description
off	No modulation
LFO1	LFO 1 Signal
LFO1*ModWheel	LFO 1 Signal shaped by Modwheel
LFO2	LFO 2 Signal
LFO2*Pressure	LFO 2 Signal shaped by Aftertouch

LFO3	LFO 3 Signal
FilterEnv	Filter Envelope Signal
AmpEnv	Amplifier Envelope Signal
Env3	Envelope 3 Signal
Env4	Envelope 4 Signal
Keytrack	MIDI note number
Velocity	MIDI Velocity
Rel Vel	MIDI note release velocity
Pressure	MIDI channel pressure
Poly Pressure	MIDI polyphonic pressure
Pitchbend	MIDI pitchbend signal
Modwheel	MIDI modulation wheel (contr. #1)
Sustain Ctrl	MIDI sustain pedal (controller #64)
Foot Ctrl	MIDI foot control (controller #4)
Breath Ctrl	MIDI breath control (controller #2)
Control W, X, Y, Z	assignable MIDI controller W to Z
Ctrl Delay	Modifier Delay
Modifier 1...4	Result of Modifier 1...4
Minimum	constant for minimum modulation (equals 0)
Maximum	constant for maximum modulation (equals +1)

Voice Num	Current voice number as Mod Source
Voice %16, %8, %4, %2	Used voice number as Mod Source
Unisono V	Number of unisono voices
Uni Detune	Detuning of all unisono voices
Uni Spread	De-panning of all unisono voices
Uni De-Oct	De-octaving of all unisono voices

## Modulation Destinations

<b>Modulation Destination:</b>	<b>Description:</b>
Pitch	Global pitch of all three Oscillators at once
O1 Pitch, O2 Pitch, O3 Pitch	Pitch of Oscillator 1...3
O1 PW, O2 PW, O3 PW	Pulse width of Oscillator 1...3
O1 FM, O2 FM, O3 FM	FM Amount modulation of Oscillator 1...3
O1 Level, O2 Level, O3 Level	Level of Oscillator 1...3
O1 Balance, O2 Balance, O3 Balance	Filter 1/2 Balance of Oscillator 1...3

Ring Level	Mixer ring modulation level
Ring Balance	Filter 1/2 Balance of Oscillator 1
Noise Level	Mixer level of Noise
Noise Balance	Filter 1/2 Balance of Noise
F1 Cutoff, F2 Cutoff	Cutoff of Filter 1...2
F1 Reson., F2 Reson.	Resonance of Filter 1...2
F1 FM, F2 FM	FM Amount modulation of Filter
F1 Drive, F2 Drive	Drive of Filter 1...2
F1 Pan, F2 Pan	Pan of Filter 1...2
Volume	Amplifier output volume
LFO1-, LFO2-, LFO3-Speed	Speed of LFO 1...3
FE Attack	Attack rate of Filter Envelope
FE Decay	Decay (and Decay 2) rate of Filter Envelope
FE Sustain	Sustain (and Sustain 2) level of Filter Envelope
FE Release	Release rate of Filter Envelope
AE Attack	Attack rate of Amp Envelope
AE Decay	Decay (and Decay 2) rate of Amp Envelope
AE Sustain	Sustain (and Sustain 2) level of Amp Envelope



AE Release	Release rate of Amp Envelope
E3 Attack	Attack rate of Envelope 3
E3 Decay	Decay (and Decay 2) rate of Envelope 3
E3 Sustain	Sustain (and Sustain 2) level of Envelope 3
E3 Release	Release rate of Envelope 3
E4 Attack	Attack rate of Envelope 4
E4 Decay	Decay (and Decay 2) rate of Envelope 4
E4 Sustain	Sustain (and Sustain 2) level of Envelope 4
E4 Release	Release rate of Envelope 4
M1 Amount	Modulation amount of M1
M2 Amount	Modulation amount of M2
M3 Amount	Modulation amount of M1
M4 Amount	Modulation amount of M2
O1 SubDiv, O2 SubDiv	Frequency divide of Sub oscillators
O1 SubVol, O2 SubVol	Volume of Sub oscillators

## Tips & Tricks

- Noise as FM Source for an oscillator gives a colored noise output whose character can be changed with FM Amount, the oscillator waveform and the pitch of the oscillator.
- Use Envelope 3 or 4 to blend in one of the oscillators. E.g. if you need a small blip in the attack phase, just use a free oscillator and a short percussive envelope to create this effect. You can do that in the Modulation Matrix by setting the source to Env3 or Env4, the destination to Osc1, Osc2 or Osc3 level and setting up a positive amount. If this blip should be noisy, use FM with Noise as source for this oscillator to control the strength of the noise.
- For a classic “Filter Trigger” sound, you don’t need to actually use a filter. You can also use a sine oscillator that is pitch modulated by an envelope. This frees up the two filters and you can use them to perform other tasks, e.g. filtering noise for a snare drum sound or the like.
- Use the LFOs as FM Sources. With this technique, you get up to six oscillators frequency-modulating each other to create extremely complex timbres.

Just keep in mind that the LFOs have a maximum speed of around 2500 Hz and that they can produce aliasing side effects.

- Use the wavetables as FM sources. If a LFO modulates the wavetable at the same time, you will create *really* amazing sounds.
- For extremely wide sounds, you can do the following: Create a sound that uses only Filter 1 with the desired settings. Now set Routing to *serial* so that both filters are routed to the panning stages with equal volume. Set Filter 2 to a comb filter type with middle or low cutoff and no resonance. Now set the Pan parameters of the filters to opposite directions and you should get a very fat sound. You can create further movement by applying an LFO to the comb filter cutoff.

## Controller Numbers

Controllers marked with \* don't change the Sound program but can perform temporary modulations or other tasks.

Ctrl #	Controller Range	Controller name or Sound Parameter	Value Range
0	- not used -	Bank Select MSB	- not used
1	0...127	Modulation Wheel*	0...127
2	0...127	Breath Control*	0...127
3	- not used -	<i>undefined</i>	- not used -
4	0...127	Foot Control*	0...127
5	0...127	Glide Time	0...127
6	- not used --	Data Entry	- not used -
7	0...127	Layer Volume*	0...127
8	- not used --	Balance	- not used -
9	- not used -	<i>undefined</i>	- not used -
10	0...127	Pan*	L64...R63
11	- not used -	Expression	- not used -
12	0...9	Arp Octave Range	1...10 octaves
13	0...15	Arp Length	1...16 steps
14	0...3	Arp Active	off, on, one shot, hold
15	0...5	LFO 1 Shape	sine, triangle, square,

			sawtooth, random, S&H
16	0...127	LFO 1 Speed	0...127 or 256 bars...1/96
17	0...1	LFO 1 Sync	off, on
18	0...127	LFO 1 Delay	0...127
19	0...5	LFO 2 Shape	sine, triangle, square, sawtooth, random, S&H
20	0...127	LFO 2 Speed	0...127 or 256 bars...1/96
21	0...1	LFO 2 Sync	off, on
22	0...127	LFO 2 Delay	0...127
23	0...6	LFO 3 Shape	sine, triangle, square, sawtooth, random, S&H
24	0...127	LFO 3 Speed	0...127 or 256 bars...1/96
25	0...1	LFO 3 Sync	off, on
26	0...127	LFO 3 Delay	0...127
27	16, 28, 40...112	Osc 1 Octave	128'...1/2'
28	52...76	Osc 1 Semitone	-12...+12

29	0...127	Osc 1 Detune	-64...+63
30	0...127	Osc 1 FM	0...127
31	0...66	Osc 1 Shape	pulse, saw, triangle, sine, wavetables
32	0...n	Bank Select LSB*	Bank A...
33	0...127	Osc 1 PW	0...127
34	0...127	Osc 1 PWM	-64...+63
35	16, 28, 40...112	Osc 2 Octave	128'...1/2'
36	52...76	Osc 2 Semitone	-12...+12
37	0...127	Osc 2 Detune	-64...+63
38	0...127	Osc 2 FM	0...127
39	0...66	Osc 2 Shape	pulse, saw, triangle, sine, wavetables
40	0...127	Osc 2 PW	0...127
41	0...127	Osc 2 PWM	-64...+63
42	16, 28, 40...112	Osc 3 Octave	128'...1/2'
43	52...76	Osc 3 Semitone	-12...+12
44	0...127	Osc 3 Detune	-64...+63
45	0...127	Osc 3 FM	0...127
46	0...5	Osc 3 Shape	pulse, saw, triangle, sine

47	0...127	Osc 3 PW	0...127
48	0...127	Osc 3 PWM	-64...+63
49	0...1	Sync	off, on
50	0...127	Pitchmod	-64...+63
51	0...9	Glide Mode	siehe "Glide Mode"
52	0...127	Osc 1 Level	0...127
53	0...127	Osc 1 Balance	F1 64...mid...F2 63
54	0...127	Ringmod Level	0...127
55	0...127	Ringmod Balance	F1 64...mid...F2 63
56	0...127	Osc 2 Level	0...127
57	0...127	Osc 2 Balance	F1 64...mid...F2 63
58	0...127	Osc 3 Level	0...127
59	0...127	Osc 3 Balance	F1 64...mid...F2 63
60	0...127	Noise Level	0...127

61	0...127	Noise Balance	F1 64...mid...F2 63
62	0...127	Noise Colour	0...127
63	0...127	Filter 1 Level	0...127
64	0...127	Sustain Pedal*	off, on
65	0...127	Glide Active	off, on
66	0...127	Sostenuto	off, on
67	0...127	Routing	serial/parallel
68	0...10	Filter 1 Type	See Filter types
69	0...127	Filter 1 Cutoff	0...127
70	0...127	Filter 1 Resonance	0...127
71	0...127	Filter 1 Drive	0...127
72	0...127	Filter 1 Keytrack	- 200%... +197%
73	0...127	Filter 1 Env. Amount	-64...+63
74	0...127	Filter 1 Env. Velocity	-64...+63
75	0...127	Filter 1 CutoffMod	-64...+63
76	0...127	Filter 1 FM	off, 1...127

77	0...127	Filter 1 Pan	L64...center... R63
78	0...127	Filter 1 Panmod	-64...+63
79	0...10	Filter 2 Type	See Filter types
80	0...127	Filter 2 Cutoff	0...127
81	0...127	Filter 2 Resonance	0...127
82	0...127	Filter 2 Drive	0...127
83	0...127	Filter 2 Keytrack	- 200%... +197%
84	0...127	Filter 2 Env. Amount	-64...+63
85	0...127	Filter 2 Env. Velocity	-64...+63
86	0...127	Filter 2 CutoffMod	-64...+63
87	0...127	Filter 2 FM	off, 1...127
88	0...127	Filter 2 Pan	L64...center... R63
89	0...127	Filter 2 Panmod	-64...+63
90	0...127	Filter 2 Level	0...127
91	0...127	Amp Velocity	-64...+63
92	0...127	Amp Mod Amt.	-64...+63

93	0...127	FX 1 Mix	0...127
94	0...127	FX 2 Mix	0...127
95	0...127	FE Attack	0...127
96	0...127	FE Decay	0...127
97	0...127	FE Sustain	0...127
98	0...127	FE Decay 2	0...127
99	0...127	FE Sustain 2	0...127
100	0...127	FE Release	0...127
101	0...127	AE Attack	0...127
102	0...127	AE Decay	0...127
103	0...127	AE Sustain	0...127
104	0...127	AE Decay 2	0...127
105	0...127	AE Sustain 2	0...127
106	0...127	AE Release	0...127
107	0...127	E3 Attack	0...127
108	0...127	E3 Decay	0...127
109	0...127	E3 Sustain	0...127
110	0...127	E3 Decay 2	0...127
111	0...127	E3 Sustain 2	0...127
112	0...127	E3 Release	0...127
113	0...127	E4 Attack	0...127
114	0...127	E4 Decay	0...127

115	0...127	E4 Sustain	0...127
116	0...127	E4 Decay 2	0...127
117	0...127	E4 Sustain 2	0...127
118	0...127	E4 Release	0...127
119	- not used -	<i>undefined</i>	- not used -
120	0	All Sound Off*	immediate silence
121	0	Reset All Controllers*	resets all controllers
122	- not used -	<i>undefined</i>	- not used -
123	0	All Notes Off*	releases all voices
124	- not used -	Omni Mode Off	- not used -
125	- not used -	Omni Mode On	- not used -
126	- not used -	Poly Mode On/Off	- not used -
127	- not used -	Poly Mode On	- not used -

---

## Glossary

### Aftertouch

The majority of contemporary keyboards are capable of generating aftertouch messages. On this type of keyboard, when you press harder on a key you are already holding down, a MIDI Aftertouch message is generated. This feature makes sounds even more expressive (e.g. through vibrato).

### Aliasing

Aliasing is an audible side effect arising in digital systems as soon as a signal contains harmonics higher than half the sampling frequency.

### Amount

Describes to which extent a modulation influences a given parameter.

### Amplifier

An amplifier is a component that influences the volume level of a sound via a control signal. This control signal is often generated by an envelope or an LFO.

### Arpeggiator

An arpeggiator is a device that splits an incoming chord into its single notes and repeats them rhythmically. Most arpeggiators feature different sequence modes to cover a wide range of applications. Typical controls for an arpeggiator are the octave range, the direction, the speed and the clock, which means the repetition interval. Some arpeggiators also feature preset or programmable rhythm patterns.

### Attack

An envelope parameter. "Attack" is a term that describes the ascent rate of an envelope from its starting point to the point where it reaches its highest value. The Attack phase is initiated immediately after a trigger signal is received, i.e. after you play a note on the keyboard.

### Band Pass Filter

A band pass filter allows only those frequencies around the cutoff frequency to pass. Frequencies both below and above the cutoff point are damped.

### Band Stop Filter

A band stop filter does the opposite of a band pass filter, i.e. it dampens only the frequencies around the cutoff point and lets all other frequencies pass through.



## Clipping

Clipping is a sort of distortion that occurs when a signal exceeds its maximum value. The curve of a clipped signal is dependent of the system where the clipping takes place. In the analog domain, clipping effectively limits the signal to its maximum level. In the digital domain, clipping is similar to a numerical overflow and so the polarity of the signal's part above the maximum level is negated.

## Control Change (Controllers)

MIDI messages enable you to manipulate the response of a sound generator to a significant degree.

This message essentially consists of two components:

- The Controller number, which defines the element to be influenced. It can be between 0 and 120.
- The Controller value, which determines the extent of the modification.

Controllers can be used for effects such as slowly swelling vibrato, changing the stereo panorama position and influencing filter frequency.

## Decay

"Decay" describes the descent rate of an envelope once the Attack phase has reached its zenith and the envelope drops to the level defined for the Sustain value.

## Envelope

An envelope is used to modulate a sound-shaping component within a given time frame so that the sound is changed in some manner. For instance, an envelope that modulates the cutoff frequency of a filter opens and closes this filter so that some of the signal's frequencies are filtered out. An envelope is started via a trigger, usually a fixed trigger. Normally, the trigger is a MIDI Note. The classic envelope consists of four individually variable phases: Attack, Decay, Sustain and Release. This sequence is called an ADSR envelope. Attack, Decay and Release are time or slope values, and Sustain is a variable volume level. Once an incoming trigger is received, the envelope runs through the Attack and Decay phases until it reaches the programmed Sustain level. This level remains constant until the trigger is terminated. The envelope then initiates the Release phase until it reaches the minimum value.

## Filter

A filter is a component that allows some of a signal's frequencies to pass through it and dampens other frequencies. The most important aspect of a filter is the filter cutoff frequency. Filters generally come in four categories: low pass, high pass, band pass, and band stop. A low pass filter dampens all frequencies above the cutoff

frequency. A high pass filter in turn dampens the frequencies below the cutoff. The band pass filter allows only those frequencies around the cutoff frequency to pass, all others are dampened. A band stop filter does just the opposite, i.e. it dampens only the frequencies around the cutoff frequency. The most common type is the low pass filter.

### **Filter Cutoff Frequency**

The filter cutoff frequency is a significant factor for filters. A low pass filter dampens the portion of the signal that lies above this frequency. Frequencies below this value are allowed to pass through without being processed.

### **High Pass Filter**

A high pass filter dampens all frequencies below its cutoff frequency. Frequencies above the cutoff point are not affected.

### **LFO**

LFO is an acronym for low-frequency oscillator. The LFO generates a periodic oscillation at a low frequency and features variable waveshapes. Similar to an envelope, an LFO can be used to modulate a sound-shaping component.

### **Low Pass Filter**

Synthesizers are often equipped with a low pass filter. A low pass filter dampens all frequencies above its cutoff frequency. Frequencies below the cutoff point are not affected.

### **MIDI**

The acronym MIDI stands for "musical instrument digital interface." It was developed in the early '80s so that diverse types of electronic musical instruments by different manufacturers could interact. At the time a communications standard for heterogeneous devices did not exist, so MIDI was a significant advance. It made it possible to link all devices with one another through simple, uniform connections.

Essentially, this is how MIDI works: One sender is connected to one or several receivers. For instance, if you want to use a computer to play the Pulse, then the computer is the sender and the Pulse acts as the receiver. With a few exceptions, the majority of MIDI devices are equipped with two or three ports for this purpose: MIDI In, MIDI Out and in some cases MIDI Thru. The sender transfers data to the receiver via the MIDI Out jack. Data are sent via a cable to the receiver's MIDI In jack.

MIDI Thru has a special function. It allows the sender to transmit to several receivers. It routes the incoming signal to the next device without modifying it. Another device is simply connected to this jack, thus creating a chain through which the sender can address a number of receivers. Of course it is desirable for the sender to be able to address each device individually. Consequently, there is a rule which is applied to ensure each device responds accordingly.

### **MIDI Channel**

This is a very important element of most messages. A receiver can only respond to incoming messages if its receive channel is set to the same channel as the one the sender is using to transmit data. Subsequently, the sender can address specific receivers individually. MIDI Channels 1 through 16 are available for this purpose.

### **MIDI Clock**

The MIDI Clock message determines the tempo of a piece of music. It serves to synchronize processes based on time.

### **Modulation**

A modulation influences or changes a sound-shaping component via a modulation source. Modulation sources

include envelopes, LFOs or MIDI messages. The modulation destination is sound-shaping component such as a filter or a VCA.

### **Note on / Note off**

This is the most important MIDI message. It determines the pitch and velocity of every generated note. The time of arrival is simultaneously the start time of the note. Its pitch is derived from the note number, which lies between 0 and 127. The velocity lies between 1 and 127. A value of 0 for velocity is similar to "Note Off".

### **Panning**

The process of changing the signal's position within the stereo panorama.

### **Pitchbend**

Pitchbend is a MIDI message. Although pitchbend messages are similar in function to control change messages, they are a distinct type of message. The reason for this distinction is that the resolution of a pitchbend message is substantially higher than that of a conventional Controller message. The human ear is exceptionally sensitive to deviations in pitch, so the higher resolution is used because it relays pitchbend information more accurately.

---

## Program Change

These are MIDI messages that switch sound programs. Program numbers 1 through 128 can be changed via program change messages.

## Release

An envelope parameter. The term "Release" describes the descent rate of an envelope to its minimum value after a trigger is terminated. The Release phase begins immediately after the trigger is terminated, regardless of the envelope's current status. For instance, the Release phase may be initiated during the Attack phase.

## Resonance

Resonance is an important filter parameter. It emphasizes a narrow bandwidth around the filter cutoff frequency by amplifying these frequencies. This is one of the most popular methods of manipulating sounds. If you substantially increase the resonance, i.e., to a level where the filter begins self-oscillation, then it will generate a relatively clean sine waveform.

## Sustain

The term "Sustain" describes the level of an envelope that remains constant after it has run through the Attack and Decay phases. Sustain lasts until the trigger is terminated.

## Trigger

A trigger is a signal that activates events. Trigger signals are very diverse. For instance, a MIDI note or an audio signal can be used as a trigger. The events a trigger can initiate are also very diverse. A common application for a trigger is its use to start an envelope.

## Volume

The term describes a sound's output level.

## Wave

Here: A Wave is a digitally memorized reproduction of one single period of a periodic waveform, such as a sawtooth wave. Insofar it is identical with a Sample that is looped after one single wave pass. In contrast to the samples in a sampler, all Waves in the Waldorf Wave-table Synthesizers have the same lengths and are played back at the same pitch.

## Wavetable

One oscillator shape in the Largo bases on waveform sets, called Wavetables. Wavetables are a collection of up to 128 Waves. The playback sequence of the Waves can be dynamically altered, which results in interesting sound transformations. If the Waves in the Wavetable are similar

in shape, the result will sound smooth and pleasant. If, however, the shapes are completely different, the result will have wild spectral changes and may sound harsh.

## Product Support

If you have any questions about your Waldorf product, feel free to contact us via one of the four options listed below:

1. Send us an email message. This is the most efficient and fastest way to contact us. Your questions will be forwarded immediately to the resident expert and you will quickly receive an answer.

**support@waldorfmusic.de**

2. Send us a letter. It will take a bit longer, but it is just as dependable as an email.

**Waldorf Music GmbH**

**Neustrasse 12**

**53498 Waldorf, Germany**