



REAKTOR 5

Application Reference



The information in this document is subject to change without notice and does not represent a commitment on the part of Native Instruments GmbH. The software described by this document is subject to a License Agreement and may not be copied to other media. No part of this publication may be copied, reproduced or otherwise transmitted or recorded, for any purpose, without prior written permission by Native Instruments GmbH, hereinafter referred to as Native Instruments. All product and company names are TM or [®] trademarks of their respective owners.

Document authored by: Aleksander Rebane

Product Version: 5.5 (06/2010)

Document version: 1.0 (06/2010)

Special thanks to the Beta Test Team, who were invaluable not just in tracking down bugs, but in making this a better product.

Germany

Native Instruments GmbH
Schlesische Str. 28
D-10997 Berlin
Germany
info@native-instruments.de
www.native-instruments.de

USA

Native Instruments North America, Inc.
5631 Hollywood Boulevard
Los Angeles, CA 90028
USA
sales@native-instruments.com
www.native-instruments.com



© Native Instruments GmbH, 2010. All rights reserved.

Table of Contents

1	Introduction	13
1.1	Welcome to REAKTOR!	13
1.2	Three in One	13
1.3	The REAKTOR 5 Documentation	14
1.3.1	Who Should Read this Manual?	14
1.3.2	Notation Used in this Manual	14
1.3.3	Other Documentation	15
1.4	System Requirements	16
1.5	The REAKTOR 5 Installation	16
1.6	Overview	17
2	Names and Concepts You Should Know	18
3	Common Operations	25
3.1	Switching Views	25
3.1.1	Sidepane	26
3.1.2	Insert Pane	29
3.2	Working with Splits and Bookmarks	30
3.2.1	Panel/Structure	31
3.2.2	Split Views	34
3.2.3	Creating, Recalling, and Removing Bookmarks	43
3.3	Operating Ensembles	48
3.3.1	Loading a KOMplete Instrument	48
3.3.2	Opening an Ensemble	51
3.3.3	Playing an Ensemble	55
3.3.4	Saving an Ensemble	61
3.3.5	Creating a New Ensemble	62
3.4	Navigating with the Browser	63

3.5	Other Common Operations	67
3.6	REAKTOR as a Plug-In	73
3.6.1	The Autosave Feature	73
3.6.2	Window Sizes in the REAKTOR Plug-In	76
4	Snapshots	79
4.1	Recalling Snapshots	80
4.2	Adding and Removing Snapshots	83
4.3	Renaming and Copying Snapshots	87
4.4	Comparing Snapshots	88
4.5	Morphing Between Snapshots	90
4.6	Randomizing Snapshots	94
4.7	Snapshot Banks	98
4.7.1	Saving, Loading, and Creating Snapshot Banks	98
4.7.2	Cloning and Deleting Banks	101
4.7.3	Renumbering and Renaming Banks	103
4.8	Defining Snapshot Behavior for Instruments	105
5	Player/Recorder Bar	109
5.1	Using the Audio Player	109
5.1.1	Loading a File into the Player and Player Auto Mode	110
5.1.2	Playback with the File Player	111
5.2	Using the Audio Recorder	112
5.2.1	Recording an Audio Signal	113
5.2.2	File Recorder Properties	115
6	Sample Map Editor	118
6.1	Overview	118
6.1.1	Sample Maps	118
6.1.2	Viewing the Sample Map Editor	118
6.2	Creating a Simple Sampler and Sample Map	120

6.2.1	Building a Simple Sampler	121
6.2.2	Creating a Simple Sample Map	123
6.3	Editing the Sample List	124
6.3.1	Basic Editing	125
6.3.2	Sample List Functions	128
6.4	Settings for Individual Samples	133
6.4.1	Auditioning Samples	133
6.4.2	Setting Sample Playback and Loop Parameters	135
6.4.3	Using an External Sample Editor	138
7	Structure View	141
7.1	Inserting Objects into the Structure	141
7.1.1	Inserting Instruments into the Structure	141
7.1.2	Inserting Macros into the Structure	144
7.1.3	Inserting Modules and Core Cells into the Structure	145
7.2	Basic Operations in the Structure	146
7.2.1	Selecting Objects	146
7.2.2	Moving Objects	148
7.2.3	Duplicating Objects	149
7.2.4	Deleting Objects	150
7.2.5	Copying, Cutting, and Pasting Objects	151
7.2.6	Saving Objects	152
7.3	Connecting Objects	152
7.3.1	Dragging Wires between Objects	152
7.3.2	Port Types	154
7.3.3	Dynamic Ports	155
7.3.4	Port Order	156
7.4	Creating an Active Signal Flow	162
7.4.1	Activating Objects	162

7.4.2	Muting Objects	163
7.4.3	Signal Flows Where REAKTOR Is an Effect Unit	164
7.5	Choosing the Right Objects	165
7.5.1	Which Object Type to Choose	165
7.5.2	Panel Elements	169
7.5.3	Oscillators	169
7.5.4	Samplers	170
7.5.5	Filters	170
7.5.6	Audio Modifiers	170
7.5.7	Delay	171
7.5.8	Sequencers	171
7.5.9	Envelopes, LFOs (Control)	171
7.5.10	Math	172
7.5.11	MIDI In, MIDI Out	172
7.5.12	Signal Path	172
7.5.13	Event Processing	173
7.5.14	Auxiliary	173
7.5.15	Terminal	173
7.6	Object Properties	173
7.6.1	Function Page	174
7.6.2	Info Page	174
7.6.3	View Page	175
7.6.4	Connections Page	175
8	Panel Editing	176
8.1	Setting Panel Layout Properties	176
8.1.1	The Panel and Instrument Layouts	176
8.1.2	Working with Panelsets	177
8.1.3	Working with Panel Views	182

8.2	Inserting and Modifying Panel Controls	185
8.2.1	Faders and Knobs	185
8.2.2	Buttons	189
8.2.3	Switches	191
8.2.4	Lists	202
8.3	Changing the Instrument Appearance	206
8.3.1	Editing the Instrument Panel Colors	206
8.3.2	Choosing the Background Picture	210
8.4	Choosing the Macro Appearance	213
8.5	Customized Panel Elements	217
8.5.1	Supported Formats and Loading Pictures	217
8.5.2	Transparency	219
8.5.3	Knob Skins and Animations	221
8.5.4	Resizability and a Custom Control	225
8.5.5	General Guidelines for Skins	234
9	The Primary Level Paradigm	237
9.1	Audio signals	237
9.2	Event Signals	238
9.2.1	An Event Is Tied to a Point in Time	239
9.2.2	The Order of Events	241
9.2.3	Events from the Same Source	242
9.2.4	Controlling the Event Order	242
9.2.5	Initialization	245
9.3	Polyphony	250
9.3.1	Polyphonic and Monophonic Signals	251
9.3.2	Changing the Number of Polyphonic and Unison Voices	256
9.3.3	Changing an Instrument's Voice Allocation Settings and Tuning	260
9.3.4	Using Voices for Parallel Processing	263

9.4	Different Value Ranges and Scales in REAKTOR	267
9.4.1	The MIDI Scale	268
9.4.2	The Linear Frequency and Logarithmic Pitch Scales	272
9.4.3	The Linear Amplitude and Logarithmic Decibel Scales	273
10	Working with Sampler Modules	274
10.1	Managing Your Samples and RAM	274
10.1.1	Sample Files and RAM	274
10.1.2	Multiple Use of Identical Samples	275
10.1.3	Missing Samples	276
10.1.4	Storing Samples with Modules	276
10.1.5	Sample Analysis	277
10.2	Differences between the Sampler Modules	277
10.2.1	“Conventional” Sampler Modules	278
10.2.2	Grain Resynthesis Modules	279
10.2.3	The Sample Lookup Module and Wavetables	280
10.3	Setting the Properties of Sampler Modules	281
10.3.1	Working with the Function page	281
10.3.2	Working with the View Page	284
11	Working with Table Modules	288
11.1	What Is a Table Module?	288
11.2	The Function page of the Audio Table: Recording and Playing a Signal	290
11.2.1	What Is the Goal?	290
11.2.2	Creating a Table, Table Files, and Linking Table Modules	291
11.2.3	Setting the Range, Resolution, and Display Units of Table Values	294
11.2.4	Clip, Wrap, and Interpolation Modes	299
11.2.5	Setting Up the Generator Signal and Activating the Audio Table	301
11.2.6	Setting Up the Recording Pointer Using a Ramp Module and the “DX” Output Port	303

11.2.7	Writing to the Table and Reading the Result	307
11.3	Graph Display and Appearance Properties	308
11.3.1	Graph Style	308
11.3.2	Scrollbars and Zooming	315
11.3.3	Displaying a Grid in the Table	319
11.3.4	Setting Basic Visibility Settings for the Table Modules	322
11.3.5	Table Modes and Basic Table Editing Operations	324
12	Working with the Sine and Modal Bank Modules	335
12.1	Working Principles of the Sine and Modal Bank Modules	335
12.2	Building a Simple Structure with the Sine Bank Module	338
12.2.1	Setting the Maximum Number of Partial	338
12.2.2	Sending Pitch Information to the Sine Bank Module	339
12.2.3	Creating a Macro for the Amplitude Controls of the Partial	342
12.2.4	Creating an Amplitude Control for a Partial	343
12.2.5	Duplicating the Structure of the First Amplitude Control	346
12.2.6	Routing the Amplitude Control Parameters to the Sine Bank Module	349
12.2.7	Creating Frequency Ratio Controls for Partial	351
12.2.8	Routing the Frequency Control Parameters to the Sine Bank Module	352
12.2.9	Creating an Amplitude Envelope for the Sine Bank	354
12.2.10	Arranging the Instrument Panel of your Additive Synth	357
12.3	Building a Simple Structure with the Modal Bank Module	359
12.3.1	Modifying the Additive Synth Structure for the Modal Synth Tutorial	359
12.3.2	Creating Decay Time Controls for Partial	363
12.3.3	Routing the Decay Control Parameters to the Modal Bank Module	366
12.3.4	Sending the Damping Parameter to the Modal Bank Module	368
12.3.5	Using a Short Impulse as an Exciter Signal	370
12.3.6	Using White Noise as an Exciter Signal	375
12.3.7	Using the Compensated Input Port for Sustained Exciter Signals	377

12.4	The Sine Bank Module	382
12.4.1	Overview	382
12.4.2	Ports	383
12.4.3	Properties: Function Page	385
12.5	The Modal Module	387
12.5.1	Overview	387
12.5.2	Ports	389
12.5.3	Properties: Function Page	391
13	Instruments: Interfaces for Your Structures	394
13.1	Why You Should Care About Instruments	394
13.2	Setting up MIDI Connections	396
13.2.1	MIDI In	397
13.2.2	MIDI Out	402
13.3	MIDI and the Instrument Panel	404
13.4	Setting up OSC Connections	407
13.4.1	Using REAKTOR with an OSC Device	408
13.4.2	Defining OSC Synchronization Settings	411
14	Advanced Operations and Preferences	414
14.1	CPU, Debugging, and Optimizing	414
14.1.1	Monitoring CPU Load	415
14.1.2	REAKTOR's Debugging Tools	417
14.1.3	Tips to Reduce CPU Load	423
14.2	Using the Internal Connection Protocol	426
14.2.1	Connecting Two Panel Elements Via the IC Protocol	426
14.2.2	Modulating a Control with a Signal from the Structure	429
14.3	Automation ID Management	431
14.3.1	Global Automation IDs	431
14.3.2	Automation IDs within an Instrument	434

14.3.3	Automatable Modules in REAKTOR	437
14.4	Advanced REAKTOR Preferences	438
14.4.1	The Options Tab	439
14.4.2	Directories Tab (Setting File Paths)	441

1 Introduction

1.1 Welcome to REAKTOR!

REAKTOR is a musical instrument. It creates sound. You can play it like any other music software – stand-alone with a MIDI keyboard, or as plug-in within a sequencing host. But you define what type of instrument it is. Each time you launch REAKTOR, it follows your musical needs. You need a synthesizer? You need a drum computer? You need this particular noise-making thing you never heard before? Just load the according instrument file and start playing.

The full version of REAKTOR comes with over 50 individual instruments. An included collection of instruments from earlier REAKTOR versions unfolds more than ten years of legacy within electronic music. And the online library holds more than 3000 instruments of any type you can imagine.

If you don't own the full version, you can still use the engine's musical power. A continuously growing set of KOMplete Instruments is separately available for REAKTOR. When you load any of those instruments, REAKTOR switches into its Player mode.

However, loading pre-configured instruments is only the surface. You can literally define the instrument REAKTOR. You can edit your favorite instrument's signal processing. Adjust its technical structure to a specific song's needs. Or start building your own instrument from scratch, to design your very personal signature sound.

1.2 Three in One

Three roads lead to this document:

- You bought, installed and activated REAKTOR, as a single product or as part of a bundle like KOMplete. Then you use the full license for REAKTOR.
- You bought, installed and activated any of the KOMplete Instruments for REAKTOR, or a bundle of these instruments like KOMplete Elements. Additionally, you installed REAKTOR as a player application for these instruments. Then you use the player license for REAKTOR.

- You did not buy and activate any of the products above but installed REAKTOR and any number of KOMLETE Instruments as demo versions. Then you use the demo license for REAKTOR.

The separate Native Instruments Setup Guide explains installation and activation in all details.

According to the license used, the single REAKTOR application can run in three distinct modes:

- In Full mode, all functionality is available without restrictions.
- In Player mode, you can play instruments but you cannot edit their internal structure. Instruments cannot be saved, but all settings are saved with your host's project.
- In Demo mode, you have all functionality but the application will stop making sound after 30 minutes. Storing is disabled

1.3 The REAKTOR 5 Documentation

1.3.1 Who Should Read this Manual?

This is REAKTOR's Application Reference Manual covering all the features and aspects of the Full version of REAKTOR software in detail.

- If you are using the Full version of REAKTOR, have already read the Getting Started Manual, and would like to learn about all aspects of the REAKTOR application in detail, then this manual is the right place to start.
- If you are using the Demo version of REAKTOR, this manual will give you a taste of what is possible with the Full version of REAKTOR. Note that although in Demo mode you have all the functionality of REAKTOR, you are not able to store any work and the application will stop making sound after 30 minutes.
- If you are using REAKTOR's Player Mode, it might be worth having a look at this manual as it will provide a deeper understanding of how to use REAKTOR's Instruments

1.3.2 Notation Used in this Manual

This subsection introduces you to the symbols and text formatting used throughout this manual.

Marginal Notes

The manual uses marginal notes to point out special facts and to warn of potential dangers. The icons included in the marginal notes let you see what kind of information is provided:



The exclamation mark stands for information that you should take note of to avoid common mistakes. Read carefully whenever you see this symbol.



The light bulb icon introduces useful extra information. This information may often help you solve a task more efficiently, but does not necessarily apply to the setup of operating system you are using.

Special Formatting

In this manual, distinctive formatting has been applied in order to let you recognize certain elements in the text at a glance:

- Text appearing in (drop-down) menus (such as *Open...*, *Save as...* etc.) and paths to locations on your hard drive or other storage devices is printed in *italics*.
- Text appearing elsewhere on the screen (labels of buttons, controls, text next to checkboxes etc.) is printed in grey with slightly condensed letters. Whenever you see this formatting, you can turn to the screen for reference.
- Important names and concepts are printed in **bold**.

1.3.3 Other Documentation

This is REAKTOR's Application Reference Manual covering all the features and aspects of the REAKTOR software in detail. Besides the Application Reference Manual, REAKTOR documentation comprises the following elements:

- **Getting Started Manual:** If you would like to quickly get going without getting too deep into the specifics, we recommend you read the Getting Started Manual which is either included in the box (if you bought REAKTOR) or as a PDF file in the subfolder “Documentation”. The Getting Started Manual will give you a fast overview of REAKTOR's main features and functions.
- **Instrument Reference:** The Instrument Reference describes all instruments that come with REAKTOR as its Factory Content. Read this if you want to explore the musical options of each individual instrument in greater detail.
- **The Module and Macro Reference:** The Module and Macro Reference provides minute information about all parts available when designing own instruments in REAKTOR. If you want to leave the beaten path, this document will become your guide.
- **The Core Reference:** The Core Reference, is divided into two parts: The first half is a tutorial that explains the Core level of REAKTOR where you can realize your own low-level digital signal processing. The second half functions as that levels reference, listing all of its components.

All of these documents are available from the application's *Help* menu. Aside from the manuals coming with REAKTOR, be sure to visit the online forum. The huge community of REAKTOR users is sharing tips and tricks, and will help you with specific questions.

1.4 System Requirements

For current information about system requirements and compatibility with current and legacy operating systems, please see our website at:
www.native-instruments.com/REAKTOR

1.5 The REAKTOR 5 Installation

To learn how to install REAKTOR on your computer, please read chapter 2 in the Getting Started Manual. After following the installation instructions listed there, you should have done the following things in the Audio and MIDI Settings dialog window:

- Set up an audio device and a corresponding driver for REAKTOR to use.
- Chosen input and output routing for the audio device.

- Set up available MIDI hardware by specifying the input and output routing of MIDI signals.

Now you are ready to delve into REAKTOR in more detail!

1.6 Overview

This section will introduce you to the REAKTOR software's user interface. If you need to look up a certain user interface element's name, you can return to this chapter at any time for a refresher!

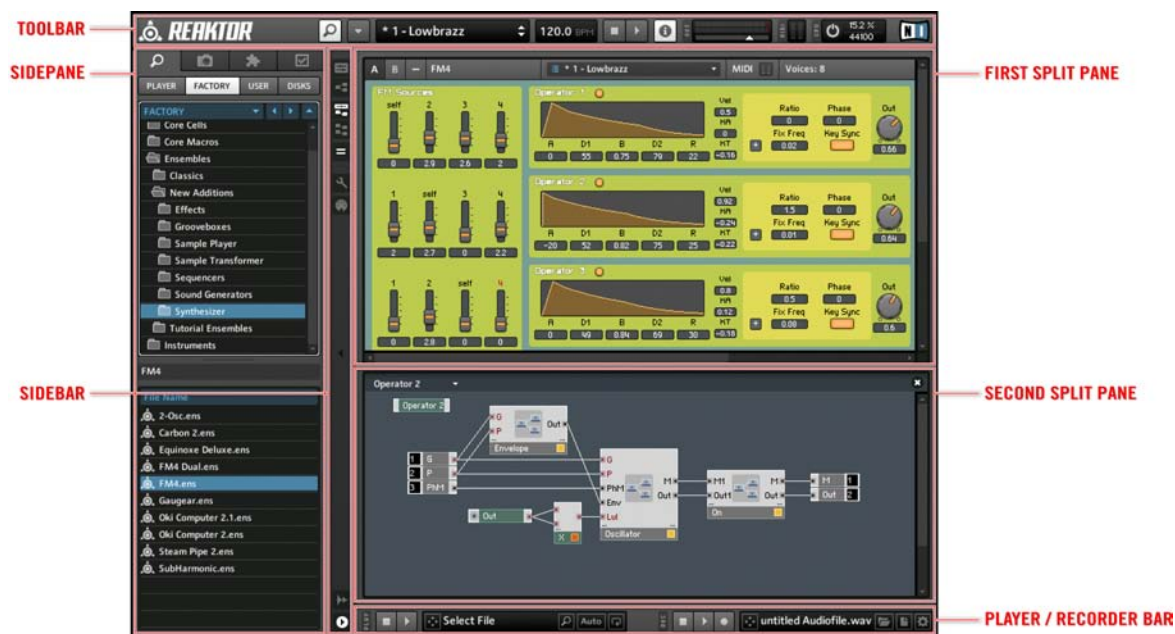


Fig. 1.1 The REAKTOR application

2 Names and Concepts You Should Know

We will start with a list defining the most important concepts and names.

Instrument

A REAKTOR Instrument is an Object that has its own internal Structure, MIDI processing, Instrument Panel, and Snapshots. It operates much like other software music instruments. However, what the Instrument actually does depends on its Structure. For example, an Instrument can be a synthesizer, a sequencer, an effect, or many other things you can come up with. Although the internal Structure of the Instrument is relevant, you should view an Instrument as a container for this Structure, providing it with additional features such as the Instrument Panel and Snapshots. The Instrument Panel is where the controls for your Instrument are placed. These controls can be tweaked and their states saved in Snapshots (also called presets or patches). Thus, you can use the Instrument and create and save different Snapshots without ever having to access the Structure. See chapter [↑13, Instruments: Interfaces for Your Structures](#) for more thorough documentation on the features and settings of Instruments. Instrument files have the extension *.ism.

Ensemble

You can view the Ensemble as an Instrument that has additional information stored with it. An Ensemble holds all the relevant information to a session of REAKTOR, including all Instruments, their Snapshots, and global settings such as Sample Rate, Control Rate, and BPM information. When you load an Ensemble, you restore the state of REAKTOR as it was when the Ensemble was saved. An Ensemble can also be loaded into another Ensemble as an Instrument. In that case the global settings of the Ensemble being loaded are not used. Ensemble files have the extension *.ens.

KOMLETE Instrument

The KOMLETE Instrument (or Player Instrument) is the only file type you can use with the Player license. Technically it is close to an Ensemble (it connects to the inputs and outputs, for instance) but it is called instrument as it is a single unit from a musical point of view. You can quickly navigate to KOMLETE Instruments using the Browser

Panel View

The Panel View displays the Instrument Panels of your Instruments. It is here that you interact with your Instruments using controls and displays. These so-called Panel Elements appear on your Instrument Panel if the corresponding Module has been inserted into the Structure of the Instrument.

Structure View

The Structure View displays the inner Structure of your Instrument. This Structure represents the processes that comprise the functionality of the Instrument. See chapter [↑7](#), [Structure View](#) for more information on working in the Structure View.

Splits

When building Structures it is often desirable to simultaneously view two Structures or a Structure and the Instrument Panel. The “Split” feature of REAKTOR enables you to split the screen into two parts and choose the content for each part of the screen separately. See section [↑3.2, Working with Splits and Bookmarks](#) for more information on working with Splits.

Bookmarks

With the “Bookmarks” feature you can bookmark frequently used parts of your Structure for quick access at some later point. See section [↑3.2, Working with Splits and Bookmarks](#) for more information on working with Bookmarks.

Browser

The Browser enables you to quickly access files relevant to the use of REAKTOR. These files, among others, include Ensemble, Instrument, Macro, Core Cell, Core Macro, and Audio files.

Snapshots

Snapshots (also called presets, programs, or patches) represent the state of all controls on the corresponding Instrument's Panel. Snapshots are stored in Snapshot Banks which turn can be saved to Snapshot Bank files on your hard disk. For more information on Snapshot management using the Snapshot tab, see chapter [↑4](#), [Snapshots](#).

Hierarchy Levels

REAKTOR Structures follow a certain hierarchy, shown in the figure below. On the large scale, one can discern two levels of functionality: the Primary Level and the Core Level. Instruments hold Primary Level Structures. They can hold other Instruments, Primary Level Macros, Primary Level Modules, and Core Cells. Core Cells hold Core Structures and hold Core Macros and Core Modules. The basic concepts behind working in the Structure and choosing on which hierarchy level to work in, are outlined in chapter [↑7, Structure View](#). The Primary Level is described in more detail in chapter [↑9, The Primary Level Paradigm](#), whereas the Core Level is described separately in the “Core Reference” that is available as a PDF file in the subfolder “Documentation” of REAKTOR's installation folder.

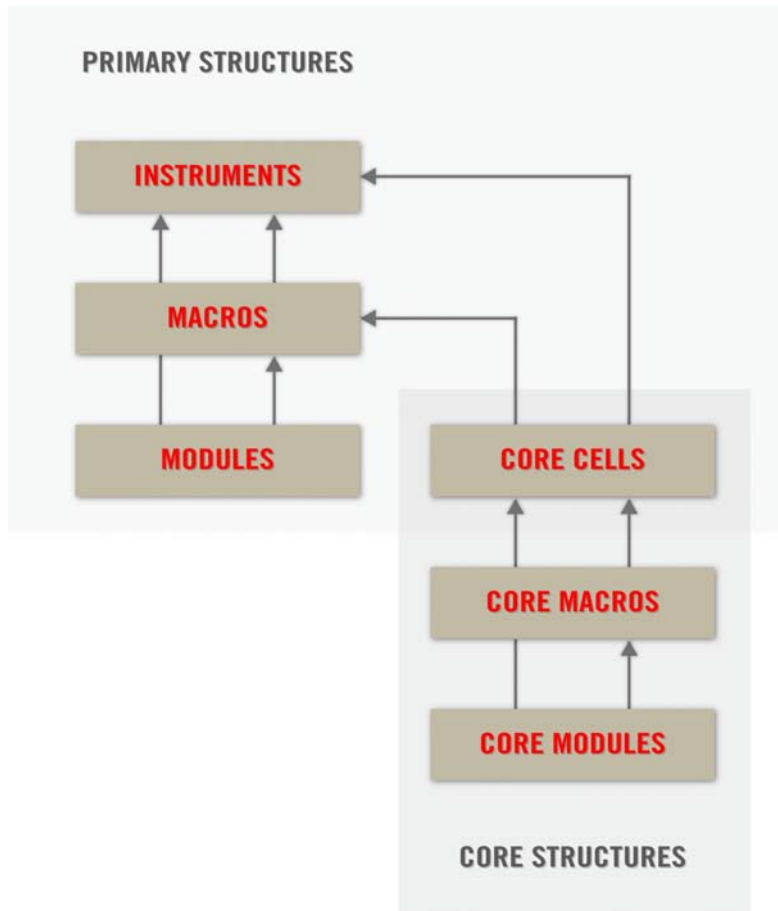


Fig. 2.1 The Primary Level comprises Instruments, Macros, Modules, and Core Cells. Core Structures are contained within Core Cells and comprise Core Macros and Core Modules.

Primary Level

The Primary Level is the upper of the two Structure levels in REAKTOR. You can place Instruments, Macros, Modules, and Core Cells into the Primary Level and thus create a Structure that serves a musical purpose. Some of the Modules in the Primary Level appear

on the Instrument Panel. This is not the case for the Core Level. Read chapter [↑9, The Primary Level Paradigm](#) for more information on the paradigm under which the Primary Level operates.

Primary Level Modules

Primary Level Modules are the most fundamental building blocks that make up Primary Level Structures. They include all functions you might be familiar with from modular synths: oscillators, filters, LFOs, envelopes, and so on. Additionally, more fundamental signal processing functions such as signal routing and mathematical operations are realized with Primary Level Modules. See section [↑7.5, Choosing the Right Objects](#) for more information on choosing the right Modules for your Structure.

Primary Level Macros

Similar to Instruments, Macros are also containers for your Structure, but with one big difference: Macros do not have dedicated features such as Snapshots or separate Instrument Panels. Macros serve to promote the building hierarchical Structures that grant the beholder a better overview of the signal flow and facilitate reusability of certain Structures. See chapter [↑7, Structure View](#) for more information on Primary Level Macros.

Core Level

Core Level is the lower of the two Structure levels in REAKTOR. It provides somewhat different functionality than the Primary Level and is more suitable for low-level signal processing tasks. Core Structures are contained in Core Cells which themselves lie in the Primary Level. The Core Level consists of Core Modules and Core Macros. An introduction to the Core Level is found in the “Core Reference” that is available as a PDF file in the subfolder “Documentation” of REAKTOR's installation folder.

Core Cells

Core Cells can be placed into the Primary Level much like Primary Level Modules. However, Core Cells hold Structures of their own — Core Structures. Essentially, Core Cells serve to interface between the Primary Level and Core Level. There are two different types of Core Cells: Event Core Cells and Audio Core Cells. Please see chapter [↑7, Structure View](#) and the “Core Reference” for more information on Core Cells.

Core Modules

Core Modules are the most fundamental building blocks in Core Level Structures. Core Modules are much more low-level than Primary Level Module. Moreover, there are only a few Core Modules compared to the large variety of Primary Level Modules. For more information on Core Modules, please refer to the “Core Reference”.

Core Macros

Core Macros operate much like Primary Level Macros. They are containers for your Structures without applying new settings to them. Core Macros promote the building hierarchical Structures that grant the beholder a better overview of the signal flow and facilitate re-usability of certain Structures. For more information on Core Macros, please refer to the “Core Reference”.

Object

To simplify terminology, an Object is considered to be anything that can be loaded into the Structure. Instruments, Ensembles, Primary Level Modules, Primary Level Macros, Core Cells, Core Modules, and Core Macros are Objects.

Object Properties

Many Objects can have varying functionality. This functionality is determined in the Object Properties. Please refer to section [↑3.1, Switching Views](#) to see how to access the Object Properties.

MIDI Learn

“MIDI Learn” is a feature which allows you to assign controls on the Instrument Panel to MIDI controllers simply by moving the desired MIDI controller such as a knob or slider. Please read subsection [↑3.4, Navigating with the Browser](#) for instructions on how to do this.

Wires and Ports

Wires are vessels for the different types of signals which flow through REAKTOR Structures. Wires always run between two Objects. The points at which a wire enters and leaves an Object are called ports. There are different types of signals for which there are different types of ports. Please refer to section [↑7.3, Connecting Objects](#) for more information on ports and connecting Objects with wires.

3 Common Operations

This chapter shows you the way around in the REAKTOR application. You will learn how to do the most common operations like navigating through REAKTOR, loading, playing, and saving Ensembles, and using other features like “MIDI Learn” or changing the Sample Rate.

3.1 Switching Views

This section describes how to navigate through the various Views in REAKTOR.


 All Views are accessible also by pressing the Menu button and choosing the corresponding menu entry from the *View* submenu, as shown in the figure below. In stand-alone mode you can also access these menu entries from the *View* application menu.



Fig. 3.1 The Menu button

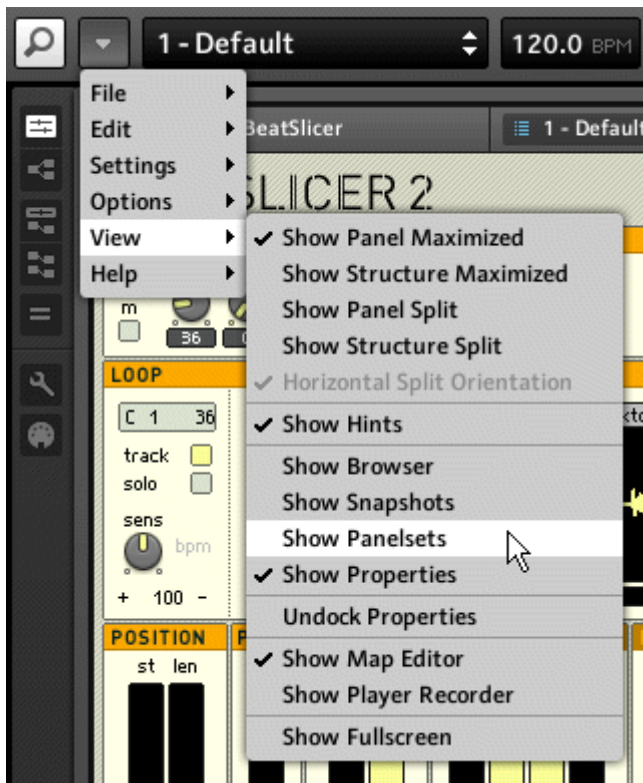


Fig. 3.2 You can access all Views from the View submenu.



In Standalone mode you can toggle the Fullscreen mode either by pressing the **Menu** button and choosing the *View > Show Fullscreen* menu entry or with the keyboard shortcut **F11**.

3.1.1 Sidepane

The Sidepane holds the Browser, Snapshot, Properties, and Panelsets tabs. To view these tabs, you need to open the Sidepane. Here's how:

- To open the Sidepane, either click on the **Sidepane** button or click on the arrow in the Sidebar (shown in the figure below).



Fig. 3.3 You can access the Sidepane by either clicking on the Sidepane button (magnifying glass icon) or by clicking on the arrow in the Sidebar. These buttons are marked in red.

Browser

- To access the Browser first open the Sidepane (see above) and then press the **Browser** tab. You can also use the shortcut key F5 to show and hide the Browser.



Fig. 3.4 The Browser tab

Snapshot Tab

- To access the Snapshot tab first open the Sidepane (see above) and then press the Snapshot tab. You can also use the shortcut key F6 to open the Snapshot tab. Another quick way to navigate to an Instrument's or Ensemble's Snapshot tab is by right-clicking it in Panel View and choosing the *Snapshots* menu entry.



Fig. 3.5 The Snapshot tab

Panelsets

- To access Panelsets first open the Sidepane (see above) and then press the Panelsets tab. You can also use the shortcut key F7 to show and hide the Panelsets.



Fig. 3.6 The Panelsets tab

Properties

- To access the Properties first open the Sidepane (see above) and then press the Properties tab. You can also use the shortcut key F8 to show and hide the Properties. Another common way to access a Module's Properties is by double-clicking the Module in the Structure or its Panel representation on the Instrument Panel. You can also right-click an Object and choose the **Object Name* Properties* menu entry. This works in both the Panel and Structure Views.



Fig. 3.7 The Properties tab icon

- You can also undock and re-dock the Properties tab from the Sidepane. To do this, press the Undock button or press the **Menu** button and choose the *View > Undock Properties* menu entry. Alternatively, you can use the keyboard shortcut Shift+F8 to toggle between docked and undocked Properties.



Fig. 3.1 The Undock button

3.1.2 Insert Pane

Sample Map Editor

- Press the **Sample Map** button (shortcut key F9) to show and hide the Sample Map Editor.



Fig. 3.8 The Sample Map button

Player/Recorder

- Press the **Player/Recorder** button (shortcut key F10) to show and hide the Player / Recorder Bar.



Fig. 3.9 The Player/Recorder button

3.2 Working with Splits and Bookmarks

REAKTOR offers two modes of interaction with your Instruments. The first mode of interaction, naturally, is playing the Instrument. This happens using the Panel View which shows the user interface of the Instrument with its faders, knobs, and displays. The second mode of interaction is modifying and building the Instrument. This happens in the Structure View. It shows what is “under the hood” of your Instrument: all the Modules, Macros, Core Cells, and their connections. You can either view only the Panel, only the Structure. By using a Split you can simultaneously view a Panel and a Structure or two Structures. Furthermore, Bookmarks let you store important locations in your Instrument's Structure so that you can always quickly navigate to these locations with little effort. This section describes how you can switch to the different Views and work with Bookmarks.



All Views are accessible also by pressing the Menu button and choosing the corresponding menu entry from the *View* submenu, as shown in the figure below. In stand-alone mode you can also access these menu entries from the *View* application menu.



Fig. 3.10 The Menu button

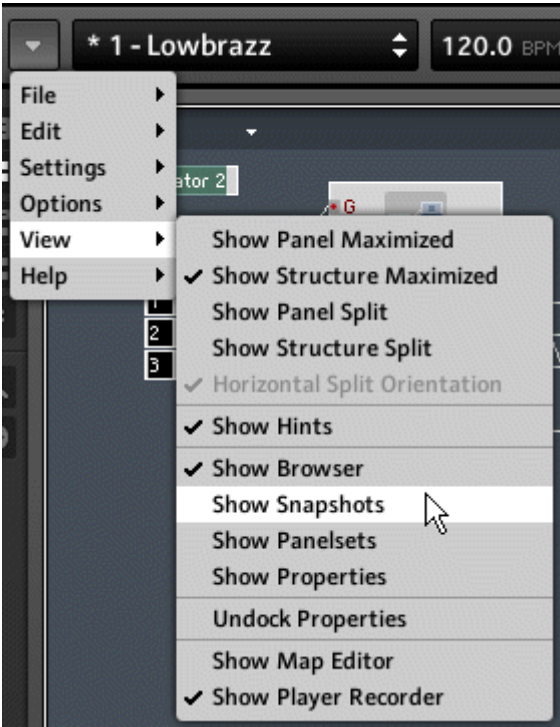


Fig. 3.11 You can access all Views from the View submenu.

3.2.1 Panel/Structure

You can switch between the Instrument Panel View and the Structure View using the Panel and Structure buttons (see figure below).



Fig. 3.12 Use the Panel and Structure buttons to view the Instrument Panel and Structure, respectively.

- To view only the Instrument Panel, press the **Panel** button (shortcut key F1). You can use this button to toggle between the Panel View and the previous View. Alternatively, if you are in the highest hierarchy level of your Structure, you can right-click the Structure background and choose the *Show Panel* menu entry.



Fig. 3.13 The Panel button



Fig. 3.14 The Panel View.

- To view only the Structure, press the Structure button (shortcut key F2). You can use this button to toggle between the Structure View and the previous View. You can also enter an Instrument's Structure by right-clicking on its Panel and choosing the *Instrument Structure* menu entry.



Fig. 3.15 The Structure button

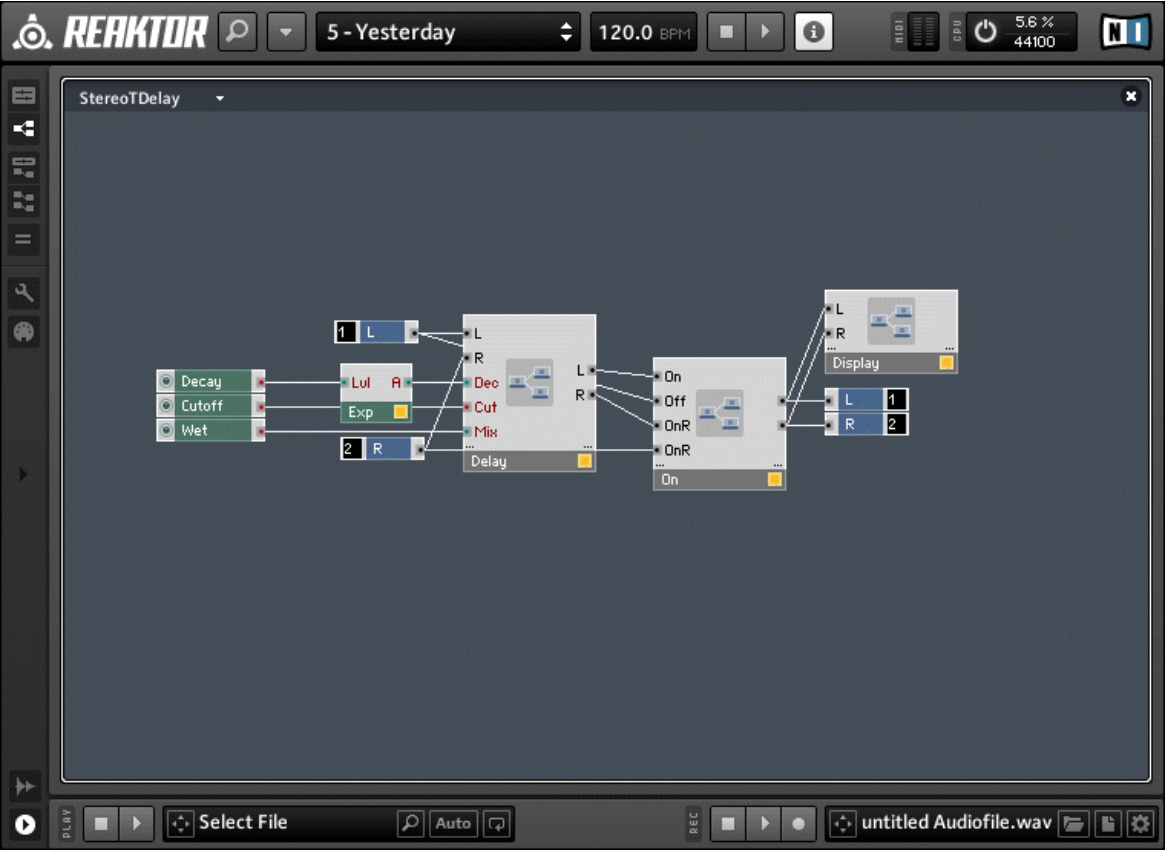


Fig. 3.16 The Structure View

3.2.2 Split Views

As mentioned above, you can simultaneously view either the Panel and a Structure or two Structures. You can directly access both Views using the two buttons shown in the figure below.

Accessing the Split Views



Fig. 3.17 Panel Split and Structure Split buttons

Use the Panel Split and Structure Split buttons to simultaneously view the Panel and a Structure or two Structures, respectively.

- To simultaneously view the Instrument Panel and the Structure, press the **Panel Split** button (shortcut key F3 for Windows, Alt+F1 for Mac OS X). You can use this button to toggle between the Panel Split and the previous View. While in Panel View, you can also right-click on an Object (Ensemble, Instrument, Macro, or Module) and choose the *Show Structure in Split* menu entry.



Fig. 3.18 The Panel Split button



Fig. 3.19 The Panel Split view

- To simultaneously view two Structures, press the Structure Split button (shortcut key F4 for Windows, Alt+F2 for Mac OS X). You can use this button to toggle between the Structure Split and the previous View.



Fig. 3.20 The Structure Split button

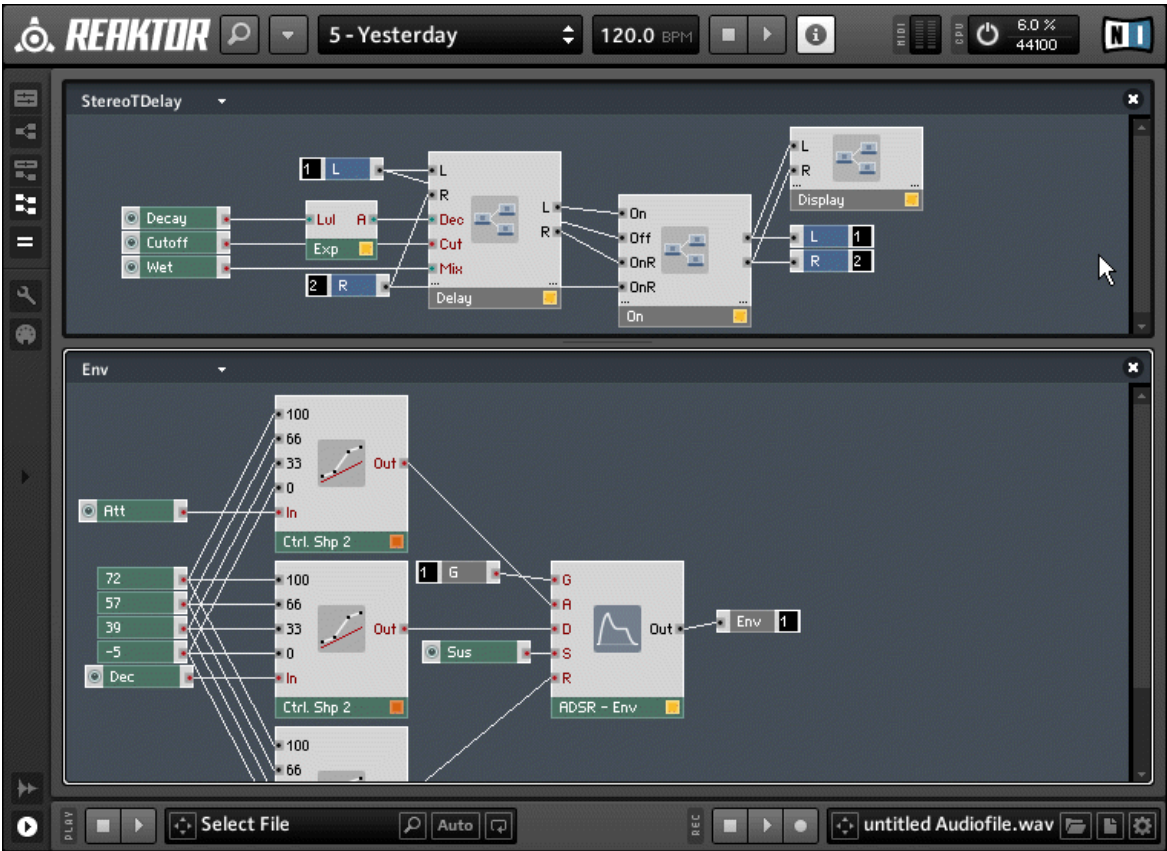


Fig. 3.21 The Structure Split view



You can easily move the Split by clicking and dragging the border dividing the two Split Panes!

Swapping Structures

- In Structure Split View both Split Panes hold a Structure. If you want to swap the Structures between their Split Panes choose the *Swap Structures* menu entry from the Navigation menu (see figure below).

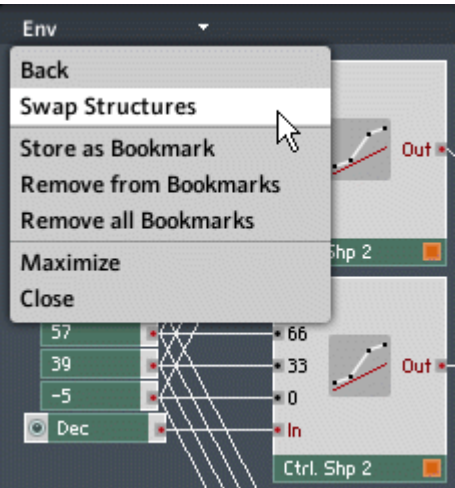


Fig. 3.22 In Structure Split View use the Swap Structures menu entry from the Navigation menu to exchange the Structures in the two Split Panes.

Changing the Split Orientation

A Split View divides the screen horizontally or vertically. To toggle between these two orientations, press the Split Orientation button (shortcut keys Ctrl+H for Windows, Cmd+H for Mac OS X).



Fig. 3.23 The Split Orientation button

Maximizing and Closing Structure Splits

- To close a Structure Split, press the Close Split Pane button in the top right of the Split Pane or choose the *Close* menu entry from the Navigation menu as shown below.



Fig. 3.24 The Close Split Pane button

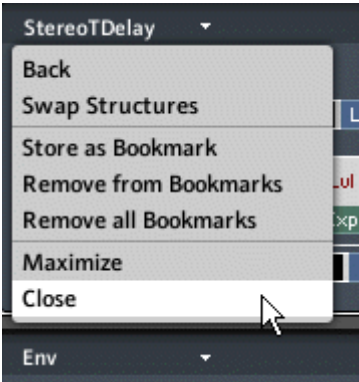


Fig. 3.25 To close a Structure Split you can use the Close menu entry from the Navigation menu

- To maximize a Structure Split Pane, choose the *Maximize* menu entry from the Navigation menu (see figure below).

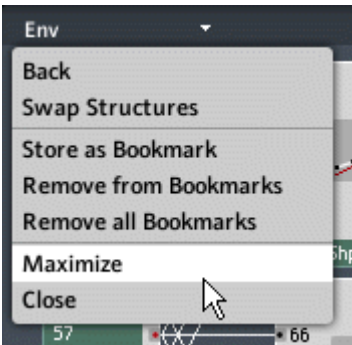


Fig. 3.26 Use the Maximize menu entry from the Navigation menu to only see the corresponding Structure.

Further Navigation

To navigate one hierarchy level higher in the Structure View, double-click the Structure background. Alternatively you can right-click the Structure background and then choose the *Parent* menu entry (see figure below).

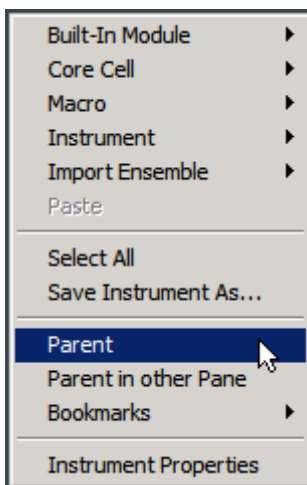


Fig. 3.27 With the Parent menu entry you can navigate one hierarchy level higher.

When in Structure View you can right-click an Object and choose the *Structure in other Pane* menu entry (see figure below). This brings you to Structure Split View and shows the contents of the corresponding Object in the other Split Pane. If you are in Structure Split View clicking on the *Structure in other Pane* menu entry replaces the contents of the other Split Pane by the contents of the Object.

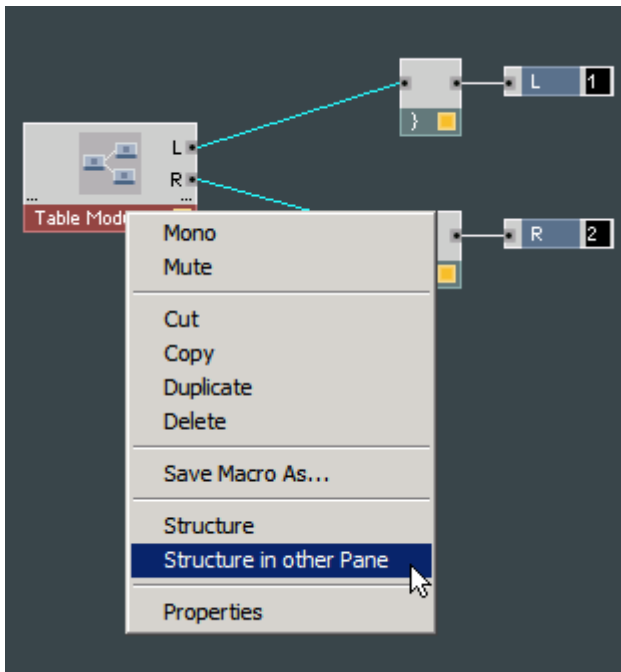


Fig. 3.28 To display the contents of an Object in the other Split Pane, choose the Structure in other Pane menu entry.

- Similarly, right-click on the Structure background and choose the *Parent in other Pane* menu entry to have the Structure of the parent hierarchy level shown in the other Split Pane (see figure below).

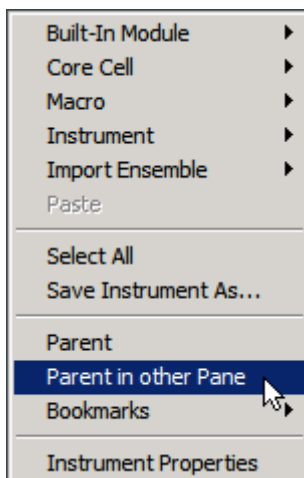


Fig. 3.29 To display the higher hierarchy level in the other Split Pane, choose the Parent in other Pane menu entry.

- To access the Panel View from the first Split whilst operating in the highest hierarchy level, right-click the Structure background and choose the *Show Panel* menu entry. If you are on the highest Structure hierarchy level and in the second Split, right-click on the Structure and choose the menu entry *Show Panel in First Pane* to replace the contents of the first Split Pane with the Panel View (see figure below).

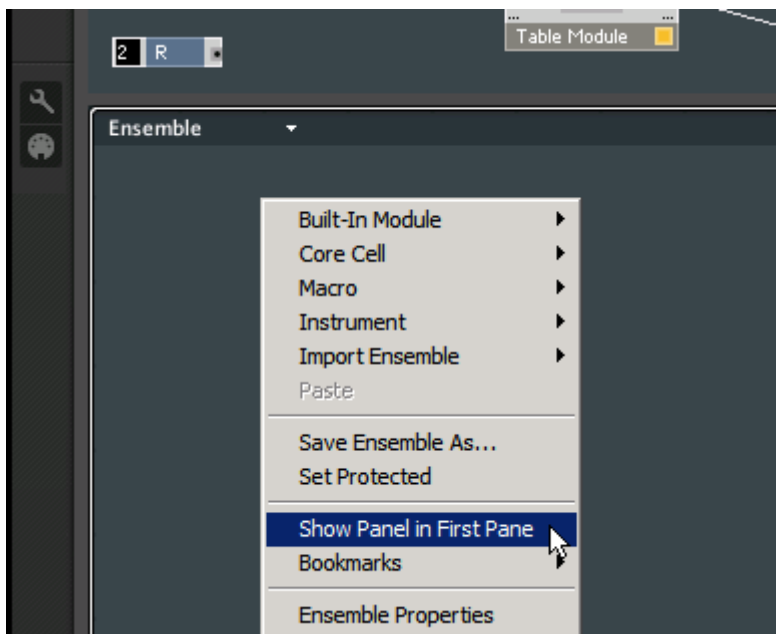


Fig. 3.30 When you are already in the highest hierarchy level and in the second Split Pane, you can use the Show Panel in First Pane menu entry to go to the Panel Split View.

3.2.3 Creating, Recalling, and Removing Bookmarks

Bookmarks allow you to store important locations in your Instrument's Structure and quickly navigate to them with little effort. This subsection discusses how to store, recall, and remove Bookmarks.

Creating a Bookmark

- To store a location in your Structure as a Bookmark, first navigate to the desired location. Then click on the Navigation drop-down menu in the top left of the Structure and choose the *Store as Bookmark* menu entry (see figure below). Note that the Navigation menu is labeled with the name of the hierarchy level you are in. For example, if you are looking at the Structure of a Macro called “Display”, then the menu is also labeled as “Display”. This is the name that will be used for the Bookmark when the location is bookmarked.

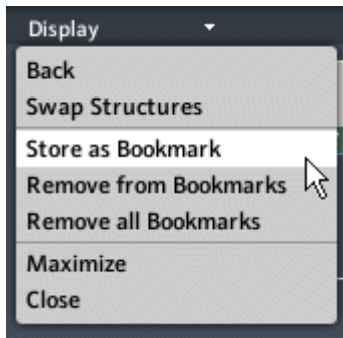


Fig. 3.31 With the Store as Bookmark menu entry you can save the current Structure location as a Bookmark.

After creating a Bookmark, the Navigation menu gets an entry that corresponds to that Bookmark.

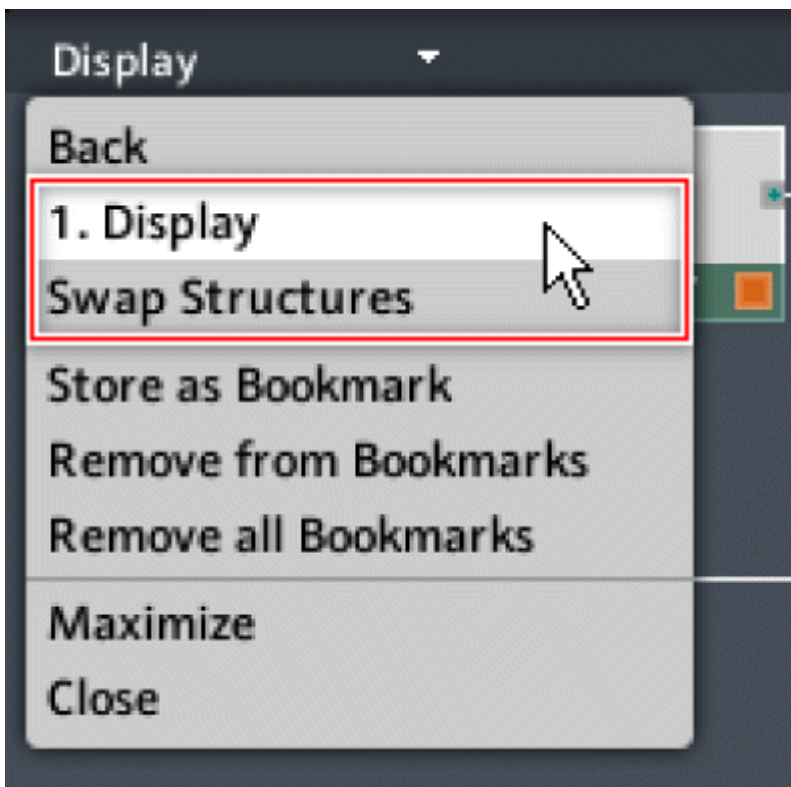


Fig. 3.32 After storing a Bookmark, the Bookmark appears in the Navigation menu.

Recalling a Bookmark

- You can easily recall a Bookmark by choosing the corresponding menu entry from the Navigation drop-down menu (see figure below). You can also right-click the Structure background and choose the corresponding menu entry from the *Bookmarks* submenu.

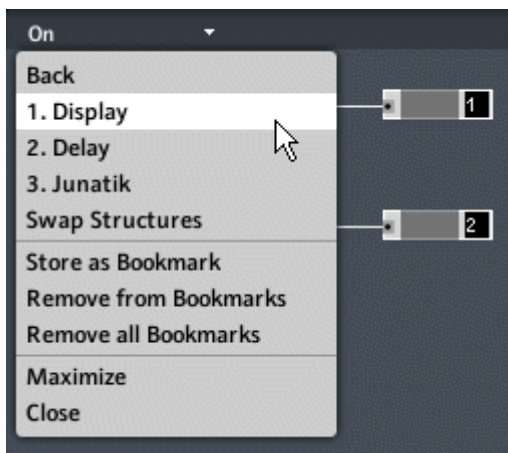


Fig. 3.33 You can recall a Bookmark by choosing the Bookmark's corresponding menu entry from the Navigation menu.



Bookmark settings are saved with the Ensemble. Therefore when you are building an Ensemble you can reuse your favorite locations between REAKTOR sessions without having to recreate their Bookmarks. However, note that the Bookmark settings are not stored with Instruments, Macros, or Core Cells.

Removing Bookmarks

- To remove a Bookmark, navigate to the Bookmarked location (for example by recalling the Bookmark) and then choose the *Remove from Bookmarks* menu entry from the Navigation menu (see figure below).

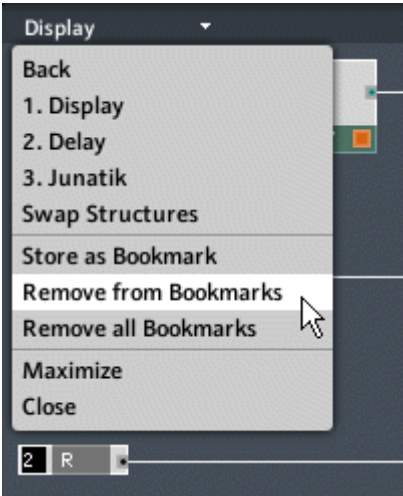


Fig. 3.34 Use the Remove from Bookmarks menu entry to remove the currently active Bookmark from the list of stored Bookmarks

- You can also delete all Bookmarks by choosing the *Remove all Bookmarks* menu entry from the Navigation menu (see figure below).

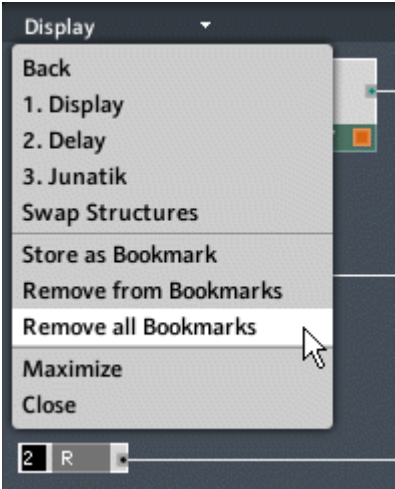



Fig. 3.35 To delete all Bookmarks, use the Remove all Bookmarks menu entry from the Navigation menu.

 Note that adding and removing Bookmarks are actions which cannot be undone using the Undo command. Be especially careful when using the *Remove all Bookmarks* menu entry.

3.3 Operating Ensembles

Although REAKTOR lets you build your own Instruments and Ensembles, the primary function is still being able to play music on them. This section describes most things you need to know when loading, playing, and saving Instruments and Ensembles.


 Note that all operations that can be done using the **Menu** button can also be done in stand-alone mode using the corresponding application menu (File, Edit, Settings, and so on...).



Fig. 3.36 The Menu button

3.3.1 Loading a KOMPLETE Instrument

In this subsection you will first learn how to load KOMPLETE Instruments using the Browser. If you have read the Getting Started Manual, and know how to open KOMPLETE Instruments, you can skip this subsection.

1. Open the Browser as shown in subsection [↑3.1.1, Sidepane](#).
2. Press the **Player** button to browse for installed KOMPLETE Instruments (see screenshot below).



3. In the screenshot above you see that The Finger and REAKTOR Spark KOMplete Instruments have been installed. Click on a folder in the folder tree to display the KOMplete Instrument files (with the extension *.rkplr) it contains in the file list below.
4. Double-click on a *.rkplr file in the file list to load it. Note that in order to use the Full versions of KOMplete Instruments, they first need to be activated using the Service Center. Please refer to the KOMplete Instrument's documentation for more

information on how to do this. Depending on the activation status of a KOMLETE Instrument it will show up with a different icon in the file list (see the two figures below).

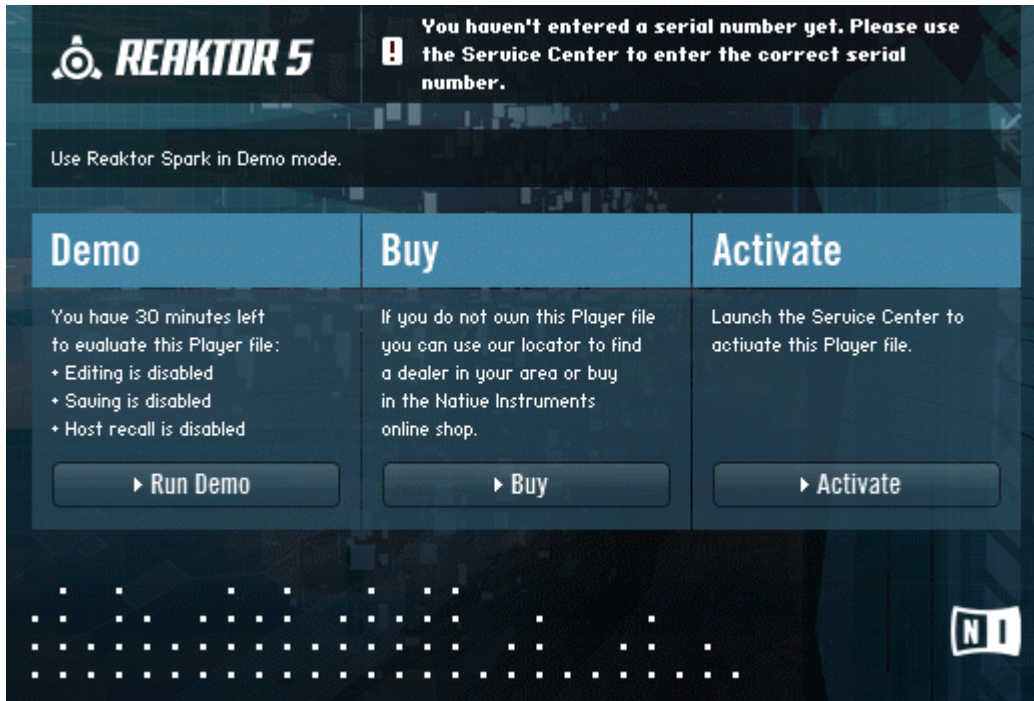
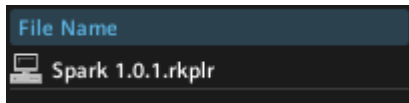
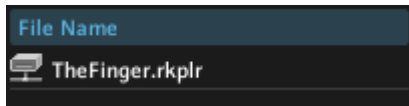


Fig. 3.2 If you load a KOMLETE Instrument that has not been activated using the Service Center, you will see this dialog window.

3.3.2 Opening an Ensemble

The following subsection discusses opening conventional Ensemble files and importing Instruments as Ensembles.

- To open an Ensemble, click on the **Menu** button and choose the *File > Open Ensemble...* menu entry. This opens up a browser dialog box with which you can locate an Ensemble (in the form of a *.ens file) on your computer's disks. Alternatively you can use the keyboard shortcut **Ctrl + O** for Windows (**Cmd + O** for Mac OS X).

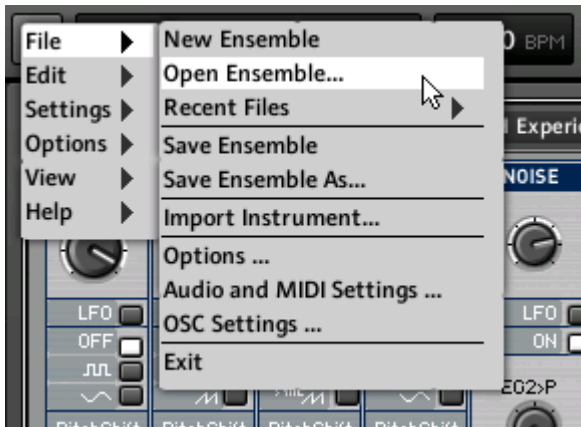
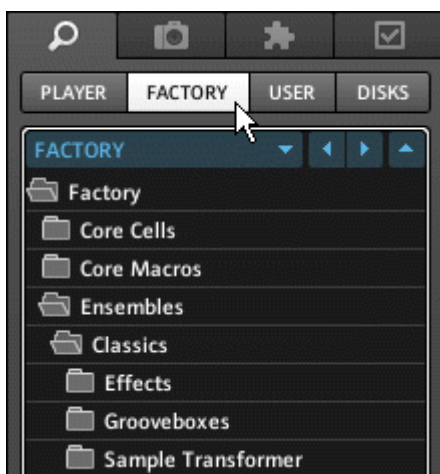


Fig. 3.37 Use the Open Ensemble... menu entry to open an existing Ensemble.

Opening an Ensemble from the Browser

The Browser conveniently lets you load REAKTOR content, in particular, Ensembles and KOMplete Instruments. To load an Ensemble from the Browser, perform the following steps:

1. Open the Browser as shown in subsection [↑3.1.1, Sidepane](#).
2. Use the different buttons to browser different locations for content. For example, to browse the Factory Library, press the **Factory** button (see figure below).



3. Use the folder tree in the top part of the Browser to choose the type of Ensemble you wish to load.
4. For each folder, the list of Ensembles (or REAKTOR files, in general) that it contains is displayed in the bottom part of the Browser. The folder tree and files list are shown below. Double-click on a file load it in the REAKTOR application.

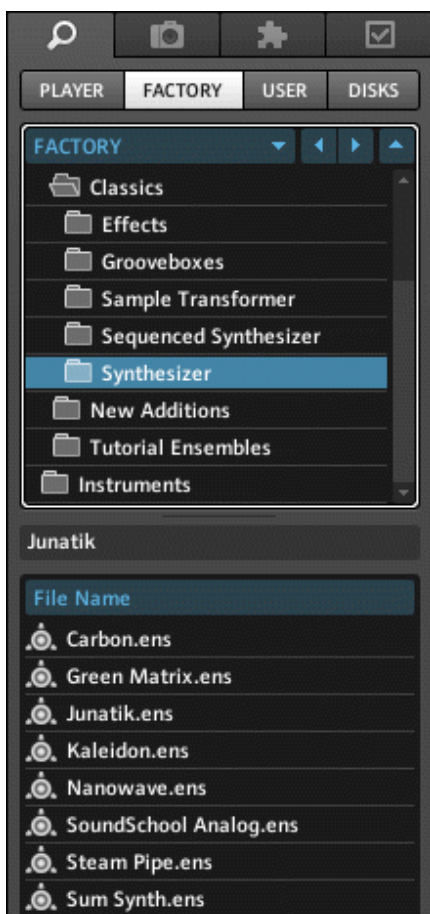


Fig. 3.38 Double-click on an Ensemble file to load that Ensemble in the REAKTOR application.

Opening Recently Loaded Ensembles

REAKTOR remembers the list of Ensembles that you have recently loaded. Using this list is a handy way of quickly loading Ensembles you are working on over a period of time.

- To load a recently loaded Ensemble, click the Menu button and then choose the corresponding file from the *File > Recent Files* submenu.



Fig. 3.39 The Menu button

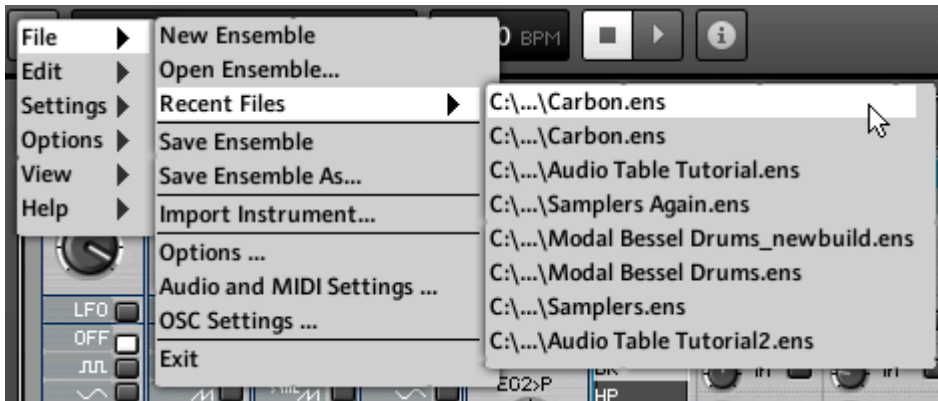


Fig. 3.40 The list of recently loaded Ensembles can be accessed using the Menu button and navigating to the File > Recent Files submenu.

Import Instrument as Ensemble

REAKTOR allows you to import Instruments as Ensembles. This means that all the additional global settings which are saved with an Ensemble are left in the current state but the top-level Instrument becomes the imported Instrument.

- To import an Instrument as an Ensemble, press the Menu button and choose the *File > Import Instrument...* menu entry. This opens up a browser dialog window with which you can navigate to an Instrument (in the form of an *.ism file) that you wish to load as an Ensemble. Alternatively you can use the Browser to navigate to an Instrument and then double-click the corresponding entry in the Files View (similar to the approach described under “Opening an Ensemble from the Browser”).



Fig. 3.41 The Menu button

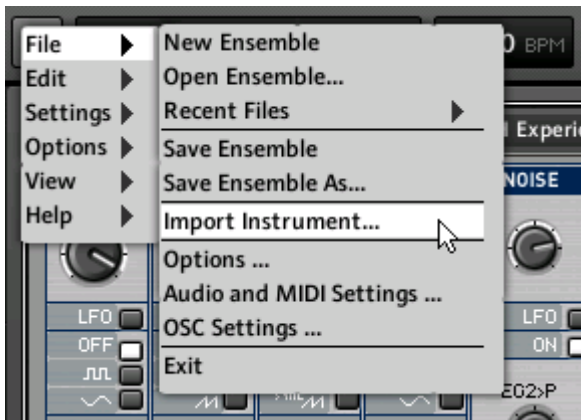


Fig. 3.42 Importing an Instrument as Ensemble

3.3.3 Playing an Ensemble

After having loaded an Ensemble (see subsection [↑3.3.1, Loading a KOMPLETE Instrument](#)) you are ready to play! This subsection discusses how to play and interact with your Ensembles without delving into their Structure. Note that Ensembles can be synthesizers, sequencers, sound generators, as well as effects. In some of these cases it is not always possible to “play notes” on the Ensemble. When talking about synthesizer type Ensembles, there are four ways you can get sound out of them:

1. Use your computer keyboard to play notes which are sent to the synthesizer. The letters and numbers on your keyboard represent MIDI notes. Pressing Q, for example, causes the MIDI note “60”, corresponding to middle C, to be played by the synthesizer.
2. Use a MIDI keyboard or MIDI controller to play notes which are sent to the synthesizer. Please refer to the Getting Started Manual to learn how to configure REAKTOR's MIDI settings.
3. Use a sequencer Instrument within the Ensemble to control a synthesizer that is also within the same Ensemble.
4. Use a host sequencer (such as Cubase, Live, Fruity Loops, or Reaper) to control a synthesizer in REAKTOR.



Holding down the Shift key while playing notes on your computer keyboard raises all incoming MIDI notes by two octaves. Holding down Ctrl+Shift for Windows (Cmd+Shift for Mac OS X) lowers all incoming MIDI notes by two octaves.



Note that in order for your Ensembles to function, REAKTOR's audio engine needs to be turned on. An active audio engine is signaled by the white Audio Engine button in the Main Toolbar. If an Ensemble requires more CPU resources than are available, a dialog window pops up to inform you of this. Thereafter the audio engine is automatically turned off, as can be seen by the red Audio Engine button. Press that button to reactivate the audio engine. Please refer to section [↑14.1, CPU, Debugging, and Optimizing](#) for more information on monitoring CPU load.



Fig. 3.43 Audio Engine button (on)



Fig. 3.44 Audio Engine button (off)

In addition to being able to play notes on them, the main feature of synthesizers and more generally, Ensembles, is that you can tweak the sound via the Instrument Panel, shown in the figure below. Some of the Panel elements with which you can interact and which help you tweak your sounds are knobs, faders, buttons, Switches, menus, and custom interfaces such as XY pads or step sequencers.

- To move a knob or fader, click on the desired control and move your mouse up or down. You can also move a selected control using the up and down arrow keys on your computer keyboard. Use the PgUp and PgDn keys to move the control up and down, respectively, in increments of 10 steps. Press the keys Home and End to move the control to the lower and upper bounds of its range, respectively.



Fig. 3.45 The Panel View is from where you interact with the user interface of your Instrument.

! Note that in order for effect Ensembles to work output sound, they need an input signal. This usually can be achieved by loading REAKTOR as a plug-in in a host sequencer or by configuring the Audio settings (see Getting Started Manual) such that REAKTOR receives a signal from an external source.

Recalling Snapshots

The settings of an Ensemble's Panel elements can be saved as Snapshots (also called patches, programs, or presets). These Snapshots can be recalled using the Snapshot menu in the Main Toolbar (see figure below). The Snapshot menu shows the Snapshots of the Instrument that has been selected as Snapshot Master. Please refer to chapter [↑4, Snapshots](#) for more information on Snapshots.

- To recall a Snapshot, click on the **Snapshot** menu and choose the desired Snapshot from the drop-down menu. Use the up and down arrow buttons to the right of the menu to move up and down in the list of Snapshots.

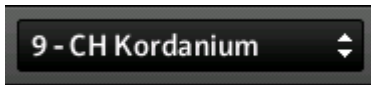


Fig. 3.46 Use the Snapshot drop-down menu in the Main Toolbar to recall Snapshots.

- You can also recall Snapshots for individual Instruments. Just click on the Snapshot area in the Instrument Header (see figure below) to access the list of Snapshots for that Instrument and click on the desired entry. With the Instrument Header selected, you can use the Up and Down arrows on your computer keyboard to browse through Snapshots.

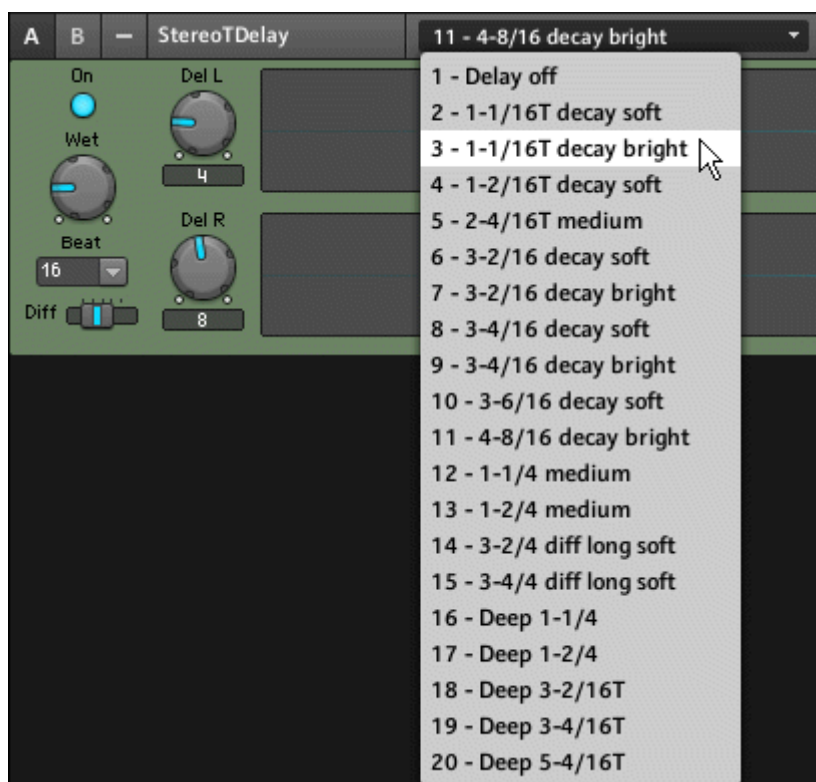


Fig. 3.47 You can recall Snapshots for individual Instruments via the Snapshot area in the Instrument Header.

MIDI Learn

The “MIDI Learn” feature allows you to quickly assign controls on your Instrument Panel to MIDI controllers.

1. Make sure that the audio engine is active (signaled by a white [Audio Engine](#) button).
2. Press the [MIDI Learn](#) button in the Side Bar.
3. Select the control that you want to assign to a MIDI controller.
4. Move your MIDI controller and see how the MIDI assignment has been picked up by the REAKTOR Instrument's control.



Fig. 3.48 The MIDI Learn button



You can also use the “MIDI Learn” feature on a control by right-clicking the control in the Instrument Panel and choosing the *MIDI Learn* menu entry. Alternatively you can press the Menu button and choose the *MIDI Learn* menu entry from the *Options* submenu. In stand-alone mode the Options menu is also an application menu.



Note that in order for your Instrument to receive MIDI signals from a particular source (like a MIDI controller), the Instrument's MIDI Channel for incoming MIDI signals needs to be set to match the MIDI Channel of your MIDI controller. Please refer to subsection [↑13.2.1, MIDI In](#) to learn how to set the incoming MIDI Channel of your Instrument.

Run/Pause Clock

Some Instruments like sequencers need a MIDI Clock to operate. To toggle between a running or paused internal MIDI Clock, use the Run button in the Main Toolbar. You can see whether the MIDI Clock is running from the state of the button. In the below figure, the Run button is depressed which means that the MIDI Clock is running.



A convenient way to toggle between a running and stopped MIDI Clock is with the use of the Space bar on your computer keyboard. You can also access this command by pressing the Menu button and choosing the *Options > Run Clock* menu entry. In stand-alone mode this menu entry is found in the *Options* application menu.

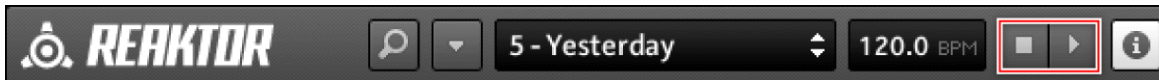


Fig. 3.49 The MIDI Clock area of the Main Toolbar lets you run, pause, and stop the internal MIDI Clock.

Stopping the Clock

To stop the MIDI Clock, press the Stop button in the Main Toolbar. This stops the MIDI Clock but also resets the playing position.

Set BPM

REAKTOR allows you to manually set the BPM (tempo in beats per minutes) of its internal MIDI Clock. To do this, just click and drag the BPM edit field up or down to increase or decrease the BPM, respectively. You can also double-click the BPM edit field and directly enter a number.

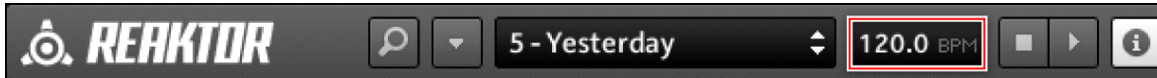


Fig. 3.50 Use the BPM edit field to change the BPM of the MIDI Clock.

Set BPM by Snapshot

To let REAKTOR extract the BPM information from each Snapshot, press the Menu button and choose the *Settings > Set BPM by Snapshot* menu entry.



All menu entries which are accessible via the Menu button are also accessible in stand-alone mode over the application menu.



Fig. 3.51 The Menu button

3.3.4 Saving an Ensemble

- To save your Ensemble, press the Menu button and choose the *File > Save Ensemble* menu entry (see below). Alternatively you can use the keyboard shortcut Ctrl+S for Windows (Cmd+S for Mac OS X). The Ensemble will be saved as a *.ens file.



Fig. 3.52 The Menu button



If you have modified your Ensemble in the Structure and save the Ensemble, the old version of the Ensemble will be overwritten! If you want save the new version of the Ensemble as a different file, use the Save As... command described below.

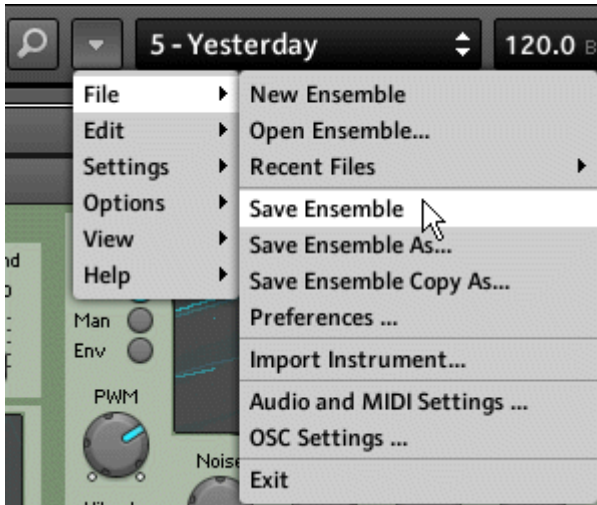


Fig. 3.53 Use the File > Save Ensemble menu entry to save your Ensemble.

Save As...

To save your Ensemble as a new *.ens file, press the Menu button and choose the *File > Save Ensemble As...* menu entry. Alternatively you can use the keyboard shortcut Ctrl+Shift+S for Windows (Cmd+Shift+S for Mac OS X).



Fig. 3.54 The Menu button

3.3.5 Creating a New Ensemble

- To create a new Ensemble, press the Menu button and choose the *File > New Ensemble* menu entry (shown in the figure below). Alternatively you can use the keyboard shortcut Ctrl+N for Windows (Cmd+N for Mac OS X). This creates a new Ensemble containing an empty Instrument.



Fig. 3.55 The Menu button

! If you have an Ensemble open that you have modified, then before creating a new Ensemble REAKTOR will ask you if you would like to save your modified Ensemble. If you choose “No”, then the changes you made to that Ensemble will be discarded.



Fig. 3.56 Use the File > New Ensemble menu entry to create a new empty Ensemble.

3.4 Navigating with the Browser

The Browser enables you to quickly access files relevant to REAKTOR. These files, among others, include Ensemble, Instrument, Macro, Core Cell, Core Macro, and Audio files.

- To access the Browser first open the Sidepane (see subsection [↑3.1.1, Sidepane](#)) and then press the Browser tab. You can also use the shortcut key F5 to show and hide the Browser.



Fig. 3.57 The Browser tab

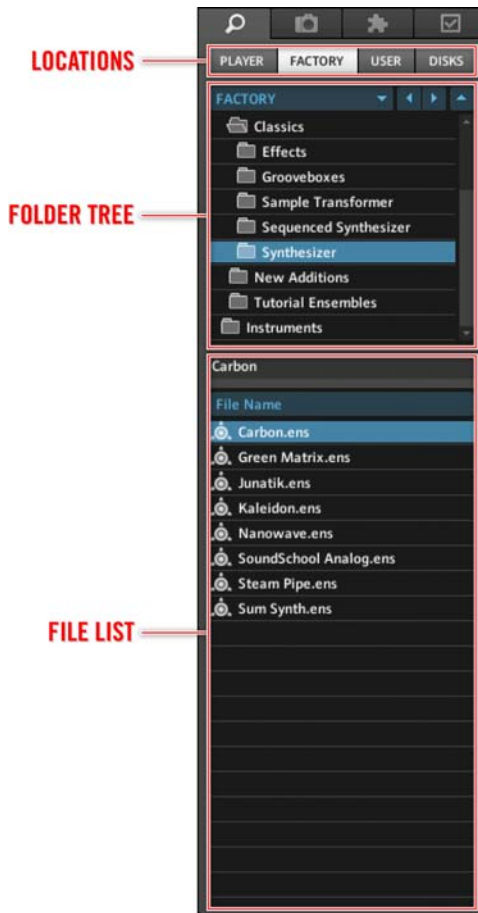


Fig. 3.3 The Browser consists of the Location area, the folder tree, and the file list.

Navigating to Common Locations



Fig. 3.4 The Player, Factory, User, and Disks buttons let you quickly navigate to the folders that hold your REAKTOR content.

- To browse for “powered by REAKTOR” products that have been installed on your system, press the **Player** button, shown above.
- To access Factory Content of REAKTOR, press the **Factory** button, shown above.
- To browse for User Content, press the **User** button, shown above.
- To browse your computer’s disks, press the **Disks** button, shown above.
- To navigate to Audio Files folder, click on the **Favorites** drop-down menu and choose the *Audio Files* menu entry as shown below.

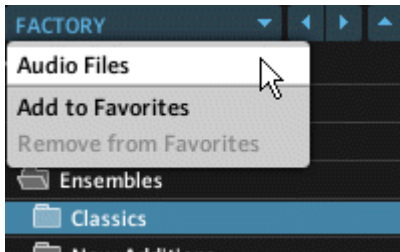


Fig. 3.5 Use the Audio Files menu entry to navigate to the location where you told REAKTOR to look for audio files.

! You need to have specified the paths of the Factory and User Content, and Audio Files directories in the Preferences dialog window. Please see subsection [↑14.4.2, Directories Tab \(Setting File Paths\)](#) to learn how to do this.

Basic Folder Navigation

- To display the files contained within a folder in the file list, click on the folder in the folder tree.
- To expand a folder to see which folders it contains, either double-click on the folder name or single-click on the corresponding folder’s icon. The same procedure applies when you wish to collapse a folder.

Use the following buttons to speed up navigation:



Fig. 3.6 The Back, Forward, and Up buttons accommodate basic navigation.

- To navigate to the previous folder, press the **Back** button (left arrow) shown above.
- To reverse the action of the **Back** button, press the **Forward** button (right arrow) shown above.

- To navigate one level upwards in the folder tree, press the Up button (up arrow) shown above.

Adding, Removing, and Navigating to Favorites

With the “Favorites” feature, REAKTOR allows you to store and recall often used locations.

- To add the location of the selected folder as a menu entry in the Favorites drop-down menu click on the *Add to Favorites* menu entry in the Favorites drop-down menu, as shown in the figure below.

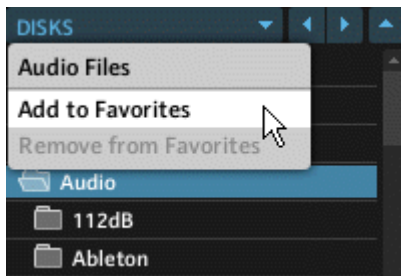


Fig. 3.7 Use the Add to Favorites menu entry to store a location on your computer disks as a menu entry in the Favorites drop-down menu.

- To navigate to a location that has previously been marked as a “Favorite”, choose the location’s corresponding menu entry from the Favorites drop-down menu, as shown below.

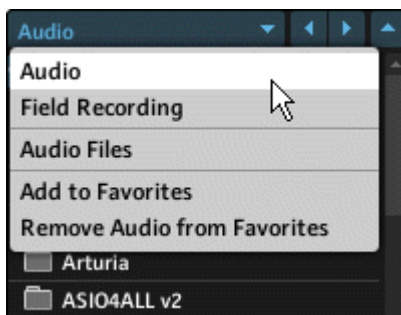


Fig. 3.8 Recalling the “Audio” folder using the corresponding Audio menu entry from the Favorites drop-down menu. Note that there are two “Favorites”: the “Audio” and the “Field Recording” folders.

- To remove a location from the “Favorites”, first navigate to that location (preferably using the corresponding menu entry) and then choosing the *Remove *Location* from Favorites* menu entry, as shown below.

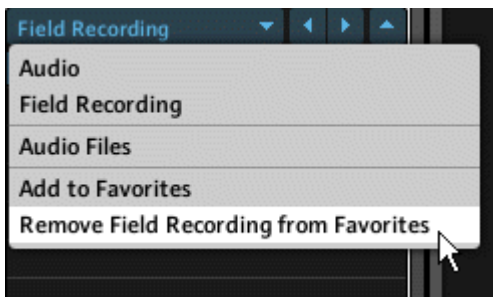


Fig. 3.9 Use the Remove *Location* from Favorites menu entry to remove an entry from the Favorites drop-down menu.

3.5 Other Common Operations

Turning Hints On/Off

- Most REAKTOR Objects (such as Instruments, Macros, Modules, and Core Cells) allow the user to write an info text in the Info page of the Object's Properties, describing the Object or its use in the Structure. Modules already have info texts written in that space describing their general use and for Modules this text cannot be edited. A handy feature in REAKTOR is that Info texts can be displayed by fly-out Hints (also known as Tooltips) when hovering over the respective Object with the mouse (both on the Instrument Panel and in the Structure).
- To toggle the Hints feature on and off, press the **Hints** button in the Main Toolbar.



Fig. 3.58 The Hints button

Undo/Redo

The “Undo” feature of REAKTOR allows you to reverse the effect of a set number of last editing operations carried out in any of the Structures. The “Undo” feature does not apply to Panel control setting changes, that is, changing the value of a knob or fader. For this, use the “Compare” feature described in section [4.4, Comparing Snapshots](#).

- To undo the last Structure editing operation, press the **Menu** button and choose the *Edit > Undo *operation** menu entry (see figure below) or simply press the keyboard shortcut Ctrl+Z for Windows (Cmd+Z for Mac OS X).

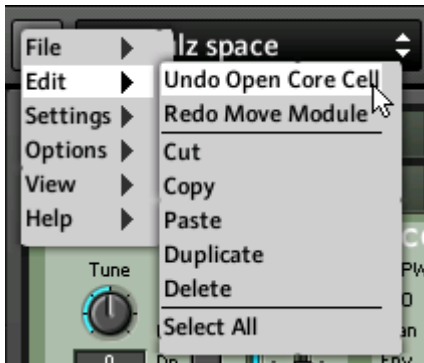


Fig. 3.59 Use the Edit > Undo *operation* menu entry to undo the last Structure editing operation.

- To redo the last operation for which you applied the Undo command, press the **Menu** button and choose the *Edit > Redo *operation** menu entry or simply press the keyboard shortcut Ctrl+Y for Windows (Cmd+Y for Mac OS X).



Fig. 3.60 The Menu button

Setting the Number of Undos

Sometimes a large number of available Undo commands are useful. However, since the “Undo” feature uses the computer memory, saving a large number of Undos can be undesirable from a resource-management point of view. To set the number of Undo commands REAKTOR saves, perform the following steps:

1. Press the **Menu** button.

2. Choose the *File > Options* menu entry. This opens the REAKTOR Preferences dialog window.
3. Click on the *Options* tab.
4. Now enter the number of desired Undo commands in the *Number of Undos* edit field.



Fig. 3.61 Edit field for the number of Undos

Audio and MIDI Information

Information regarding the output audio level of REAKTOR and the activity of incoming and outgoing MIDI messages are displayed in the Main Toolbar area (see figure below). You can conveniently control the level of the outgoing audio signal by using the horizontal slider below the Audio Level Meter. The MIDI Activity display consists of two meters. The left and right meters show the activity and approximate values of incoming and outgoing MIDI signals, respectively.

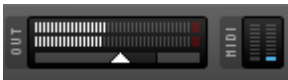


Fig. 3.62 The Audio Level Meter shows the level of the outgoing audio signal which you can control using the horizontal slider. The MIDI Activity display shows activity of incoming and outgoing MIDI signals.

You can also monitor the MIDI activity for individual Instruments in the Instrument header, shown below. As above, the MIDI Activity display in the Instrument header consists of two meters. The left and right meters show the activity and approximate values of incoming and outgoing MIDI signals, respectively.

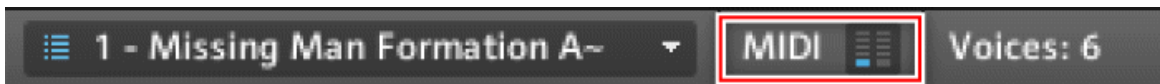


Fig. 3.10 The MIDI Activity display in the Instrument header helps you monitor MIDI activity for individual Instruments.

Turning the Audio Engine On/Off



Fig. 3.63 The Audio Engine button (turned on)

- The audio engine requires CPU resources. Especially when running REAKTOR in plug-in mode, you might want to turn REAKTOR's audio engine off for while. To do this, just press the Audio Engine button shown in the figure above.
- To activate the audio engine, press the Audio Engine button again.

! If an Ensemble requires more CPU resources than are available, a dialog window pops up to inform you of this. Thereafter the audio engine is automatically turned off, as can be seen by the red Audio Engine button. Press that button to reactivate the audio engine. Please refer to section [↑14.1, CPU, Debugging, and Optimizing](#) for more information on monitoring CPU load.



Fig. 3.64 The CPU area holds the Audio Engine button, the CPU Meter, and the Sample Rate drop-down menu.

Changing the Sample Rate

The Sample Rate drop-down menu sets the Sample Rate at which REAKTOR generates and processes audio signals. With higher Sample Rates you can achieve better sound quality, but the CPU load increases proportionally. You can change the internal Sample Rate to any of the values in the Sample Rate drop-down menu. The range of available values depends on your soundcard or host sequencer (when running REAKTOR as a plug-in). If the internal Sample Rate is different from the soundcard's or host sequencer's Sample Rate, the Audio In and Audio Out Modules will do the necessary Sample Rate conversion.

- To change the internal Sample Rate of REAKTOR, click on the Sample Rate drop-down menu (see figure below) and choose the menu entry corresponding to the desired Sample Rate. Alternatively you can press the Menu button and choose the Sample Rate from the *Settings > Sample Rate* submenu.

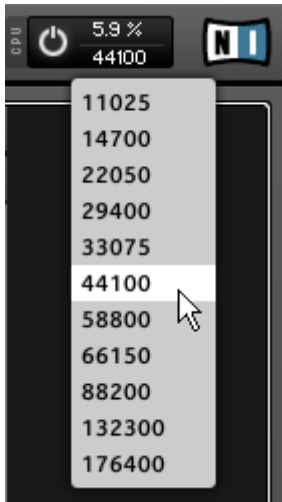



Fig. 3.65 The Sample Rate drop-down menu offers you a list of possible internal Sample Rates for REAKTOR.

 The internal Sample Rate is a setting that is saved along with your Ensemble.

Changing the Control Rate

Some Modules in REAKTOR (like the LFO Module) send their output Event signals at the Control Rate. Event processing requires CPU resources and the rate of Event Processing often depends on the Control Rate. Therefore adjusting the Control Rate not only affects the quality of your LFO's and other Modules' output signal, but also determines how much CPU resources are used to process the Events resulting from those Event streams.

 Please refer to section [↑9.2, Event Signals](#) for more information on Event signals.

- To change the Control Rate of your Ensemble, press the **Menu** button and choose the menu entry corresponding to the desired Control Rate from the *Settings > Control Rate* submenu (see below).



Fig. 3.66 Use the Settings > Control Rate submenu to choose the desired Control Rate.

! The Control Rate is a setting that is saved along with your Ensemble.

External Sync

The External Sync option toggles between the internal REAKTOR clock and an external clock (received via MIDI) for all Sync Clock and 1/96 Clock Modules. It also enables control of Start/Stop Modules by external MIDI Start/Stop messages. When the “External Sync” feature is turned on, the tempo cannot be adjusted by the BPM edit field in the Main Toolbar; instead, the internal clock is adjusted according to the external clock.

- To activate the “External Sync” feature, click on the Menu button and choose the *Settings > External Sync* menu entry.

MIDI Clock Out

If you enable the “MIDI Clock Out” feature, REAKTOR sends out MIDI Clock Ticks on all active MIDI out ports (as set in the Audio and MIDI Setup dialog window).

- To activate the “MIDI Clock Out” feature, click on the Menu button and choose the *Settings > Send Clock to MIDI Output* menu entry.

REAKTOR 4 Legacy Mode

Although this mode is not a common preference by any means, it is listed in this section for the sake of completeness. By enabling the REAKTOR 4 Legacy mode, Event Initialization behavior (see subsection [↑9.2.5, Initialization](#)) is set to the way it was implemented in REAKTOR 4. Some very old Ensembles that were built in REAKTOR 4 only function correctly when this option is activated.

- To activate the “Reaktor 4 Legacy Mode” feature, click on the **Menu** button and choose the *Settings > Reaktor 4 Legacy Mode* menu entry.

Accessing Product Information

- To access product information, either press the **Native Instruments** or **REAKTOR** logo in the Main Toolbar.



Fig. 3.67 The Native Instruments logo from the Main Toolbar



Fig. 3.68 The REAKTOR logo from the Main Toolbar

3.6 REAKTOR as a Plug-In

A common use case for REAKTOR is loading it as a plug-in in a host application. This section will explain the “Autosave” feature and describe preferences strictly tied to the REAKTOR plug-in.

! The OSC protocol is disabled when REAKTOR is running as a plug-in.

3.6.1 The Autosave Feature

Usually, when using an Ensemble (in the REAKTOR plug-in) in a project, you make changes to the Ensemble by tweaking its knobs and so on. Some changes to the Ensemble such as moving controls on the Instrument Panel are saved by the host application when

saving a project with that application. This means that when you load that project again in the host application, these changes are recalled by REAKTOR. Other changes, such as editing data in Audio or Event Tables or modifying an Ensemble's Structure are not saved with the host's project file.

The Change Alert Dialog Window

When an Ensemble that has been loaded into the REAKTOR plug-in has been modified in such way that the changes are not saved by the host application, you will be alerted with the dialog window shown below. To keep from losing the changes you have made to your Ensemble, you should save a local copy of the Ensemble along with these changes.

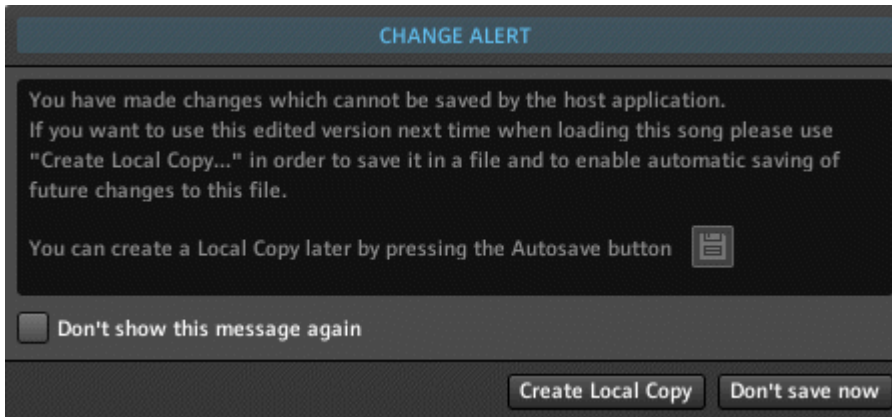


Fig. 3.11 The Change Alert dialog window

of the local copy of the Ensemble.

- In the Change Alert dialog window shown above, you can also create a local copy by clicking on the **Create Local Copy** button. This opens up your operating system browser which lets you choose a name and location for the local copy of your Ensemble. After choosing and confirming a name and location for the local copy, you will see the dialog window shown in the figure below, telling you that the „Autosave“ feature has been enabled.

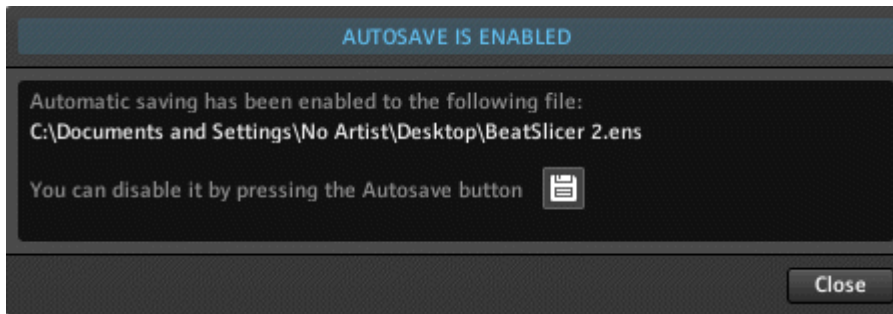


Fig. 3.12 The Autosave Enabled dialog window tells you that the “Autosave” feature has been enabled and give you the location

1. The Change Alert dialog window also gives you the option not to save a local copy of the Ensemble. Press the *Don't save now* button to skip creating a local copy of your Ensemble. All changes not saved by the host application are lost when you close the project (unless you change your mind and decide to save the Ensemble or a local copy of it before closing).
2. If there are changes that will be lost when you close the host project, the *Auto-Save* button in the Sidebar will be red, as shown below.



3. If you change your mind and wish to save a local copy of the Ensemble, you can press the red *Auto-Save* button. This opens up your operating system browser which lets you choose a name and location for the local copy of your Ensemble. After choosing and confirming a name and location for the local copy, you will see the dialog window shown in the figure below, telling you that the „Autosave“ feature has been enabled.

The Autosave Button

As described under the previous heading, when the “Autosave” feature has been disabled the red *Autosave* button is used to enable the “Autosave” feature. However, when the “Autosave” feature has been enabled, then the *Autosave* button appears in the Sidebar like shown in the figure below.



Fig. 3.13 With the “Autosave” feature enabled, the Autosave button appears as shown here.



You can also disable or change the “Autosave” feature by pressing the **Menu** button and choosing the *File > Change/Disable Local Copy* menu entry. Enabling the “Autosave” feature can be done by pressing the **Menu** button and choosing the *File > Create Local Copy* menu entry.

Disabling the Change Alert Dialog Window

1. If you don’t wish to be confronted by the Change Alert Dialog window, you can keep it from appearing when critical changes are made to an Ensemble. To do this, go to the Preferences by pressing the **Menu** button and choosing the *File > Preferences...* menu entry.
2. Press the **Options** tab.
3. To keep the Change Alert dialog window from appearing, disengage the **Show edited warning** checkbox, shown below.



4. Press the **Close** button to exit the Preferences. Now your only source of information regarding the state of the “Autosave” feature is the **Autosave** button. A red **Autosave** button means the the “Autosave” feature is disabled, a white **Autosave** button means that the “Autosave” feature has been enabled.

3.6.2 Window Sizes in the REAKTOR Plug-In

When REAKTOR is running as a plug-in, you can define and choose two fix window sizes: Large and Small.

Choosing a Window Size

- To choose any of these two sizes, press the **Menu** button and choose the *View > Small Size* or *View > Large Size* menu entries for the Small and Large Sizes, respectively. The *View* submenu when REAKTOR is running as a plug-in is shown below.
- To minimize the REAKTOR plug-in window to match that of the Main Toolbar, press the **Menu** button and choose the *View > Minimize* menu entry (shown below)

- To use an optimal window size as determined by the area of the Structure or the Instrument Panel on the screen (depending on what is being displayed at any given time), press the **Menu** button and choose the **View > Optimal Size** menu entry (shown below). Note that whenever you navigate to a location in the Structure or change the area occupied by the Instrument Panel, you need to click on this menu entry again in order to have REAKTOR resize its plug-in window to the new optimal size.

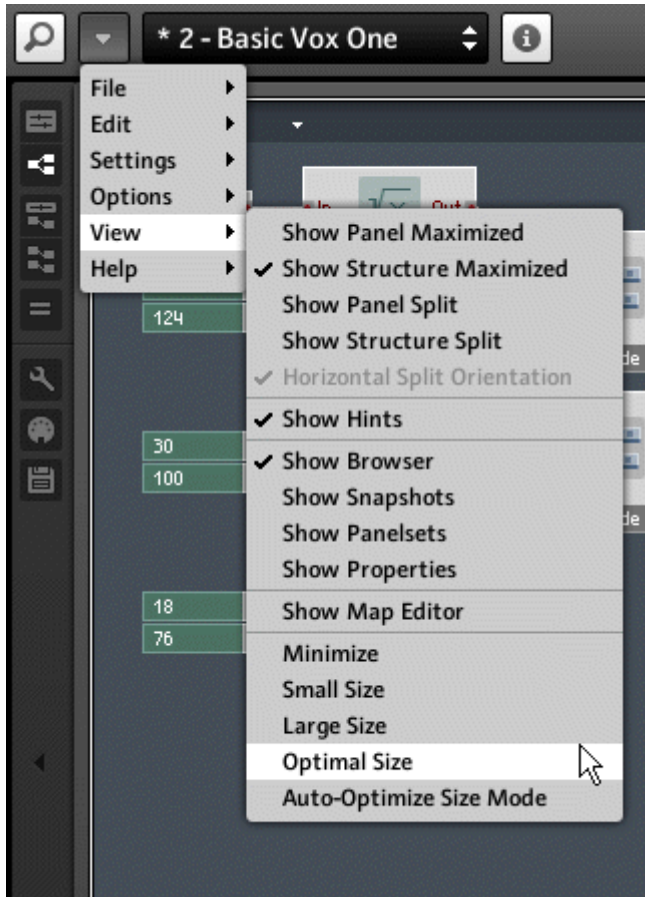


Fig. 3.14 When REAKTOR is running in plug-in mode, the View submenu lets you choose different sizes for the plug-in window.

Defining Window Sizes in the Preferences

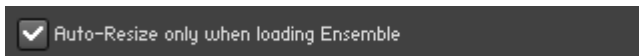
1. As mentioned above, you can define the dimensions for the fixed Small and Large window sizes. Additionally, you can choose beyond which size the REAKTOR plug-in window will not grow when operating with the Optimal Size. To set these values, open the Preferences by pressing the **Menu** button and choosing the *File > Preferences...* menu entry.
2. Press the **Options** tab to navigate to the edit fields relevant to window sizes. You should see them as shown in the screenshot below.



The screenshot shows the 'Options' tab of the REAKTOR Preferences dialog. It contains six input fields arranged in a 3x2 grid. The first row is for 'Small' window sizes, the second for 'Big' window sizes, and the third for 'Max. Opt.' (Optimal Size) window sizes. Each field has a numerical value entered.

Small Window Width	Small Window Height
565	410
Big Window Width	Big Window Height
1200	745
Max. Opt. Window Width	Max. Opt. Window Height
1200	750

3. To define a window width and height for the Small window size, enter the desired values into the **Small Window Width** and **Small Window Height** edit fields, respectively.
4. To define a window width and height for the Large window size, enter the desired values into the **Big Window Width** and **Big Window Height** edit fields, respectively.
5. To define the window width and height beyond which the window will not grow in the Optimal Size mode, enter the desired values into the **Max. Opt. Window Width** and **Max. Opt. Window Height** edit fields, respectively.
6. To have REAKTOR only resize its window when loading a new Ensemble, engage the **Auto-Resize only when loading Ensemble** checkbox, shown below.



The screenshot shows a single checkbox labeled 'Auto-Resize only when loading Ensemble'. The checkbox is checked, indicated by a small square icon to the left of the text.

7. Press the **Close** button to close the Preferences.

4 Snapshots

Snapshots (also known as patches, programs, or presets) enable you to store and recall an Instrument's sounds. When you create a Snapshot, the current settings of all the Instrument's Panel controls and MIDI controllers are stored in the Snapshot. Among the Panel controls, settings like the knob and fader positions, list box and switch settings, and button states are saved. When you recall a Snapshot, all the Instrument's controls are restored to the settings they were in when the Snapshot was originally created. Each REAKTOR Instrument can store 16 Snapshot Banks containing 128 Snapshots each, for a total of 2048 Snapshots. In this chapter you will learn how to recall and create Snapshots in addition to working with Snapshot Banks and features like “Snapshot Compare” and “Snapshot Randomize”. All of these features are accessed from the Snapshot tab (see below). Please refer to subsection [↑3.1.1, Sidepane](#) to learn how to open the Snapshot tab.



Fig. 4.1 The Snapshot tab lets you do all the possible Snapshot operations in REAKTOR.

4.1 Recalling Snapshots

There are five different methods you can use to recall an Instrument's Snapshots:

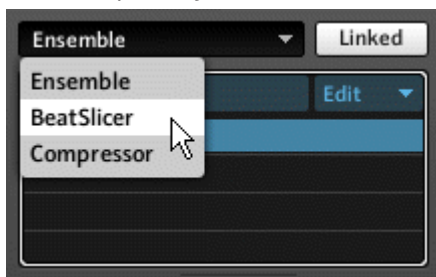
- The first way to recall a Snapshot is by using your mouse to select the desired Snapshot from the Snapshot drop-down menu in the header. Use the up and down arrow buttons to the right of the menu to move up and down in the list of Snapshots. The Instrument for which the Snapshot List is shown must have the Snapshot Master for Plugin checkbox engaged in the Function page of its Properties (see section [↑4.8, Defining Snapshot Behavior for Instruments](#) for more information on this).

1 - Baltik State

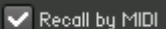
- You can also use your mouse to select Snapshots from the Snapshots drop-down menu in the Instrument Header. With the Instrument header selected, you can use the Up and Down arrow keys on your computer keyboard to select previous and next Snapshots, respectively.

1 - Baltik State

- Another option is to go to the Snapshot tab (see subsection [3.1.1, Sidepane](#) to learn how to navigate there). In the Snapshot tab you first need to select the desired Instrument (or Ensemble) for which the Snapshot recall is to take place. This is done using the Select Instrument drop-down menu shown in the figure below. Then click on the Snapshot you wish to recall in the Snapshot List.



- Additionally, you can select Snapshots by issuing MIDI Program Change messages from a MIDI keyboard (or some other MIDI device). For this to work, the Recall by MIDI checkbox in the Function page (see section [4.8, Defining Snapshot Behavior for Instruments](#)) of the corresponding Instrument's Properties should be engaged. The MIDI Program Change message selects a Snapshot by its number (an integer in the range [0 ... 127]). A Program Change value of “0” selects the first Snapshot, a Program Change value of “1” selects the second Snapshot, and so on.



- Last, if you are a REAKTOR builder, you can use the Snapshot Module to recall, store, randomize, and morph Snapshots. Please read the entry of the Snapshot Module in the Module reference for more information on this Module.

Selecting the Instrument for the Snapshot List

If you have several Instruments in your Ensemble, there will be a Snapshot List for each of these Instruments in the Snapshot tab. This will be the case even if some of these Instruments have no Snapshots at all; that Snapshot List will then be empty. The Snapshot tab can only display one Snapshot List at a time. How do you choose the Instrument for which the corresponding Snapshot List should be displayed? There are two ways:

- You can select the Instrument for which the Snapshots are displayed in the Snapshot List from the **Select Instrument** drop-down menu.
- Another option is to activate the **Linked** button. If this button is activated, the Snapshot List is linked to the Panel View. This means that if you select an Instrument in Panel View, its Snapshot List is automatically displayed in the Snapshot tab. If the **Linked** button is not active, you must use the **Select Instrument** drop-down menu explained above to manually choose which Instrument's Snapshots are displayed in the Snapshot List.



Fig. 4.1 The Linked button

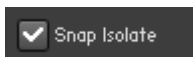
Making a Control Immune to Snapshot Recall

When you select a Snapshot, the Panel controls and MIDI controllers jump to their positions as stored in the Snapshot. In some cases you might not want this to happen. For example, you might create a sequencer with a BPM (tempo) knob whose settings you want to be independent of the Snapshot. To prevent a Panel control or MIDI controller from jumping to the position designated by the Snapshot, follow these instructions:

1. Double-click on the desired control like a knob, fader, list, or button.
2. This should have opened up the control's Properties View. Now click on the **Function** button.



3. Engage the **Snap Isolate** checkbox in the Snapshots area of the control's Function page.



Control IDs

Every REAKTOR Panel control has a unique ID number, as displayed in the ID for Files edit field in the control's Function page. This ID number is called the Control ID, and it is shown in the corresponding Edit field (see below).

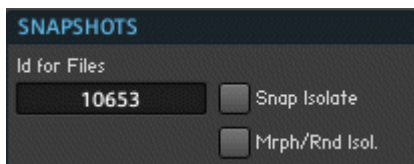


Fig. 4.2 The ID for Files edit field holds the value by which a Snapshot identifies the Panel control.

REAKTOR lets you change these Control IDs. Before you do so, you have to be aware of a few dangers that are tied to this. Snapshots assign specific (control) values to specific Control IDs. For example, Snapshot 1 might assign the value “0.5” to a knob with the Control ID “112” and the value “0.75” to a fader with the Control ID “222”, and so on. If you change the knob and fader Control IDs, it will change the values that Snapshot 1 assigns to these controls. Thus you can break Snapshots which you have worked on so arduously.

! Do not change the Control ID Numbers unless you really have to. Be aware that changing Control ID Numbers breaks the Snapshots which were used for these controls.

4.2 Adding and Removing Snapshots

The basic operations that you can do with the Snapshot List are to add, overwrite, and remove Snapshots. This section describes which buttons you need to press for the different Snapshot List operations. Before pressing these buttons, though, you need to make sure that you are working with the Snapshot List of the intended Instrument. You can see which Instrument is selected with the *Select Instrument* drop-down menu. Also, you need to click the *Append*, *Store*, and *Insert* buttons *twice* for these operations to be finalized. The first click lights up the button and places a blinking cursor in the appended, overwritten, or inserted Snapshot slot, giving you the opportunity to type a name for it. The second click dims the button and saves the appended, stored, or inserted Snapshot.



Fig. 4.3 The Select Instrument drop-down menu

! Remember to click *twice*! If you forget the second click, you might end up doing something very different from what you intended.

Appending Snapshots

If you want save the current Instrument settings as a new Snapshot to the first empty slot in the corresponding Snapshot List, then use the **Append** button in the Snapshot List Operations area. If the current Snapshot Bank is full, pressing the **Append** button causes the Snapshot to be saved to the first empty slot in the next Snapshot Bank. If there are no more empty Snapshot Slots, pressing the **Append** button causes nothing to happen. This would be the case when you have used up all 16 Snapshot Banks, each filled with 128 Snapshots.

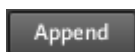


Fig. 4.4 The Append button

Inserting Snapshots

You can also save the current Instrument settings as a new Snapshot that is inserted into the Snapshot List directly after the selected Snapshot. To do this, press the **Insert** button in the Snapshot List Operations area. Note that this can cause Snapshots to move from the current Bank to the next Bank.



Fig. 4.5 The Insert button

Overwriting Snapshots

To replace the selected Snapshot with the current Instrument settings, use the Store button. Note that when you overwrite a Snapshot, you lose its original settings.



Fig. 4.6 The Store button

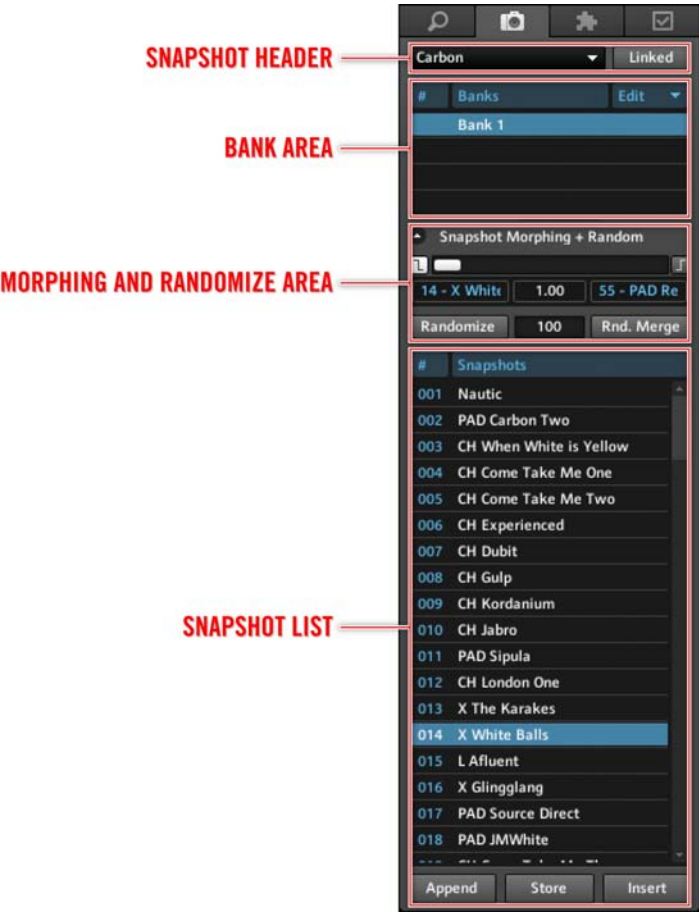


Fig. 4.7 The different areas of the Snapshot tab

Deleting Snapshots

If you want to delete one or more Snapshots from the Snapshot List, select the Snapshots and press choose the *Delete selected Snapshots* menu entry from the *Edit Bank* drop-down menu, as shown in the figure below. Note that deleting Snapshots creates empty slots in the Snapshot List; you can use the *Sort and Compress Snapshots* entry from the *Edit Bank* drop-down menu to remove these gaps (see subsection [↑4.7.3, Renumbering and Renaming Banks](#)).

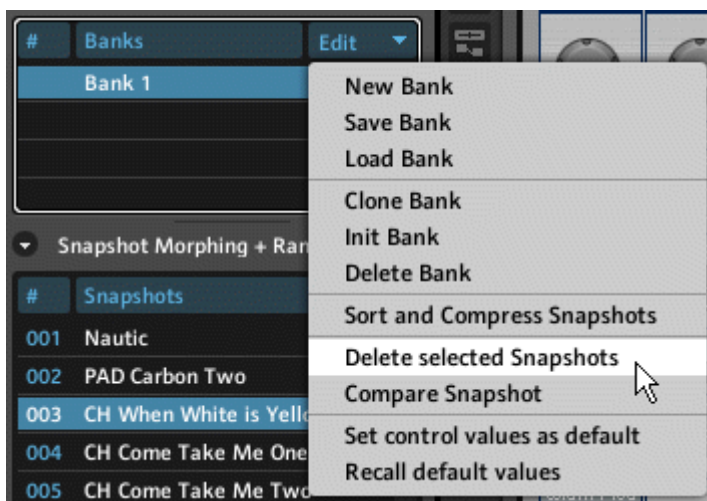


Fig. 4.8 The Delete selected Snapshots menu entry.

Setting Instrument Settings to Default Values

You can change the settings of all Instrument controls (which do not have the Snap Isolate checkbox engaged, see section [↑4.1, Recalling Snapshots](#)) to their default values. The Default value for a control is set with the *Default* edit field in the corresponding Module's Function page. You set the Instrument controls to their Default values by choosing the *Recall default values* menu entry (see figure below) from the *Edit Bank* drop-down menu. This changes the selected Snapshot's settings, but does not save the changed Snapshot. To do this, you must subsequently press the *Store* button.

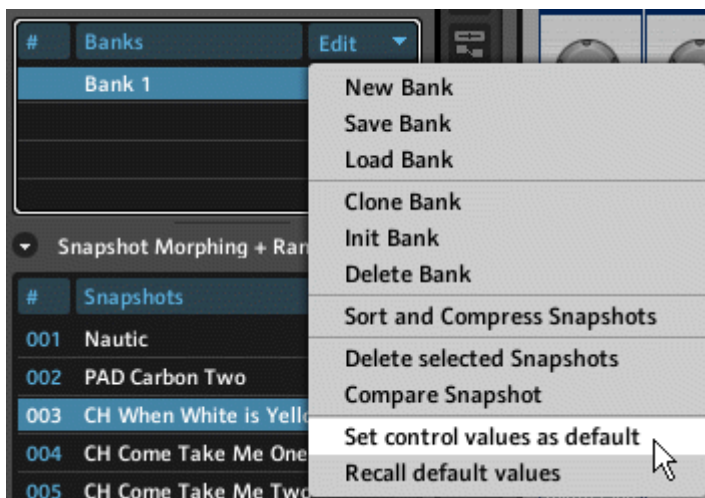


Fig. 4.9 Use the Set control values as default and Recall default values menu entries to set and recall default settings.



Fig. 4.10 The Store button

Setting Control Values as Default

You can acquire the current settings of all of the Instrument's controls as the Default values to be used when the state of a control is to be reset. This is also the case when applying the *Recall default values* menu entry discussed above.

- To set the current control values as the Default values, choose the *Set control values as default* menu entry from the *Edit Bank* drop-down menu,

4.3 Renaming and Copying Snapshots

Since there are no explicit “rename” and “copy” operations in the Snapshot tab, these two common tasks will be explained in this section.

Renaming Snapshots

There are two ways to rename an existing Snapshot in the Snapshot List:

1. Double-click on the Snapshot and type the desired name. Then press the Enter key on your keyboard to save the renamed Snapshot.
2. Alternatively, select the Snapshot and click on the **Store** button. Next you must type in the desired name and then click on the **Store** button once more to save the renamed Snapshot.

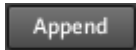


Fig. 4.11 The Store button

Copying Snapshots

As with renaming, there are two ways to copy Snapshots within the Snapshot List of an Instrument:

1. For the first way select the Snapshot, then click on the **Append** button. Next rename the appended Snapshot if desired, and then click on the **Append** button a second time to save the appended Snapshot. Note that this copies that Snapshot to the first empty slot in the Snapshot List.



1. You can also first select the Snapshot, then click on the **Insert** button. Following that, rename the inserted Snapshot if desired, and then click on the **Insert** button once more to save the inserted Snapshot. Note that this copies the Snapshot to a new slot below the originally selected slot in the Snapshot List.



Fig. 4.12 The Insert button

4.4 Comparing Snapshots

REAKTOR makes sound design easy for you by letting you compare a Snapshot with a modified version of the same Snapshot or with another, completely different Snapshot. This section describes how to do these two tasks. The basic idea behind the “Compare”

feature is simple. The modified (or different) Snapshot is stored in the “compare buffer”, and the *Compare Snapshot* menu entry is used to toggle between the original Snapshot and the modified (or different) Snapshot.

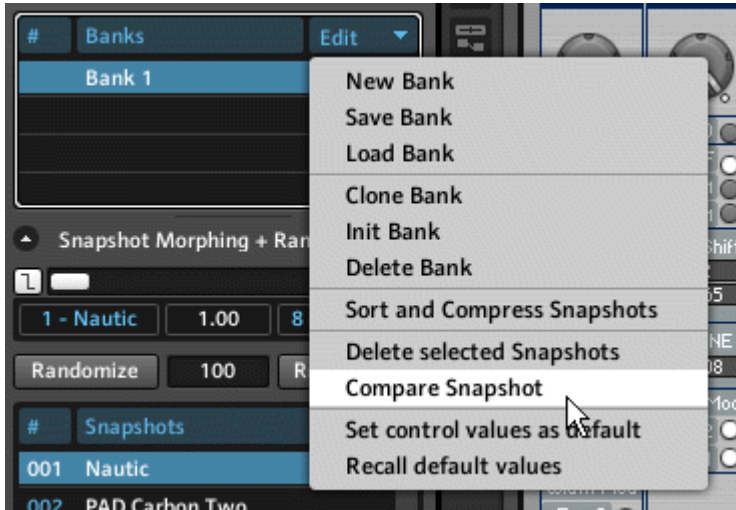


Fig. 4.13 The Compare Snapshot menu entry in the Edit Bank drop-down menu

Comparing an Original Snapshot with a Modified Version

To compare a Snapshot with a modified version of the same Snapshot, follow these instructions:

1. Make sure you have an Instrument open.
2. Open the Snapshot tab.
3. Select a Snapshot in the Snapshot List.
4. Make sure there is no checkmark in front of the *Compare Snapshot* menu entry in the Edit Bank drop-down menu.
5. Modify the settings of the Instrument controls as desired.
6. Use the *Compare Snapshot* menu entry in the Edit Bank drop-down menu to toggle between the original and modified Snapshot versions.
7. If you want to make further modifications, repeat instruction steps 5-6.

Comparing Two Different Snapshots

To compare two different Snapshots in the Snapshot List of an Instrument, follow these instructions:

1. Make sure you have an Instrument open with at least two Snapshots.
2. Open the Snapshot tab.
3. Select a Snapshot in the Snapshot List.
4. Next, select another Snapshot in the Snapshot List. The first Snapshot is now stored in the “compare buffer”.
5. Now you can use the *Compare Snapshot* menu entry in the Edit Bank drop-down menu (shown in the figure above) to toggle between the two Snapshots.



If you are modifying a Snapshot and accidentally select a different Snapshot, you can recover your modifications by choosing the *Compare Snapshot* menu entry right away (before you make any changes to the new Snapshot).

4.5 Morphing Between Snapshots

Not only can you compare two Snapshots, but you can even morph between them, that is, change an Instrument's Panel control settings gradually (over a time interval of [0 ... 60] seconds) from their values in one Snapshot to their values in another Snapshot. The Snapshot Morph area (see below), holds the controls for morphing between Snapshots.

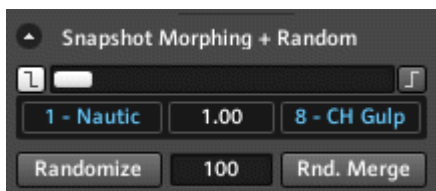


Fig. 4.14 The Snapshot Morph area in the Snapshot tab lets you morph between two Snapshots in the Snapshot List.

To morph between two Snapshots, follow these instructions:

1. Make sure you have an Instrument open with at least two Snapshots.
2. Open the Snapshot tab and if the Snapshot Morph area depicted above is not open, press the arrow button next to the text “Snapshot Morphing and Random”, shown below.



3. Set your desired morphing time in seconds (a value in the interval [0 ... 60]) in the Morph Time edit field. This value determines how long it will take the controls to morph (move) from their current settings to their new settings.



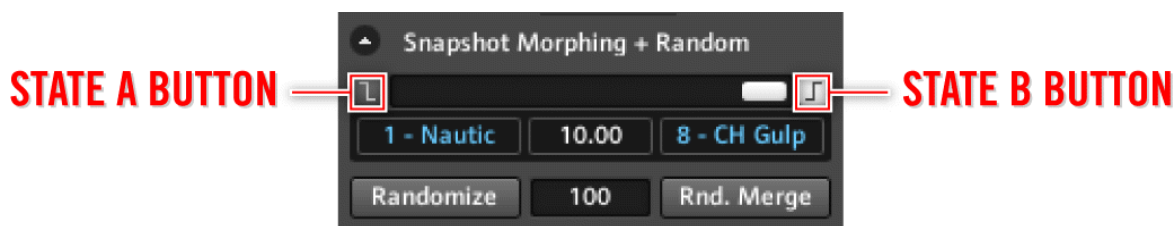
4. Next click on the **Select A** button to activate it (the text will change to “Select...”).



5. Then select the first of the two Snapshots (Snapshot A) between which you want to morph from the Snapshot List.



6. Now click the **Select B** button to activate it (it is the button that lies symmetrically to the **Select A** button to the right).
7. Select the second of the two Snapshots (Snapshot B) from the Snapshot List.
8. Gradual, incremental change between two Snapshots is the basis of morphing. Because button and switch settings cannot be changed gradually, REAKTOR does not let you morph them. Therefore, before you begin to morph, you must decide whether to use the button and switch settings from Snapshot A or Snapshot B. To use the button and switch states from Snapshot A, press the **State A** button to the left of the **Morph** slider (see below). The button should light up. Alternatively, to use the button and switch states from Snapshot B, press the **State B** button to the right of the **Morph** slider (in this case this button should light up).



- Now you are ready to morph between the two Snapshots. Move the horizontal Morph slider (see figure below) to a new position. When the slider is set fully to the left, the Instrument control settings (except perhaps the button states) correspond purely to Snapshot A. Similarly, when the slider is set fully to the right, the control settings correspond to Snapshot B. A slider setting right in the middle corresponds to an arithmetic average between the control settings of the two Snapshots, that is, 50 % of Snapshot A and 50 % of Snapshot B. Other slider settings have analogous results. After moving the Morph slider to a new position, the Instrument's controls will move from their current settings to the settings specified by the new Morph slider position in a time interval of the number of seconds specified in the Morph Time edit field (see figure below).

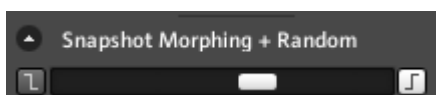


Fig. 4.15 Once you have configured the morph settings you can use the Morph slider to morph between two Snapshots.



Shorter Morph Time values decrease the delay between changing the Morph slider position and having the Panel controls complete their morphs. Longer Morph Time values increase this delay.

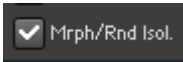
Making a Control Immune to Morphing

You can specify which Panel controls should not be affected by “morph” operations. To make a control immune to Snapshot morphing, follow these instructions:

- Double-click on the desired control like a knob, fader, list, or button.
- This should have opened up the control's Properties View. Now click on the Function button.

FUNCTION

- Engage the Mrph./Rnd Isolate checkbox in the Snapshots area of the control's Function page. Now the Morph slider will have no effect on that control.



To see how to assign a MIDI controller to the Morph function, please refer to the header “Specifying Common MIDI Controller Numbers” in subsection [↑13.2.1, MIDI In.](#)



To see how to assign a MIDI controller to the Morph function, please refer to the header “Specifying Common MIDI Controller Numbers” in subsection [↑13.2.1, MIDI In.](#)

4.6 Randomizing Snapshots

Probably the fastest way to create a new Snapshot is by using the **Randomize** or **Random Merge** buttons. The Snapshot Randomize area in the Snapshot tab (see figure below) holds the controls with which you can quickly create a new Snapshot with varying degrees of random deviation from the current Snapshot.



Fig. 4.16 The Snapshot Randomize area in the Snapshot tab lets you create new Snapshots with varying degrees of random deviation from the current Snapshot.

Randomizing Snapshots

If you want to randomize all of the selected Instrument's Panel controls (except those with the **Random Isolate** checkbox engaged, see below) follow these instructions:

- Make sure you have an Instrument open.
- Open the Snapshot tab and if the Snapshot Randomize area depicted above is not open, press the arrow button next to the text “Snapshot Morphing and Random”, shown below.



- First, decide what maximum amount of random deviation from the Instrument's current Panel controls you would like. Then enter the corresponding value into the Random Amount edit field. The Random Amount edit field accepts values in the range [0 ... 100]. "0" corresponds to 0 % and "100" corresponds to 100 % of the control's range. If we say the value in the Random Amount field is "R" then pressing the Randomize button can change the control's current setting up to $\pm R$ % of the control's range. For example, if a knob with a range [-1 ... 1] is set to its middle point "0" and the Random Amount edit field is set to "25" (meaning 25 %), then clicking on the Randomize button can change the knob's value to anywhere from -0.5 to 0.5, that is, $(0 \pm 0.25 * 2)$. Note that a control can never be randomized to a value beyond its range.



4. After having entered the desired value into the Random Amount edit field, click on the Randomize button.



Random Merging of Two Snapshots

You can also use two existing Snapshots as seeds and randomly merge these two “parent” Snapshots to create a “child” Snapshot. The resulting “child” Snapshot is a random combination of the two “parent” Snapshots with the amount of randomness being set with the Random Amount edit field. To use this feature, you need to follow these instructions:

1. Make sure you have an Instrument open with at least two Snapshots.
2. Open the Snapshot tab and if the Snapshot Randomize area depicted above is not open, press the arrow button next to the text “Snapshot Morphing and Random”, shown below.



3. You need to choose the first “parent” Snapshot. For that, click on the **Select A** button to activate it (its text should change to “Select...”).
4. Then select the first of the two “parent” Snapshots (Snapshot A) from the Snapshot List.
5. Now click the **Select B** button to activate it (its text should change to “Select...”).
6. Select the second of the two “parent” Snapshots (Snapshot B) from the Snapshot List.
7. To decide, what value you want to enter into the **Random Amount** edit field, consider the behavior of the “Random Merge” feature. The “Random Merge” operation is taken for each Instrument Panel control separately. Let's look at one control and let the value in the **Random Amount** edit field be denoted by the letter “R” (it is a value in the range [0 ... 100]). For a $R = 0$, pressing the **Random Merge** button results in the control receiving a new value that is exactly between the two values corresponding to the “parent” Snapshots. For $R = 100$, pressing the **Random Merge** button results in the control receiving either the value of “parent” Snapshot A or “parent” Snapshot B. In general, if $R > 0$, then pressing the **Random Merge** button causes the control randomly to be assigned to one of the two parents and the value “R” decides how much the new value of the control deviates from the perfect merge (arithmetic mean) of the two “parent” Snapshots in the direction of the assigned “parent” Snapshot. As mentioned above, for $R = 100$, the deviation is complete, that is, the new control value is that of the assigned “parent” Snapshot. Enter the value corresponding to the desired randomness into the **Random Amount** edit field.



8. Last, press the **Random Merge** button.



Making a Control Immune to Randomization

You can specify which Panel controls should not be affected by “randomize” operations. To make a control immune to randomization, follow these instructions:

1. Double-click on the desired control like a knob, fader, list, or button.
2. This should have opened up the control's Properties View. Now click on the **Function** button.



3. Engage the **Mrph./Rnd Isolate** checkbox in the Snapshots area of the control's Function page. Now the **Randomize** button will have no effect on that control. You can use this technique in conjunction with the **Random Amount** value to limit the amount of randomization an Instrument's controls receive.



4.7 Snapshot Banks

Each and every Snapshot in REAKTOR belongs to a Snapshot Bank. You can save your Snapshots on your hard drive or any other medium and even share your Snapshots by saving the Snapshots Bank to which that Snapshot belongs to. In this section you will learn all about Snapshot Banks in REAKTOR.

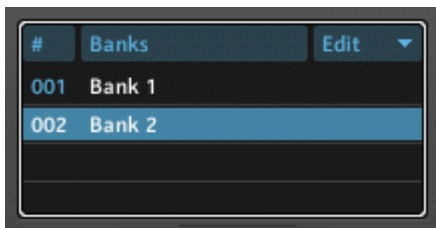
4.7.1 Saving, Loading, and Creating Snapshot Banks

This subsection will discuss the basic operations tied to Snapshot Banks.

Selecting Snapshot Banks

Some Instruments already come with several Snapshot Banks. To see which banks have already been loaded and to select a new bank, follow these instructions:

1. First, open an Instrument that has several Snapshot Banks already loaded. Space Drone is a good example of one, you can find it in the factory library in under **New Additions > Sound Generators**.
2. Open the Snapshot tab and look at the Snapshot Banks area shown in the figure below.

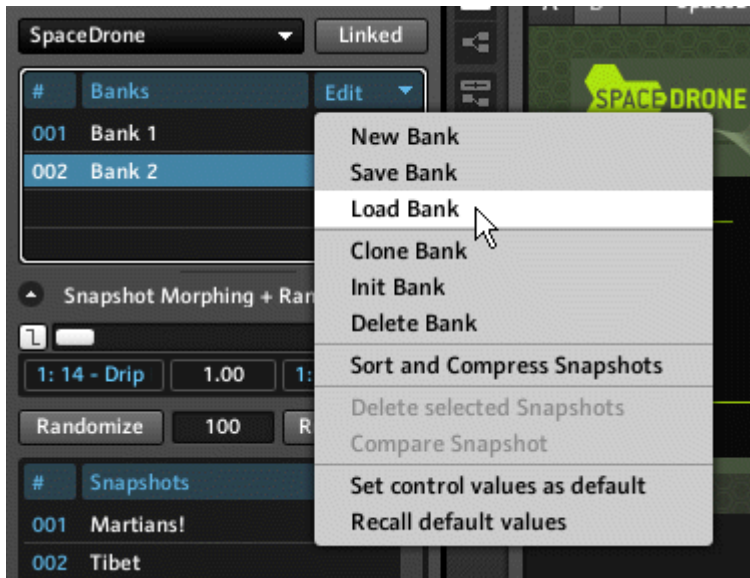


3. As you can see in the figure above, two banks have been loaded. Click on Bank 2 to load the second Snapshot Bank into the Snapshot List.

Saving and Loading Snapshot Banks

To save a Snapshot Bank, follow these steps:

1. Make sure you have an Instrument open.
2. Open the Snapshot tab and look at the Snapshot Banks area shown in the figure above.
3. To save the selected Snapshot Bank (see above), click on the **Edit Bank** drop-down menu and choose the *Save Bank* menu entry. This will open a dialog box to choose where you want to save the Snapshot Bank file. The Snapshot Bank file is save an *.ssf file.
4. To load a new Snapshot Bank into REAKTOR, choose the *Load Bank* menu entry from the **Edit Bank** drop-down menu, as shown in the figure below. This will open a dialog box to browse for the *.ssf Snapshot Bank file you wish to load.



5. After clicking on “OK” in the dialog box, the Snapshots contained in the Snapshot file you just loaded are appended to the current Snapshot List. If the list is full, then a new Snapshot Bank is created and the remaining Snapshots from the Snapshot file are loaded into that new Bank.
6. Usually one creates a new Bank right away before loading the Snapshots that are in a file into the Bank. This way the Snapshots in different Banks and Snapshot files don't get mixed up. To create a new Bank, click on the *Edit Bank* drop-down menu and choose the *New Bank* menu entry. This creates a new Snapshot Bank consisting of empty slots. You can now either start creating and saving your own Snapshots with the Instrument (see section [4.2, Adding and Removing Snapshots](#)) or load the Snapshots from a Snapshot Bank file into the Bank as shown in steps 5 and 6.

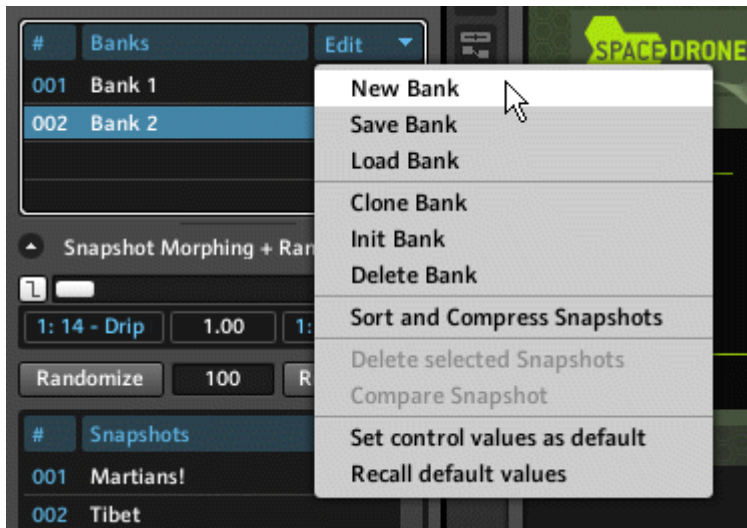


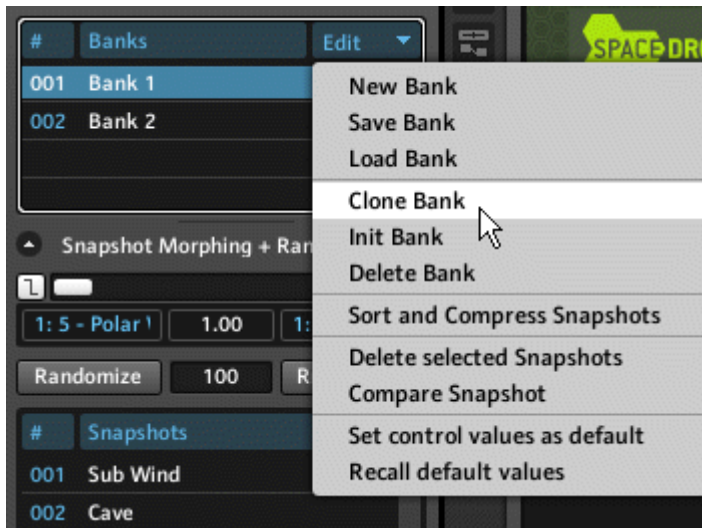
Fig. 4.17 To create a new Snapshot Bank, select the New Bank menu entry from the Edit Bank drop-down menu.

4.7.2 Cloning and Deleting Banks

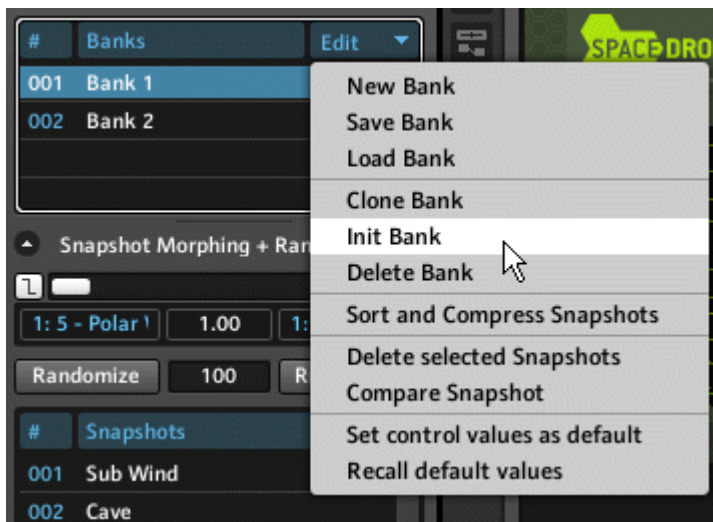
This subsection describes how to clone (duplicate), init (clear all Snapshots), and delete Snapshot Banks.

Cloning Banks can be useful in many cases. Maybe you want to duplicate a factory Snapshot Bank to create your own modifications to the Snapshots or perhaps you want to use the duplicate Bank to pick out only Snapshots that have a certain flavor to them. This is how you clone a Bank:

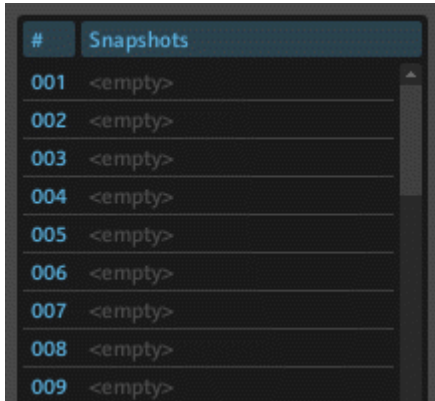
1. Make sure you have an Instrument open.
2. Open the Snapshot tab and click on the *Edit Bank* drop-down menu.
3. Click on the *Clone Bank* menu entry. A new Snapshot Bank will be created with its Snapshot List identical to the Bank that you cloned.



4. To clear all Snapshots from the currently selected Bank and replace them with empty slots, you need to click the *Init Bank* menu entry (see below).



5. This results in an empty Snapshot Bank (see figure below). And if you want to delete a Snapshot Bank completely, choose the *Delete Bank* menu entry. Note that you cannot delete a Snapshot Bank if it's the only one left for that Instrument.



If you delete a Bank by mistake, don't panic! Simply use REAKTOR's *Undo Delete Snap Bank* menu entry in the *Edit* application menu to restore it.



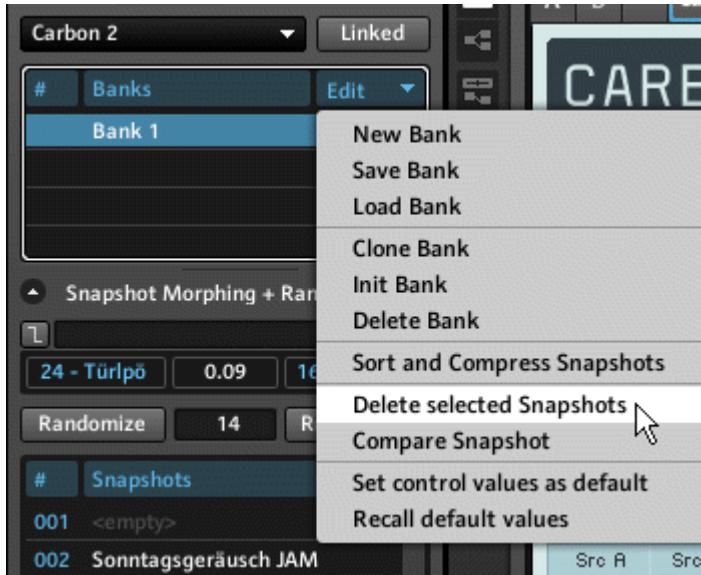
If you delete a Bank by mistake, don't panic! Simply use REAKTOR's *Undo Delete Snap Bank* menu entry in the *Edit* application menu to restore it.

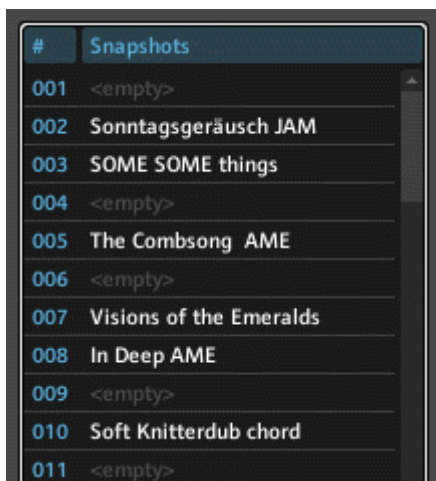
4.7.3 Renumbering and Renaming Banks

Let's say you want to take a factory Snapshot Bank, remove the Snapshots you don't like, and rename the new Bank. In this subsection you will first learn how to get rid of the holes in the Bank that result from deleting Snapshots by using the "Renumber Bank" feature. Then you will see how to rename your Bank. Perform the following steps:

1. Make sure you have an Instrument with at least one Snapshot Bank.
2. Now open the Snapshot tab and clone a Bank so that you don't lose the Snapshots you are about to delete in the cloned Bank (see subsection [↑4.7.2, Cloning and Deleting Banks](#) on how to clone Banks).

- Next, delete some Snapshots from the cloned Snapshot Bank by selecting the Snapshot Entry and then choosing the *Delete Snapshot* menu entry from the *Edit Bank* drop-down menu. The resulting Snapshot List should look something like the one shown below.





4.8 Defining Snapshot Behavior for Instruments

In Ensembles consisting of several Instruments, Snapshots of one Instrument often go together with particular Snapshots of another Instrument. Akkord is an example of such an Ensemble: it consists of a sequencer Instrument and a synthesizer Instrument. Therefore it is often desirable to recall or save the Snapshots for several Instruments simultaneously. You can also specify other specifics regarding Snapshot behavior of Instruments. This section discusses all things connected to an Instrument's behavior regarding Snapshots. The picture below shows the Snapshot area of an Instrument's Function page where the properties of its Snapshot behavior are set.

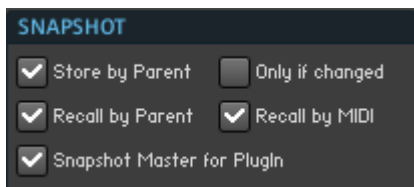


Fig. 4.18 An Instrument's Snapshot behavior is determined by the settings in the Snapshot area of the Instrument's Function page.

Snapshot Master and Linking Snapshot Save and Recall Operations

By default, Snapshots are stored and recalled independently for each Instrument. For example, say you have an Ensemble that contains two Instruments, a sequencer and a synthesizer. Recalling a Snapshot for the sequencer Instrument does not recall a new Snapshot for the synthesizer Instrument and vice versa. Sometimes this is just what you want. Other times, you might want to select Snapshots for multiple Instruments at the same time. You can do this by linking the Snapshots of one or several Instruments to the Snapshots of a “parent” Instrument or the Ensemble. When you recall the Snapshot of a “parent”, the corresponding Snapshots of all other Instruments in the Ensemble which have the “Recall Snapshot by Parent” feature enabled will also be recalled. Similarly, saving a new Snapshot for the “parent” Instrument causes all Instruments which have the “Store Snapshot by Parent” feature enabled to save a new Snapshot under the same name.

An Instrument is a “parent” of another Instrument if it holds that Instrument within its Structure. It follows, that the Ensemble is “parent” to all Instruments. In the case of the synthesizer and the sequencer, the Ensemble is the “parent” of both Instruments. You could enable the “Recall Snapshot by Parent” and “Store Snapshot by Parent” features in both the sequencer and synthesizer. This way if you store or recall Snapshots of the “parent” Ensemble, both the sequencer and synthesizer store and recall the corresponding Snapshots, respectively.

The “Snapshot Master” feature is particularly relevant when running REAKTOR as a plugin. When this feature is activated for an Instrument or Ensemble, the Snapshots List of that Instrument is forwarded to the host. It also appears in the Snapshot menu in the Main Toolbar. However, since most hosts can handle only one Snapshots List, this feature can be enabled for only one Instrument (or the Ensemble).

To see how to activate the previously discussed features, perform the following steps:

1. Make sure you have an Ensemble with two active Instruments open.
2. First you need to choose which Instrument is the Snapshot Master for your Ensemble. By default, it is the top level Instrument. To set an Instrument to be the Snapshot Master Instrument, go to the Function page of the Instrument Properties.

FUNCTION

3. Then engage the “Snapshot Master” checkbox. When the Snapshot Master for Plugin checkbox is engaged, the Instrument's Snapshots are available in the host program and appear in the Snapshot menu in the Main Toolbar. There can only be one Snapshot Master.
4. We will use the Ensemble (top-level Instrument) to as the “parent”. The “child” Instruments it holds can recall their Snapshots when the “parent” Instrument's Snapshots are recalled. To activate this feature for a “child” Instrument, go to that Instrument's Function page of the Instrument Properties.
5. Then engage the Recall by Parent checkbox.
6. Finally, choose which Instruments should store their Snapshots when the “parent” Instrument's Snapshots are stored. To activate this feature for an Instrument engage the Store by Parent checkbox in that Instrument's Function page.

Saving a “Child” Snapshot Only if Changed

Let's assume that you have a “parent” Instrument which holds a “child” Instrument that has the Store by Parent checkbox engaged. Then you can choose to have the “child” Instrument save a new Snapshot upon a “parent” Snapshot save operation only when the “child” Snapshot has been changed. This saves space in the Instrument's Snapshot List. You can activate this option by engaging the Only if Changed checkbox in the “child” Instrument's Function page.



Fig. 4.19 The Function button

Recalling Snapshots by MIDI

Each Snapshot Bank can hold 128 Snapshots. This number is not arbitrary; it has been chosen to provide optimal compatibility when recalling Snapshots via MIDI Program Change messages. To enable the Recall by MIDI option, engage the Recall by MIDI checkbox in the Instrument's Function page.



Fig. 4.20 The Function button

When the Recall by MIDI checkbox has been engaged, an incoming MIDI Program Change message with the value “N” (where “N” is an integer in the range [0 ... 127]) will recall the Snapshot with the number “N + 1” (if that Snapshot exists). Thus a Program Change message of “0” will recall Snapshot “1”, a message of “1” will recall Snapshot “2”, and so on. This way you can quickly and easily recall Snapshots from your MIDI device by issuing MIDI Program Change messages for the desired Snapshot number. If the “parent” Instrument also has this feature enabled and is on the same MIDI channel as another “child” Instrument, then the “parent” Instrument's Snapshot will take precedence. Please refer to subsection [↑13.2.1, MIDI In](#) to learn how to change the incoming MIDI channel number for an Instrument.

5 Player/Recorder Bar

The Player/Recorder Bar (see figure below) holds the File Player and the File Recorder. These two utilities allow you to play back and record audio files. The File Player is best suited for file playback and the “Player Auto Mode” feature makes it particularly easy to load files from the Browser. You can also listen to audio files using the File Recorder. However, the File Recorder's playback capability is meant for reviewing the content of a file after recording it. Further differences between the file playback of these two utilities are outlined in the following sections which describe the use of File Player and Recorder Player in detail.



Fig. 5.1 In the Player/Recorder Bar the File Player is on the left and File Recorder is on the right.

5.1 Using the Audio Player

The File Player (see figure below) enables you to route the Audio signal from an audio file (*.wav, *.aif, or *.aiff) into your REAKTOR Structure. The output of the File Player is sent directly to the first two Audio In Modules. This is particularly useful when building an effect Instrument. In such a case you would usually test how the Instrument processes the incoming Audio signal by using the output of the File Player. However, this also means that before you can actually hear the output of the File Player, you have to first route the signal from the Audio In Modules to the Audio Out Modules. In the case of testing an effect Instrument, the Audio In Modules would be connected to the effect's input ports that are designated for the incoming dry signal and the output ports of the Instruments should be connected to the Audio Out Modules. In general, an active signal flow from the Audio In Modules to the Audio Out Modules has to be created (see section [↑7.4, Creating an Active Signal Flow](#) for more information on active signal flows).



When the File Player is playing, then all other signals that would otherwise arrive at the first two Audio In Modules are overridden in favor of the File Player's output signal.

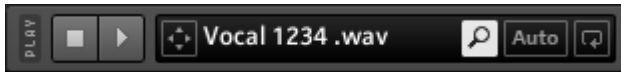


Fig. 5.2 The File Player is located on the left side of the Player/Recorder Bar.

Below you see a structure where you can hear the output of the File Player directly at RE-AKTOR's Audio output.

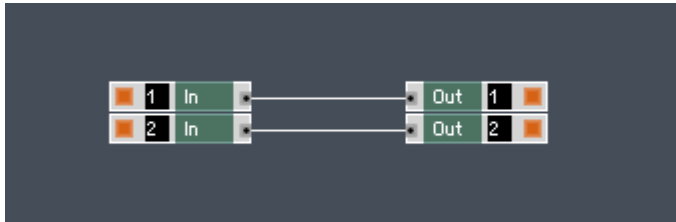


Fig. 5.3 Routing the Audio In Modules to the Audio Out Modules as shown in the figure results in RE-AKTOR's Audio output coming directly from the File Player.

! The audio file will play at the current RE-AKTOR Sample Rate, as set in the Main Toolbar (see section [↑3.5, Other Common Operations](#)).

5.1.1 Loading a File into the Player and Player Auto Mode

After making sure that the signal routing can accommodate hearing the File Player at RE-AKTOR's Audio output, the second thing you need to do in order to play back a file is load an audio file into the File Player. There are three ways to do this, described in the following list.

1. The first way is to double-click on any audio file in the Browser. This action opens the Player/Recorder Bar (if not already open), loads the file into the File Player and automatically starts the file playback.
2. Alternatively, you can drag and drop audio files into the text field of the File Player. In addition to dragging and dropping from the Browser, you can fetch files from other places like from the Desktop of your operation system. Again, after you have performed this action the File Player automatically starts the file playback.
3. The third way is to press the Player **Auto** button of the File Player. This activates the Player's Auto Mode. A single click on any file audio file in the Browser results in the file being loaded and played back in the File Player.

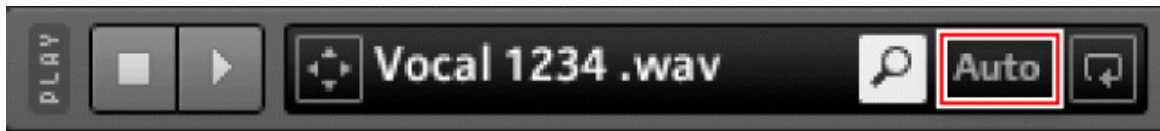


Fig. 5.4 The Player Auto button

! Pressing the Player Auto Button causes the Browser's Auto Mode to be deactivated.

5.1.2 Playback with the File Player

To learn how to use the Player Start and Player Reset buttons of the File Player, follow these instructions:

1. Make sure you have the first two Audio In Modules routed to the Audio Out Modules, either directly or through an Instrument.
2. Open the Player/Recorder Bar by pressing the Player/Recorder Bar button.



3. Now load an audio file into the File Player, as described in subsection [↑5.1.1, Loading a File into the Player and Player Auto Mode](#).
4. To playback the audio file, click the Player Start button. During playback, the Player Start button is lit and if the Audio In Modules are active (see section [↑7.4, Creating an Active Signal Flow](#)) then the Input Level Meter in the Main Toolbar shows the levels of the File Player's output signal (the keyboard shortcut for the Player Start button is Space).

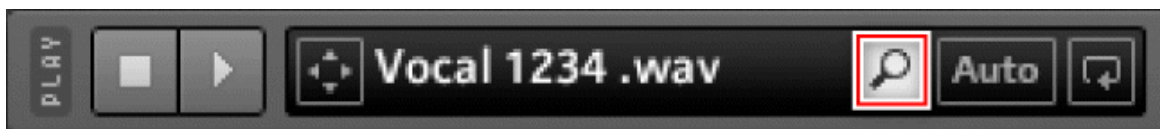


Fig. 5.5 The Player Browser button

Dragging and Dropping Audio Files from the Player

REAKTOR allows you to drag audio files which are loaded in the Player and directly drop them into Sample Maps (see chapter [↑6, Sample Map Editor](#) for more information on Sample Maps).

- To drag and drop an audio file from the Player, make sure that a file has been loaded into the Player. Then click on the File Drop button and drag and drop it onto a Sample Map or the Panel representation of a Sampler Module. This adds the loaded audio file to the corresponding Sample Map.



Fig. 5.6 The File Drop button

5.2 Using the Audio Recorder

The File Recorder (see figure below) enables you to record an Ensemble's audio output at its first two Audio Out Modules. The recorded audio signal is written to an audio file on your disk. Windows systems record *.wav files and OS X systems record *.aif files. You can then drag and drop the recorded files into Sample Maps, for example, making the resampling process fast and easy. The File Recorder can also play back audio files. The output is directly routed to the top two Audio Out Modules. This means that unlike for the File Player, you don't have to route anything in the Structure to hear the output of the File Recorder. However, since the output is routed to the Audio Output Modules, you cannot use the output of the File Recorder for your REAKTOR Structures.

- !** While playing back an audio file in the File Recorder, other signals at the input ports of the top two Audio Out Modules will be ignored.



Fig. 5.7 The File Recorder is located on the right side of the Player/Recorder Bar.

5.2.1 Recording an Audio Signal

To learn how to operate the File Recorder, follow these instructions:

1. Open the Player/Recorder Bar by pressing the Player/Recorder Bar button.



2. To create a new file to which the signal from the top two Audio Out Modules will be recorded to, press the New Recorder File button. The file name is created automatically by using the name of the last recorded file and appending the respective incremental number to its file name. If there is no last file name, the new filename will be set to “Untitled”.



3. Next edit the name of the file name, if you wish, by double-clicking the edit field that contains the file name and entering the new file name.
4. To start recording to the file, you need to press the Arm Recorder button. If in the previous step you entered a file name that does not exist in the currently selected folder, a new file will be created upon activating the Arm Recorder button. If the file name already exists, activating the Arm Recorder button causes a warning to pop up asking you if you want to overwrite the existing file. After switching the Arm Recorder button on, the button should have lit up. As long as the Recorder Start button has not yet been pressed, you can disarm the File Recorder by deactivating the Arm Recorder button. If the File Recorder has been disarmed without starting the recording, the empty file will be used when the New Recorder File button is pressed again. This way the creation of unnecessary empty files is avoided.



5. With the Arm Recorder button active, you can now press the Recorder Start button to start recording the signal at the first two Audio Out Modules. You can know that a file is being recorded by seeing both the Arm Recorder and Recorder Start buttons having an active state. For convenience there are two more ways to start the recording process. This is discussed in subsection [↑5.2.2, File Recorder Properties](#).



6. Just like with the File Player, press the lit Recorder Start button during the recording to be pause the recording.
7. You can see that the recording process is in the paused state by seeing the Recorder Reset and Recorder Start buttons both be unlit, whereas the Arm Recorder button to be lit. Press the Recorder Start button to continue with the recording process.
8. Now press the Recorder Reset button to stop the recording and reset the playback and record positions of the File Recorder to the beginning of the file. This also disarms the File Recorder, as you will see by the unlit Arm Recorder button. For convenience there are two more ways to reset the File Recorder. This is discussed in subsection [↑5.2.2, File Recorder Properties](#).



Dragging and Dropping Files from the Recorder

After recording an audio file, you can directly drag and drop them into Sample Maps (see chapter [↑6, Sample Map Editor](#) for more information on Sample Maps).

- To drag and drop an audio file from the Recorder, make sure that a file has been recorded and is still present in the Recorder. Then click on the File Drop button and drag and drop it onto a Sample Map or the Panel representation of a Sampler Module. This adds the recorded audio file to the corresponding Sample Map.



Fig. 5.8 The File Drop button

5.2.2 File Recorder Properties



Fig. 5.1 The Recorder Settings button

You can access the File Recorder's settings by pressing the Recorder Settings button. This opens up the Recorder Settings dialog box, shown in the figure below.

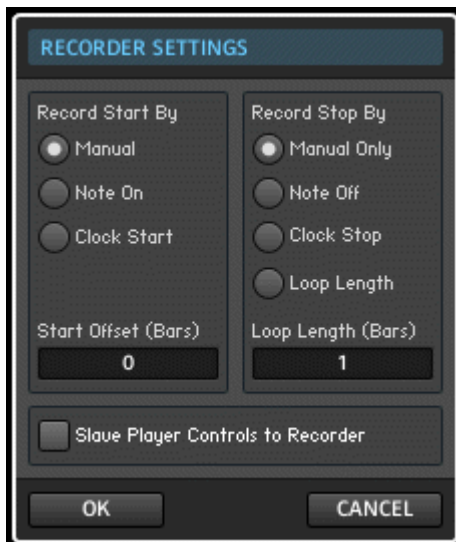


Fig. 5.9 Recorder Settings Dialog box

The Recorder Settings dialog box provides the following options:

- **Record Start By:** This radio button selector allows you to cause recording to be started (when the File Recorder is armed) not only by manually pressing the Recorder Start button, but also upon a Note On or a Clock Start event. When the Manual radio button is selected, recording can only be started by pressing the Recorder Start button. When the Note On radio button is selected, you can also start recording by a MIDI Note On event, that is, when a key is pressed on your MIDI or computer keyboard. When the Clock Start radio button is selected, recording starts when the REAKTOR Clock is started with the Start Clock button in the Main Toolbar (see subsection [↑3.3.3, Playing an Ensemble](#)) for more information on the Clock).



Fig. 5.10 The Recorder Start button

- **Record Stop By:** This radio button selector allows you to choose under which conditions the recording process is stopped. With the Manual Only radio button selected, you stop the recording process using the Recorder Stop button. When the Note Off radio button is selected, you can also stop recording by a MIDI Note Off event, such as when a key is released on your MIDI or computer keyboard. When the Clock Stop radio button is selected, recording stops when the REAKTOR Clock is stopped with the Stop Clock button in the Main Toolbar (see subsection [↑3.3.3, Playing an Ensemble](#) for more information on the Clock). When the Loop Length radio button is selected, the File Recorder stops recording when the loop length duration, as specified by the Loop Length edit field in bars, has been reached.



Fig. 5.11 The Recorder Stop button

- **Start Offset:** With the Start Offset edit field you can specify a time delay in bars before the recording starts after a “record” event (specified in the Record Start By radio button selector) has arrived. This can be useful if you want to record an echo effect in an advanced phase for instance, but you don’t want the beginning.
- **Loop Length:** The Loop Length edit field lets you specify the duration of the recording in bars. This value is only used when the Loop Length radio button has been selected.

- **Slave Player Controls to Recorder:** With this checkbox you can cause the File Player transport controls to follow those of the File Recorder. This can be useful when you want to use the File Player to send an audio file through a REAKTOR effect Instrument and simultaneously record the result at the output using the File Recorder.

6 Sample Map Editor

All Sampler Modules use Sample Maps to access and manage samples. Therefore it is important that you learn how to use the Sample Map Editor, particularly if you wish to load your own samples into sampler-based Ensembles. In this chapter you will learn how to access the Sample Map Editor, how to edit the Sample Maps, and how to audition, load, and configure samples for these Sample Maps.

6.1 Overview

6.1.1 Sample Maps

One common use of Sample Maps is to simulate the sound of acoustic instruments. This is usually done by assigning multiple samples (rather than a single sample) from the instrument to the instrument's pitch range. For example, rather than assigning a single clarinet sample to its entire three-octave range, you might assign a dozen samples to the range: one for the lowest three pitches in the range, another for the next higher three pitches, and so on, all the way to the highest three pitches. The further you stretch a sample (that is, the more pitches you assign it to), the less natural the simulation will sound. Therefore, to make an instrumental simulation sound natural, you need to use an appropriate number of samples.

Additionally, you can use Sample Maps to trigger multiple samples with one key. This is a technique used to further mirror an instrument's dynamics. Usually an instrument is sampled playing the same note with different dynamics (for example, soft to hard). Then the dynamically different samples are triggered by different key velocities (MIDI note velocity messages), resulting in a more expressive representation of the original sound.

6.1.2 Viewing the Sample Map Editor

Accessing the Sample Map Editor

There are four ways to access the Sample Map Editor:

- Press the **Sample Map** button in the Sidebar.



Fig. 6.1 The Sample Map button

- Choose the *View > Show Map Editor* menu entry.
- Press the shortcut key F9.
- Right-click on a Sampler Module on the Instrument Panel and choose the *Open Map Editor* menu entry.

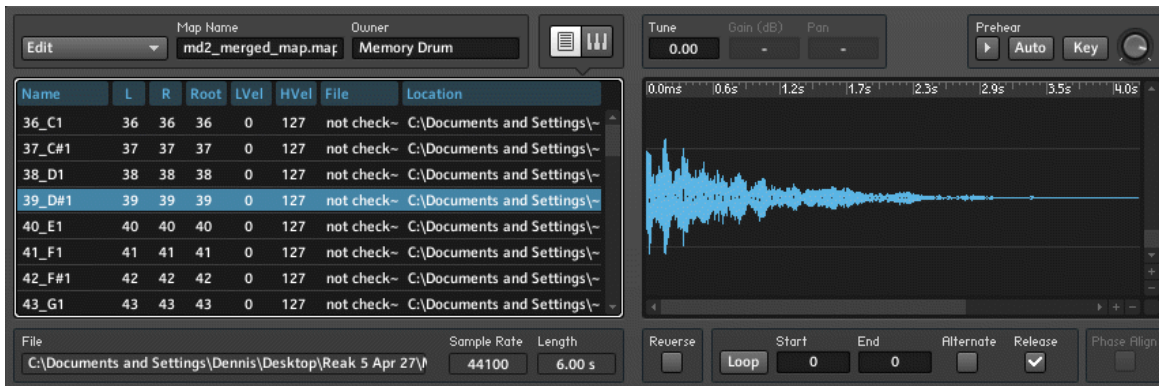


Fig. 6.2 The Sample Map Editor always shows two areas. On the left, you see either the Map List View (shown here) or the Map Keyboard View. On the right you always see the Sample Waveform View.

Switching Views

The Sample Map is displayed on the left side of the Sample Map Editor. There are two ways to display the Sample Map: the Map List View and the Map Keyboard View. These two Views are shown in the screenshot below.

- To switch to the Map List View, press the Map List button.



Fig. 6.3 The Map List button

- To switch to the Map Keyboard View, press the Map Keyboard button.



Fig. 6.4 The Map Keyboard button

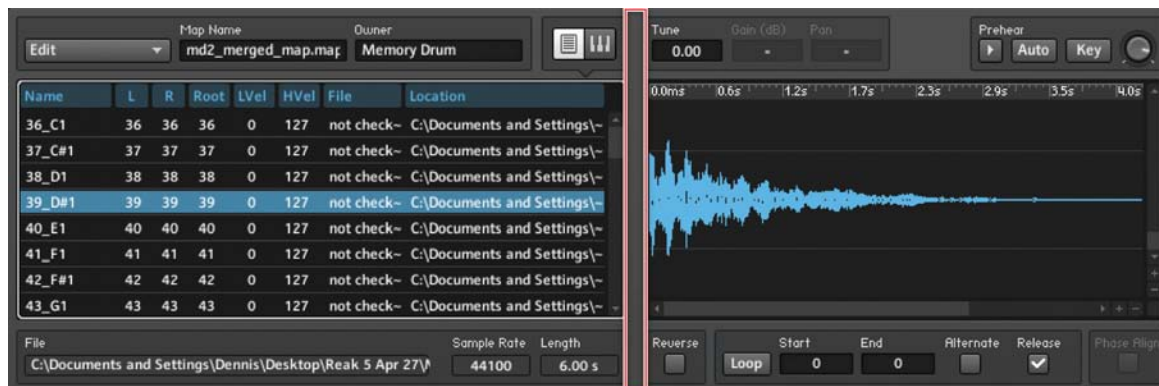


Fig. 6.5 The Sample Map can be displayed in the Map Keyboard View (left) and the Map List View (right).

6.2 Creating a Simple Sampler and Sample Map

Although the Sample Map Editor allows you to create Sample Maps for complex use cases, you might just be interested in creating a simple Sample Map to playback a few samples. In this subsection you will learn how to create a simple sampler Instrument that uses a Sample Map that comprises two samples.

As mentioned at the beginning of this chapter, Sample Maps are used to access and manage samples. Each Sampler Module uses a Sample Map to determine which samples to play back. Furthermore, every Sampler Module also has a “P” (pitch) input port. When a

sample is triggered, the value received at the “P” input port is used to look up in the Sample Map which sample should be played back and at which pitch. Usually the “P” input port is connected to a Note Pitch Module, which sends the current MIDI note.

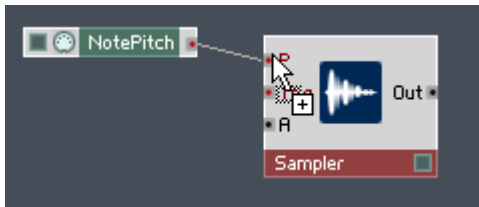
Some Sampler Modules also have a “Sel” (select) input port. If this input port is connected, the “Sel” value overrides the sample selection according to the value at the “P” input port, which then only sets the pitch. This can result in interesting sound variations.

Sample Maps are not only sensitive to the incoming pitch value, but also the MIDI note velocity with which the trigger or gate signal arrives at the Sampler Modules “Trig” or “G” input port, respectively. This means that you could trigger different samples with one key by just changing the velocity with which the key is pressed. Whether a Sampler Module receives its velocity information at a “Trig” or “G” input port, depends on the Module.

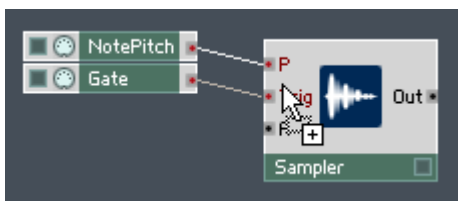
6.2.1 Building a Simple Sampler

To create a simple sampler for which you will create a Sample map in the next subsection, perform the following steps:

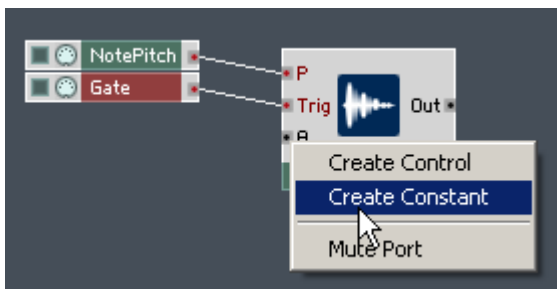
1. Create a new Ensemble and insert a Sampler Module from the *Built-In Module > Sampler* submenu into the Structure.
2. Next insert a Note Pitch Module from the *Built-In Module > MIDI In* submenu into the Structure.
3. Connect the output of the Note Pitch Module to the “P” input port of the Sampler Module, as shown in the figure below. Now the Note Pitch Module sends information about which MIDI note has been played to the Sampler Module and hence the Sample Map. Based on the MIDI note value (pitch), the Sample Map chooses which sample the Sampler should play back.



4. Next we will use the Gate Module to trigger the playback of the sample. The Sample Map not only uses MIDI note pitch but also MIDI note velocity to choose which sample should be played back. The velocity information is delivered by the Gate Module and should arrive at the “Trig” input port of the Sampler Module. Insert a Gate Module into the Structure. You will find it in the *Built-In Module > MIDI In* submenu.
5. Connect the output of the Gate Module to the “Trig” input port of the Sampler Module, as shown below.



6. The “A” input port of the Sampler Module determines the amplitude factor by which the sample playback signal is multiplied before it reaches the output of the Sampler Module. Currently the “A” input port is disconnected and therefore the sample playback signal is multiplied by zero. This results in an output signal that is constantly zero. Let's change this by setting the amplitude factor to “1”. To do this, right-click the “A” input port and choose the *Create Constant* menu entry, as shown below. This should create a Constant with the value “1” at the “A” input port of the Sampler Module.



7. Last, connect the output of the Sampler Module to the Audio Voice Combiner Modules, as shown below. You now have a simple sampler Instrument. However, since the Sampler has no Sample Map, it cannot play back any samples. Proceed to the next subsection to learn how to create a simple Sample Map and get some sound out of this Structure.

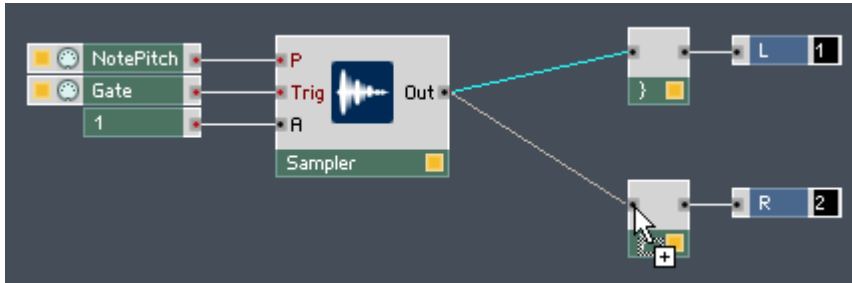


Fig. 6.6 Since your sampler Structure is Polyphonic, you need to send the output signal through Audio Voice Combiner Modules before routing it to the Instrument's outputs. Please refer to section 9.3 for more information on Polyphony.

6.2.2 Creating a Simple Sample Map

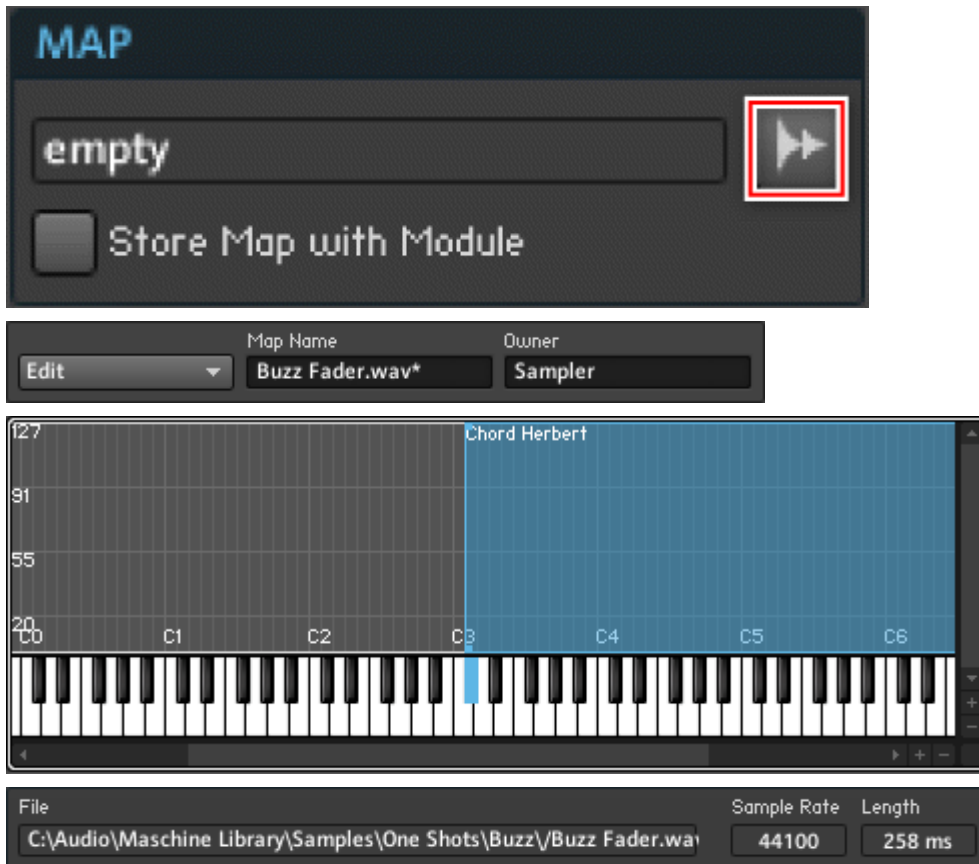
Now that you have the framework of the sampler that you created in the previous subsection you can create a simple Sample Map comprising two samples. Perform the following steps:

1. Make sure that you have the Structure open that you built in the previous subsection. Then open the Sampler Module's Properties by double-clicking the Module.
2. Go to the Function page by pressing the **Function** button.

FUNCTION

3. Open the Sample Map Editor by clicking on the Sample Map button. The Map area of the Sampler Module's Function page is shown in the below figure.





6.3 Editing the Sample List

The Sample Map Editor allows you to edit Sample Maps. In this section you will learn how the Sample Map functions and how to edit it in both the Map List and Map Keyboard Views.

6.3.1 Basic Editing

This subsection covers the most basic editing of the Sample Map: adding, replacing, saving, deleting, and updating samples. Also, you will see how to load and save Sample Maps.

Adding, Replacing, Saving, and Removing Samples

- To add a sample to the Sample Map, choose the *Add* menu entry from the *Edit* menu (see figure below). Alternatively you can right-click the Panel representation of the corresponding Sampler Module and choose the *Add Sample...* menu entry.

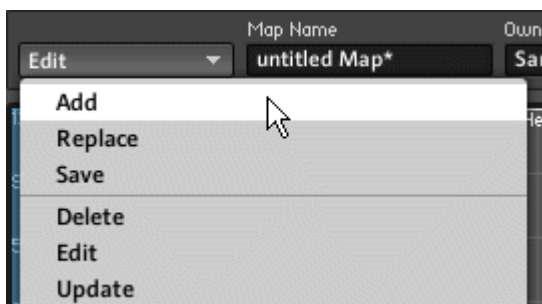


Fig. 6.7 Choose the Add menu entry to add a sample to the Sample Map.

- To replace a sample in the Sample Map with another one, select the sample and choose the *Replace* menu entry from the *Edit* menu (see figure below). Alternatively you can right-click the Panel representation of the corresponding Sampler Module and choose the *Replace Sample...* menu entry.

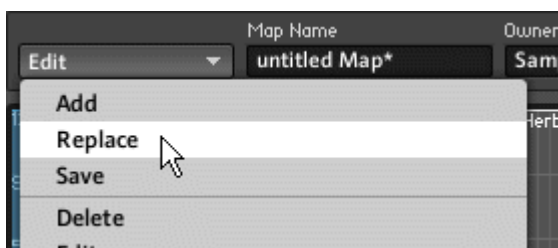


Fig. 6.8 Choose the Replace menu entry to replace the selected sample in the Sample Map.

- To save a sample, select the sample and choose the *Save* menu entry from the *Edit* menu, shown in the figure below. This option might be relevant when you have engaged the *Store Map with Module* checkbox in the *Sampler Module's Function* page. When the sample has been saved with the Module, but is no longer present on your computer's hard drive (see subsection [↑10.1.4, Storing Samples with Modules](#)), this is a good way to retrieve the sample. Also, the Sample Map Editor allows you to change loop information of the sample (see subsection [↑6.4.2, Setting Sample Playback and Loop Parameters](#)). Saving the sample also stores the loop information you edited in REAKTOR's Sample Map Editor.



Fig. 6.9 Choose the *Save* menu entry to save the selected sample.

- To remove a sample from the Sample Map, select the sample and choose the *Delete* menu entry from the *Edit* menu (see figure below).

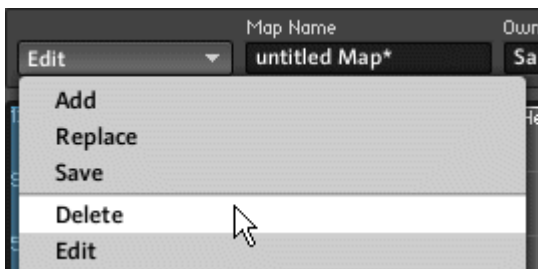


Fig. 6.10 Choose the *Delete* menu entry to remove the selected sample from the Sample Map.



You can also use the keyboard to cycle between samples in a Sample Map. Press *Tab* to cycle forward and *Shift+Tab* to cycle backward.

Loading and Saving Sample Maps

You can load and save whole Sample Maps as *.map files. This way Sample Maps into which you have put a lot of work can be reused by different Modules or Ensembles without much extra effort. Of course, in order for the Sample Map to remain intact, the samples to which the Sample Map refers should not be moved after the Sample map has been saved. However, if samples have been moved, then you can help the Sample Map Editor locate them using the *Replace* menu entry described above.

- To save a Sample Map as a *.map file, choose the *Save Map* menu entry from the Edit menu (see figure below). The *.map file can contain all the sample data that is used by the Sample Map or it can be just a small file with references to the sample files. When saving, a dialog box asks you whether you wish to embed the samples into the *.map file. By choosing this option the loss of samples can be avoided.

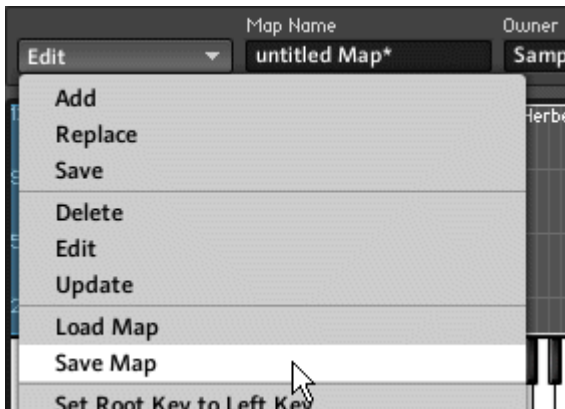


Fig. 6.11 Choose the Save Map menu entry to save the current Sample Map as a *.map file.

- To load a Sample Map from a *.map file, choose the *Load Map* menu entry from the Edit menu (see figure below). Alternatively you can right-click the Panel representation of the corresponding Sampler Module and choose the *Load Map...* menu entry.

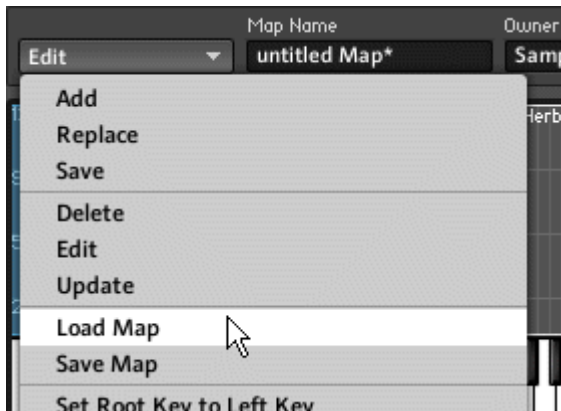


Fig. 6.12 Choose the Load Map menu entry to replace the current Sample Map with one that is in the form of a *.map file.

6.3.2 Sample List Functions

In addition to the basic options for editing Sample Maps, REAKTOR offers a few functions which can make editing easier when working with more complex Sample Maps.

- Choose the *Set Root Key to Left Key* from the *Edit* menu (see figure below) to set the Root value of all samples equal to the “L” value. This means that playing the pitch corresponding to the lower bound of a sample's MIDI pitch interval will play the sample at its original pitch. If you play higher pitches which are still within the MIDI pitch interval of the same sample, the sample will be played back but transposed to the corresponding pitch.

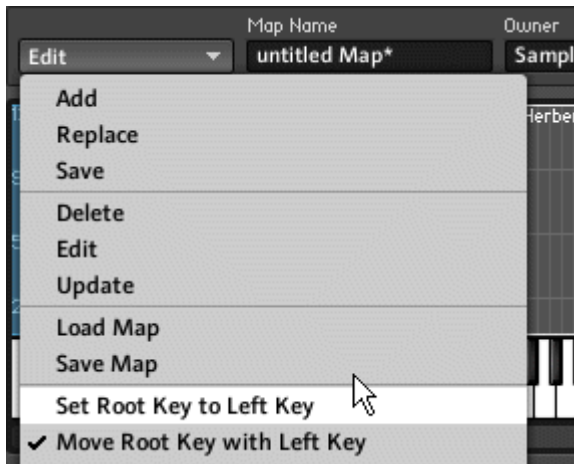


Fig. 6.13 The Set Root Key to Left Key menu entry sets the Root value of all samples equal to the “L” value.

- Sometimes you have already determined the relative Root key for a sample but still want to move the sample around in the Sample Map. Then, after moving the sample or its lower bound (“L” value) it is often convenient to have REAKTOR automatically find the new Root key relative to the new lower bound. To activate this feature, choose the *Move Root Key with Left Key* menu entry from the *Edit* menu (see figure below).

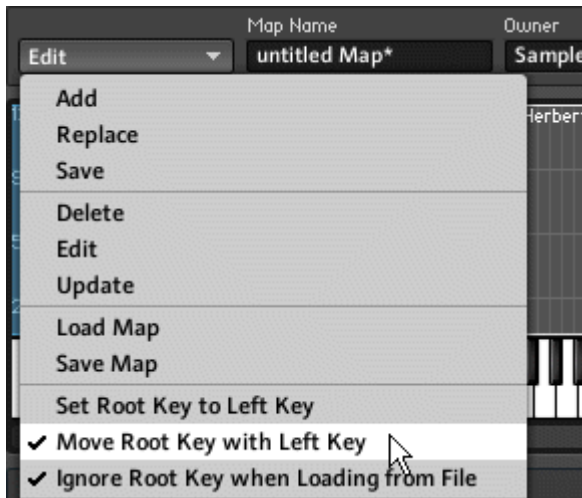


Fig. 6.14 Choosing the Move Root Key with Left Key menu entry causes the Root value to be moved along with any change of the lower bound of the sample's MIDI note pitch interval.

- It is possible to save the Root key information with the sample file. This can be conveniently reused when loading the sample into a Sample Map again. However, sometimes you don't want this to happen. In such a case you should choose the *Ignore Root Key when Loading from File* menu entry from the *Edit* menu (see figure below).



Fig. 6.15 To ignore any information tied to the Root value when loading a sample, choose the Ignore Root Key when Loading from File menu entry.

- Especially when working with Sample Maps where one drum sample is assigned to each key, it can be time-saving to have REAKTOR automatically assign one key to each sample. If you want to take advantage of this feature, load all the desired samples into the Sample Map and choose the *Remap to Single Keys* menu entry from the Edit menu (see figure below).

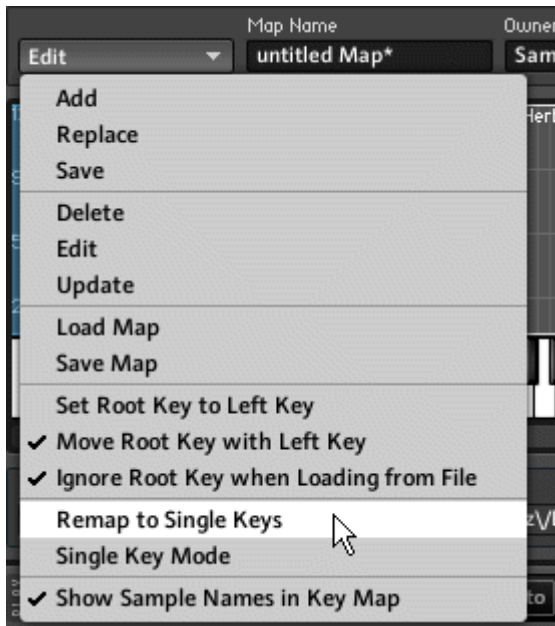


Fig. 6.16 Select the Remap to Single Keys menu entry if you want to completely remap all loaded samples, each to a different key.

- When working in the Map List View, you might want to work with individual samples that have only one key assigned to them. In such a case activate the *Single Key Mode* menu entry from the *Edit* menu(see figure below). As a result, changing the “L” value of a sample will automatically change the “R” value to the same value. Note that this feature is only meaningful in the Map List View.

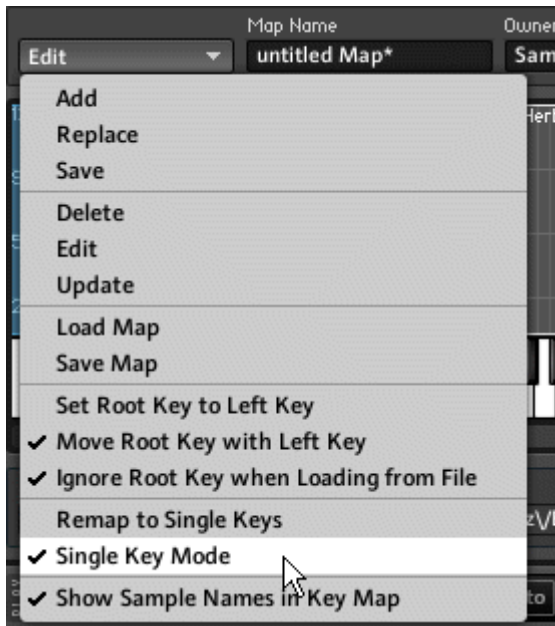


Fig. 6.17 In the Map List View you can use the Single Key Mode menu entry to have a sample be assigned to a single key, no matter what “L” or “R” value you choose.

6.4 Settings for Individual Samples

Since the Sample Map contains samples, REAKTOR also offers you options to customize the settings for individual samples. This section discusses these features which fall into three categories: auditioning samples, setting sample playback and loop properties, and calling up an external sample editor to further edit your samples.

6.4.1 Auditioning Samples

The Sample Map Editor has an integrated player which enables you to audition the samples in your Sample Map. This is particularly useful if you have many samples in the Sample Map and don't exactly know what all the samples sound like. The Sample Map Editor's

player is shown in the figure below. The Sample Map Editor's player is directly connected to REAKTOR's audio output. You will hear it play even if the associated Sampler Module is not connected to the Audio Out Module.



Fig. 6.18 The Sample Map Editor's player allows you to audition the samples that you have loaded into the Sample Map.

- To audition a sample in the Sample Map, select the sample and then press the **Prehear** button.



Fig. 6.19 The Prehear button

- If you want the player to automatically play back any sample you select in the Sample Map, press the **Auto** button.



Fig. 6.20 The Auto button

- Press the **Key** button to enable sample selection via the value at the “P” input port of the Sampler Module. Remember that for automatic auditioning the **Auto** button should be activated.



Fig. 6.21 The Key button

- Use the **Volume** knob to control the playback volume of your sample auditioning signal.

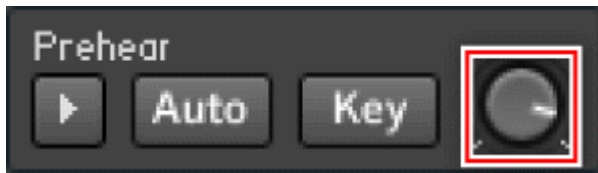



Fig. 6.22 The Volume knob

6.4.2 Setting Sample Playback and Loop Parameters

For each sample you can set some basic playback and loop parameters. This subsection discusses the options available to you.

 Not all options are available for every one of the Sampler Modules.

Basic Sample Playback Parameters

The edit fields, as shown in the below figure, allow you to set the basic sample playback parameters. Depending on the Sampler Module, some or all of the parameters are available for editing.

- To transpose the sample, change the value in the **Tune** edit field accordingly. The units are semitones where the fractions of the semitone are given in cents.
- You can also change the amplitude of the sample playback with the **Gain (dB)** edit field.
- The **Pan** edit field allows you to change the sample's placement in the stereo panorama field.

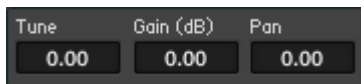


Fig. 6.23 The basic parameters of tuning, gain, and pan can be set with these edit fields.

Loop and Grain Parameters

The Sample Waveform View, shown in the below figure, provides you with the visual feedback and necessary controls to create loops out of loaded samples.

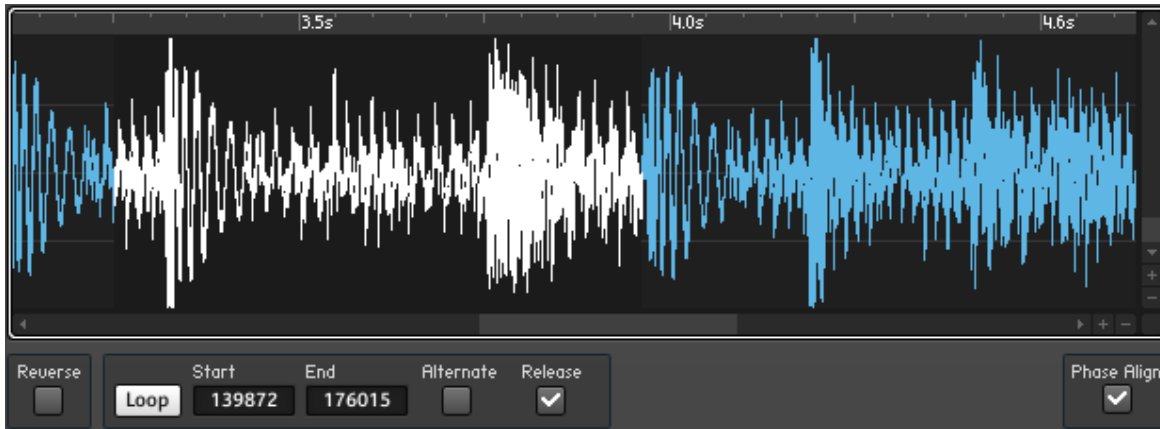


Fig. 6.24 The Sample Waveform View gives you convenient controls to change the loop and grain parameters. Use the scrollbars and Zoom buttons to navigate the sample waveform.

- To loop the whole or part of the selected sample, press the Loop button.

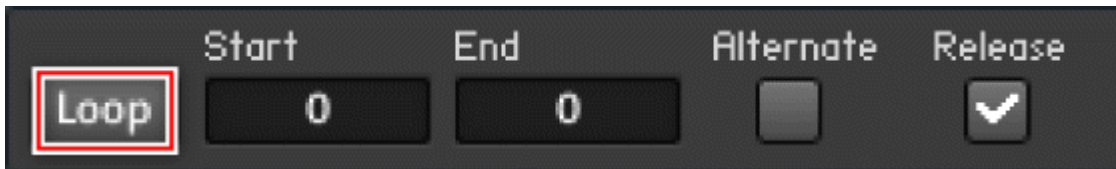


Fig. 6.25 The Loop button

- Change the loop starting and end points either with the Start and End edit fields (which are given in samples) or click and drag the loop area in the Waveform display. Clicking and dragging the left or right border of the loop area changes the loop start and end points, respectively. Clicking and dragging the middle of the loop area moves the loop area forward or backward. Note that these controls only have an effect if the Loop button has been engaged.

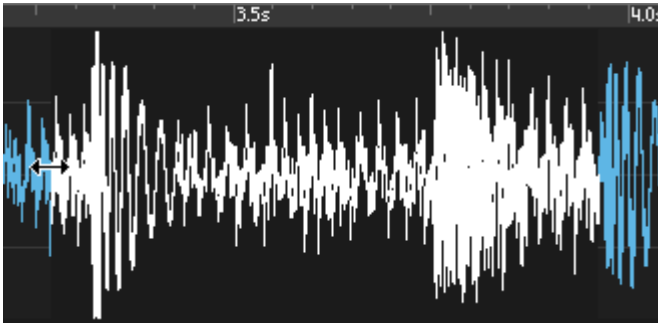


Fig. 6.26 You can also change the start and end points of the loop area by clicking and dragging the corresponding borders in the Waveform display.

- You can activate the Alternating Loop Direction Mode by engaging the **Alternate** checkbox. This causes the playback direction of the sample to change when the playback position has reached either one of the loop area's borders.

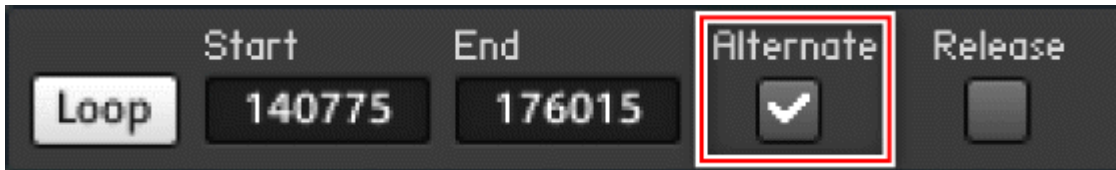


Fig. 6.27 The Altern. checkbox

- To change the (initial) playback direction of the sample, engage the **Reverse** checkbox.



Fig. 6.28 The Reverse checkbox

- Engage the **Release** checkbox to cause the loop to continue playing after the key has been released.

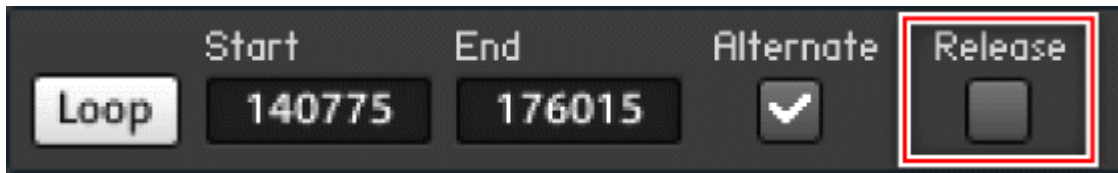


Fig. 6.29 The Release checkbox

- Sampler Modules which use granular resynthesis have the option to phase align the grains. This can improve the overall sound quality, although it requires more CPU resources. To activate this feature, engage the *Phase Align* checkbox.

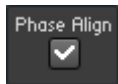


Fig. 6.30 The Phase Align checkbox

6.4.3 Using an External Sample Editor

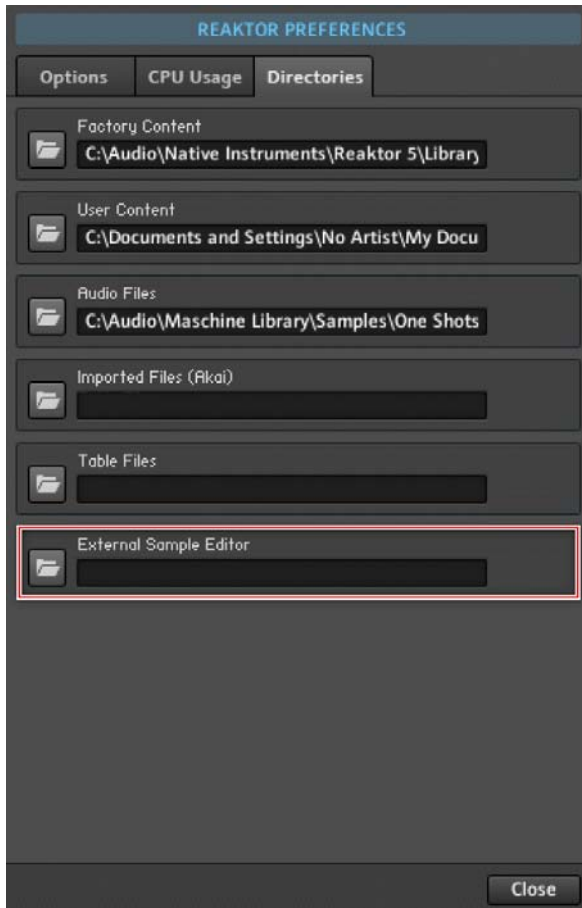
REAKTOR doesn't have its own internal sample editor. To edit sample files, you have to use an external audio editor. For example, in Windows you could use Sound Forge, Wave-Lab, Audition, Audacity, or similar. For Mac OS X Peak, Soundtrack Pro, and Audacity are some options. To facilitate this process, REAKTOR enables you to open a sample file in your chosen editor from within the Sample Map Editor. To do this, follow these instructions:

1. First, specify the location of your sample editor by entering the pathname of the program in the Preferences dialog. To access the Preferences dialog, click on the *File > Preferences* menu entry.
2. Now click on the *Directories* tab.



3. You should see the dialog shown in the screenshot below. As described in subsection [↑14.4.2, Directories Tab \(Setting File Paths\)](#), this dialog is meant for setting up common paths for REAKTOR. The bottom-most path is meant for the pathname of your sample editor. Type in the pathname directly into the *External Sample Editor* edit field or

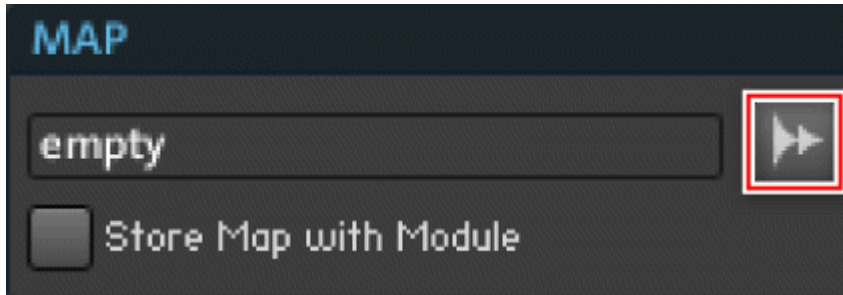
click on the folder icon next to that edit field to browse for the location of your sample editor. If you choose the latter method, find and click on the executable or application of your sample editor and click “OK”.




4. Now close the dialog box by clicking on the “x” icon on the top right of the window.
5. Once you have specified the location of your sample editor, you can try out the link between REAKTOR and the sample editor. Make sure you have a Sampler Module in your Structure. Double-click on it to open its Properties.
6. Now click on the Function button.

FUNCTION

- Next, open its Sample Map by clicking on the Sample Map button.



- Now choose or load a sample which you would like to edit.
- Click on the Edit menu and select the *Edit in External Editor* menu entry. The specified sample editor program should now open the selected sample.
- Edit your file in the program as desired. Once you are done, save it.
- Now go back to the Sample List menu in the Sample Map Editor and select the sample you edited.
- Choose the *Update* menu entry from the Edit menu to reload the modified sample file.

 Keep in mind that some sample editors will ignore loop information present in sample files. In such cases the loop regions will be lost.

7 Structure View

The focus of this chapter is learning how to create your own Ensembles by combining and modifying Instruments in the Primary Level Structure View. You will get an overview of the fundamental operations in Structure View as you are shown how to work with Objects such as Instruments and Modules, connect those using wires, and route the signal to the output so that an audible result is achieved.

! Note that you should have read the REAKTOR Getting Started Manual before reading Part II and III of the REAKTOR Main Manual. By reading the Getting Started Manual, you will get an intuitive understanding for building functioning Primary Level Structures using Instruments, Macros, and Modules. This will help you grasp many of the concepts described in these Parts of the Main Manual.

7.1 Inserting Objects into the Structure

In this section you will be shown how to insert Instruments, Macros, Modules, and Core Cells into the Structure. There are two ways to do this. The first one, as shown in subsection [↑3.3.1, Loading a KOMplete Instrument](#), is by using the Browser. The second way is by right-clicking on the Structure background when in the Structure View. We will look at the second option in a little more detail and also explore what the differences between the different types of objects are.

7.1.1 Inserting Instruments into the Structure

In a nutshell, Instruments are the containers for your Structures. In addition, Instruments let you specify musically relevant properties for your Structure such as the Polyphony and connections to other Instruments and the MIDI interface of your computer. See chapter [↑13, Instruments: Interfaces for Your Structures](#) for a more detailed discourse on the subject of Instruments. To insert an Instrument into the Structure, follow these steps:

1. First create a new Ensemble as shown in subsection [↑3.3.5, Creating a New Ensemble](#).

2. Click on the Structure View Icon to open the Structure View. You should see an (empty) Instrument connected to the Audio In and Audio Out Modules by a total of four wires.



3. Right-click on the Instrument to open the context menu and choose the menu entry *Delete*. This will remove the empty Instrument and let you insert your own Instrument into the Structure.
4. To insert an Instrument into the Structure without the use of the Browser, right-click on the Structure background to open up the context menu.
5. Move your mouse over the *Instrument* menu entry. The *Instrument* submenu should now offer you different categories of Instruments to insert into the Structure (see screenshot below).
6. Click on the menu entry of the desired Instrument to insert it into the Structure.

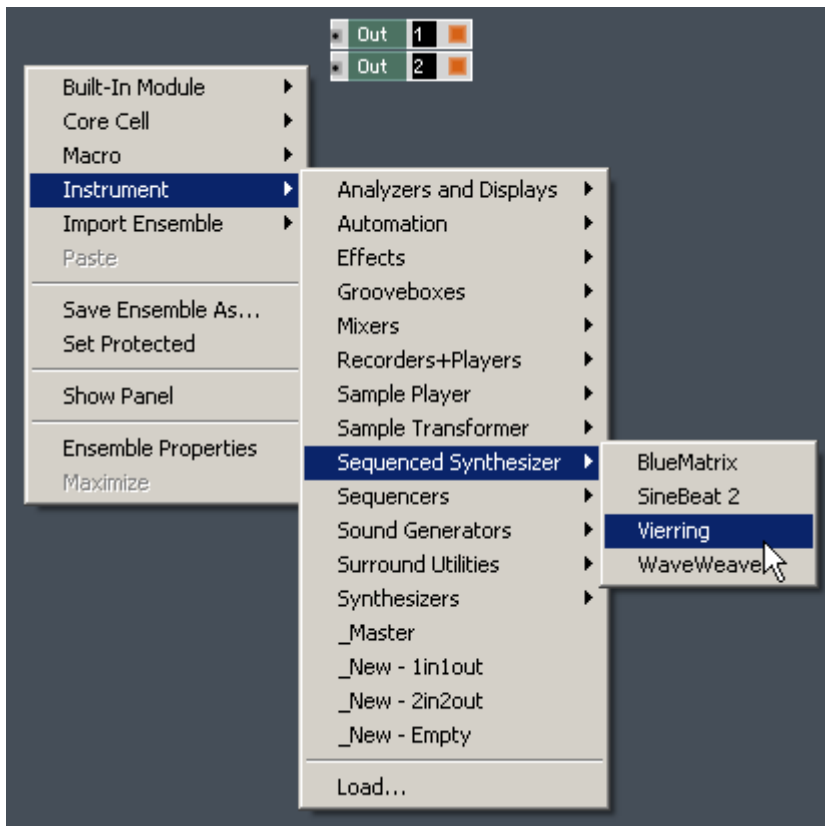


Fig. 7.1 Right-clicking on the Structure background opens up a menu for inserting Modules, Core Cells, Macros, or Instruments into the Structure.

- 💡 If your submenu does not look like the one shown in the figure above, make sure your user paths are set correctly, as instructed in subsection [↑14.4.2, Directories Tab \(Setting File Paths\)](#).
- 💡 You can browse your computer for additional Instruments by clicking on the *Load...* menu entry in the *Instrument* submenu.
- 💡 You can also insert Instruments when in Panel View: just right-click on an empty area in the Panel and then choose an Instrument from the *Insert Instrument* menu.

Import Ensemble as Instrument

REAKTOR allows you to import Ensembles as Instruments into your Structure. This means that all the global settings of the Ensemble that you are importing are discarded. The imported Ensemble then appears as an Instrument Object in your Structure.

- To import an Ensemble into your Structure, right-click the Structure background and choose the menu entry corresponding to the desired Ensemble from the *Import Ensemble* submenu (see figure below).

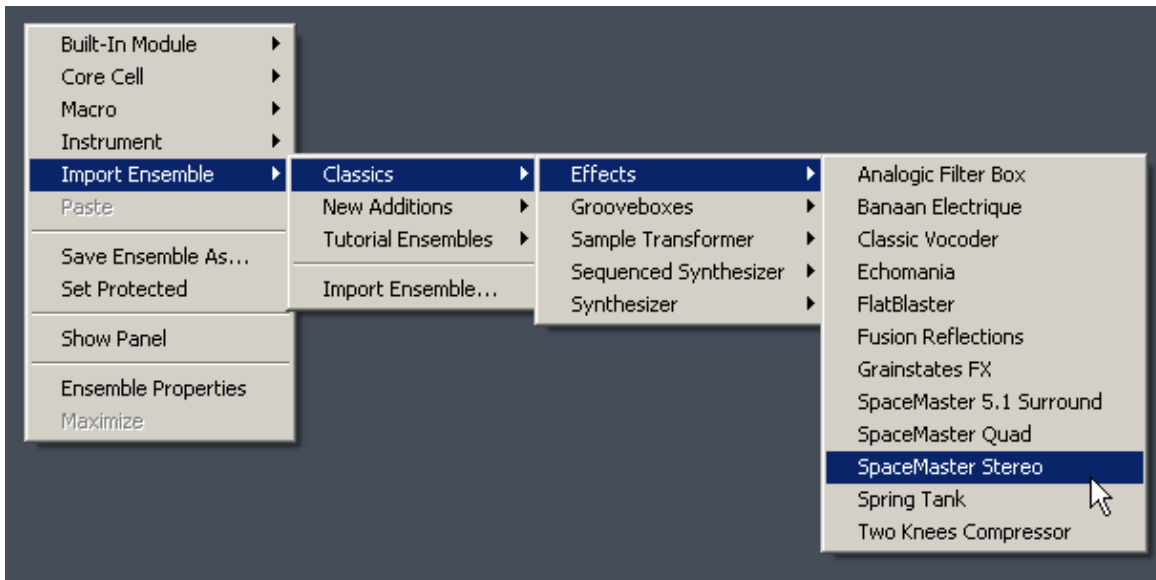


Fig. 7.2 Use the Import Ensemble submenu in the Structure background context menu to find an Ensemble to import as an Instrument.

7.1.2 Inserting Macros into the Structure

Similar to Instruments, Macros are also containers for your Structure, but with one big difference: Macros do not have dedicated settings for the number of Voices, MIDI connections, and so on. Macros serve to promote the building hierarchical Structures that grant the beholder a better overview of the signal flow and facilitate reusability of certain Structures. If you decide that for your tastes, the ideal combination of oscillators for a subtractive synthesizer is made up of an impulse, sawtooth, and parabol oscillator, then you can

pack these into a Macro and add this Macro to your Macro library. Now, every time that you want to implement these oscillators in a Structure, you can just insert the Macro into the Structure. This saves you time and energy by circumventing the hassle of combining the oscillators and equipping them with Panel Elements all over again.

- To insert a Macro into the Structure, follow the same steps as given in subsection [↑7.1.1, Inserting Instruments into the Structure](#), but instead of choosing from the *Instrument* submenu, you can find the Macro library in the *Macro* submenu.



You can add Macros to your Instrument when in Panel View. First unlock the Panel by clicking on the Panel Lock button. Then right-click on the Panel and choose a menu entry or submenu from the *Insert Macro* menu.



Fig. 7.3 The Panel Lock button

7.1.3 Inserting Modules and Core Cells into the Structure

Modules and Core Cells are building blocks of your Primary Level Structure. Everything that you can imagine doing in the Primary Level can be done using these two components. With Module you can create signals with oscillators, filter signals with a variety of filters, sequencing external gear with MIDI-specific Modules, or even visualize an incoming signal with a Scope Module. In this context, one main difference between Primary Level Modules and Core Cells is that while Primary Level Modules include Panel Elements, Core Cells cannot communicate with the Instrument Panel without having Primary Level Modules as mediators. Another aspect is that Core Cells can be modified in the Core Level of REAKTOR and you are free to create your own Core Cells whenever you like. These aspects regarding the Core Level of REAKTOR are addressed in much detail in the Core Reference.

- To insert a Module or a Core Cell into the Structure, follow the same steps as given in subsection [↑7.1.1, Inserting Instruments into the Structure](#), but instead of choosing from the *Instrument* submenu, you can find the Module and Core Cell libraries in the *Built-In Module* and *Core Cell* submenus, respectively.



In section [↑7.5, Choosing the Right Objects](#) you are presented with an overview of the different types of Modules, and Core Cells so you get an idea when to use them.



You can also insert Modules from the “Panel” category of the Built-In Modules when in Panel View. First unlock the Panel by clicking on the *Panel Lock* button. Then right-click on the Panel and choose a menu entry from the *Insert Built-In Module* menu.



Fig. 7.4 The Panel Lock button

7.2 Basic Operations in the Structure

In order for you to work effectively with the Structure, you need to know how to move, select, duplicate, delete, copy, cut, and paste Objects. This section will show you how to do just that.

7.2.1 Selecting Objects

- To select an Object, just click on it.
- To select multiple Objects, click on the Structure background and drag a rectangle to encompass the Objects to be selected (see figure below).

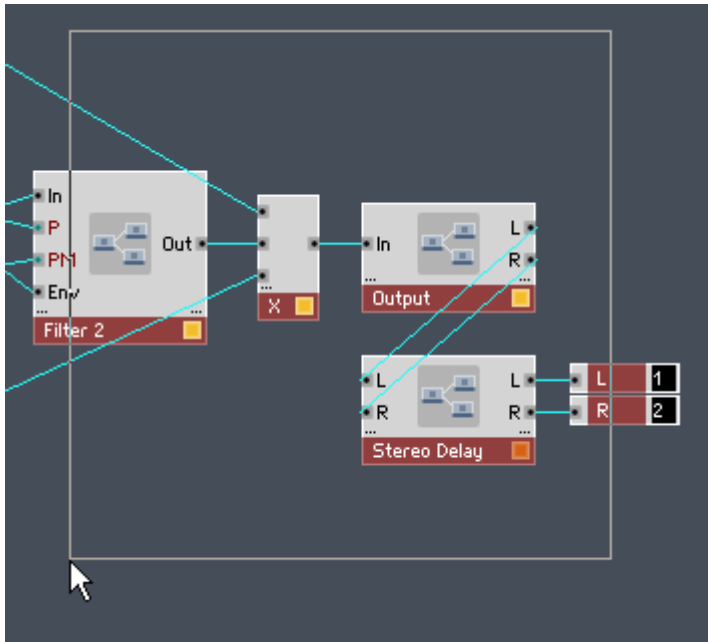


Fig. 7.5 Click and drag to create a rectangle which selects the desired Objects.

- Add an additional selection to your existing selection by holding down Shift while dragging a rectangle to select further Objects.
- You can add additional single Objects to your selection by clicking on the desired Object while holding down Ctrl for Windows (Cmd for Mac OS X).
- To deselect an Object, click on it while holding down the Ctrl key for Windows (Cmd key for Mac OS X).

Select All Objects

Sometimes you want to perform an operation on all the Objects that are in the Structure hierarchy level. You could manually select all Objects as described above, or you could save time and use the Select All command by performing the following steps:

1. First insert a few Objects into the Structure. If you don't remember how to do that, see section [↑7.1, Inserting Objects into the Structure](#) for instructions.
2. Right-click the Structure background to open the context menu.

3. Click on the *Select All* menu entry.

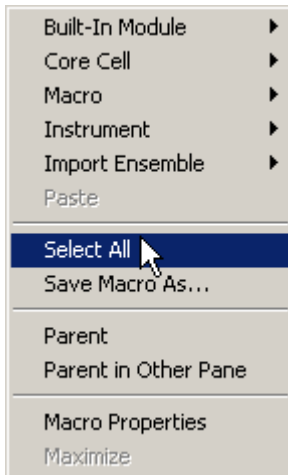


Fig. 7.6 Right-click on the Structure background and choose the *Select All* menu entry to select all Objects.



The keyboard shortcut for selecting all Objects is Ctrl+A on Windows and Cmd+A on Mac OS X. The *Select All* command also exists as the *Select All* menu entry in the *Edit* application menu. You can also press the *Menu* button and choose the *Edit > Select All* menu entry after having selected the Object to be deleted.



You can also *Select All* Modules and Macros on the Instrument Panel. For this you have to be in Panel View (see subsection [3.3.1, Loading a COMPLETE Instrument](#)): just right-click on an empty part of the Instrument Panel and then select the *Select All* menu entry. In order to be able to select all Modules or Macros on the Panel, you need to have unlocked the Panel by clicking on the *Panel Lock* button.



Fig. 7.7 The Panel Lock button.

- To deselect all objects, click on the Structure background.

7.2.2 Moving Objects

- To move an Object, simply click it and drag it to its new location.

7.2.3 Duplicating Objects

When you duplicate an Object in the Structure View, a copy of the Object is inserted into the Structure along with all the incoming connections that the original Object had. This operation also applies when you have several Objects selected. To duplicate an Object, follow these steps:

1. First insert an Object into the Structure. If you don't remember how to do that, see section [↑7.1, Inserting Objects into the Structure](#) for instructions.
2. Right-click the object to open the context menu.
3. Click on the *Duplicate* menu entry.



The keyboard shortcut for duplicating the selected object is Ctrl+D on Windows and Cmd+D on Mac OS X. The duplicate command also exists as the *Duplicate* menu entry in the *Edit* application menu. You can also press the **Menu** button and choose the *Edit > Duplicate* menu entry after having selected the Object to be duplicated.



Fig. 7.8 The Menu button



To duplicate an Object when in Panel View, right-click on the Module, Instrument, or frame of the Macro. Then left-click on the *Duplicate* menu entry. In order to be able to duplicate Modules or Macros from the Panel, you need to have unlocked the Panel by clicking on the **Panel Lock** button.



Fig. 7.9 The Panel Lock button

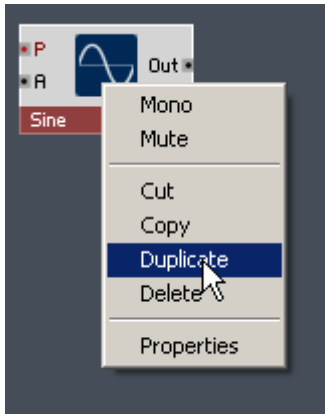


Fig. 7.10 The context menu of an Object in the Structure.

7.2.4 Deleting Objects

Removing unwanted Objects from the Ensemble happens through the *Delete* menu entry. This operation also applies when you have several Objects selected.

1. First insert an Object into the Structure. If you don't remember how to do that, see section [↑7.1, Inserting Objects into the Structure](#) for instructions.
2. Right-click the object to open the context menu.
3. Click on the *Delete* menu entry.



The keyboard shortcut for deleting the selected object is Del on Windows and on Mac OS X. The delete command also exists as the *Delete* menu entry in the *Edit* application menu. You can also press the *Menu* button and choose the *Edit > Delete* menu entry after having selected the Object to be deleted.



Fig. 7.11 The Menu button



You can also delete an Object when in Panel View: just right-click on the Module, Instrument, or frame of the Macro and then left-click on the *Delete* menu entry (shown in the figure below). In order to be able to delete Modules or Macros from the Panel, you need to have unlocked the Panel by clicking on the Panel Lock button.



Fig. 7.12 The Panel Lock button

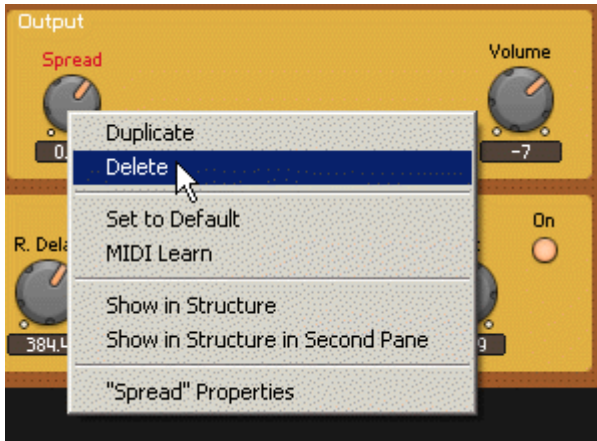


Fig. 7.13 You can use the Delete menu entry from the Instrument Panel to delete Objects which have Panel representations.

7.2.5 Copying, Cutting, and Pasting Objects

The commands of Copy, Cut, and Paste are implemented in the same way in REAKTOR Structure as on other conventional platforms such as Windows or Mac OS X. To do any of these actions, follow the following steps (they apply also when you have several Objects selected):

1. First insert an Object into the Structure. If you don't remember how to do that, see section [↑7.1, Inserting Objects into the Structure](#) for instructions.
2. Right-click the object to open the context menu.
3. Click on the *Cut*, *Copy*, or *Paste* menu entries for the corresponding action. A cut or copy action is generally followed by a paste action anywhere in the Structure of the same Ensemble.



The keyboard shortcuts for the Copy action is Ctrl+C for Windows (Cmd+C for Mac OS X), for the Cut action Ctrl+X (Cmd+X for Mac OS X), and for the Paste action Ctrl+V (Cmd+V for Mac OS X). The copy, cut, and paste commands also exist as the corresponding menu entries in the *Edit* application menu. You can also press the *Menu* button and choose the corresponding menu entry from the *Edi* submenu.



Fig. 7.14 The Menu button

7.2.6 Saving Objects

- You might not only want to save Ensembles, but also Instruments, Macros, or Core Cells. To save one of those Objects, right-click on it (either in Structure or Panel View) and choose the *Save *Object* As...* menu entry. This brings up a browser dialog window with which you can choose the name and path of the corresponding file.



Instruments are saved as *.ism files, Macros are saved as *.mdl files, Core Cells are saved as *.rcc files, and Core Macros are saved as *.rcm files. Additionally, Ensembles are saved as *.ens files.

7.3 Connecting Objects

Objects in REAKTOR are connected using wires. The wires carry a signal and the points at which the signal enters and leaves an Object are called ports. There are different types of signals for which there are different types of ports. In this section you will learn about these signal types and how to connect Objects.

7.3.1 Dragging Wires between Objects

Wires are vessels for the different signals that flow through your Instrument. For an Instrument that creates an audible signal at its output ports, this signal needs to be connected to the Audio Out Module that lies at the top level of your Ensemble Structure. The Audio Out Module represents the Audio interface of your computer. If your Audio and MIDI settings are correct, every Audio signal that is connected to the input ports of the Audio Out Module reaches your soundcard. To demonstrate how to connect a freshly inserted Instrument to your soundcard by connecting its output ports to the Audio Out Module by dragging wires to its input ports, follow these instructions:

1. Create an empty Structure and insert the Instrument “2-Osc.ism” into the Structure. You will find it in the *Instruments > Synthesizers* submenu. If you don't remember how to do this, refer to the Getting Started Manual.

2. Left-click on the “L” output port of the 2-Osc Instrument and while holding the mouse button, drag to the “L” input port of the Audio Out Module. A wire should follow your mouse cursor. When you reach the input port, release the mouse button. If a wire has successfully been created, a line should connect the “L” input and output ports. See figure below for an illustration of this.
3. Repeat the process for the “R” input and output ports.
4. Play the Instrument using your MIDI keyboard or computer keyboard. You should be able to hear the sound of the synthesizer that is carried from the Instrument to your soundcard by the wires you just created.

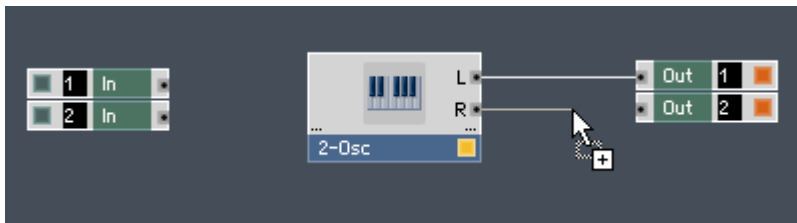


Fig. 7.15 Connecting two Objects with by wires.



See the Getting Started Manual if you are having problems with configuring your soundcard or MIDI interface in REAKTOR.

Rules for Wiring

When connecting Modules with wires, it is best to remember these general rules:

- A wire can only connect an output port to an input port or vice-versa. It can never connect an input port to an input port or an output port to an output port.
- An output port can be connected to as many 40 input ports.
- When no wire is connected to an input port, it receives a zero valued signal, meaning that the value at the port is “0”.
- An Event input port cannot process Audio signals. If an Event input port is to be fed from an Audio output port, the signal must first be converted to an Event signal. This is done using the A to E Module. For more information on this Module, please refer to the Module Reference manual that is available as a PDF file in the subfolder “Documentation” of REAKTOR's installation folder.

- An Event output port can be connected to Audio input ports as well as Event input ports.

7.3.2 Port Types

In REAKTOR's Primary Level there are two different types of signals: Audio signals and Event signals. An Audio signal is a signal that is updated at the sampling rate and can be routed to your sound card. Event signals can roughly be defined as control signals that are updated at a much slower, sub-audio, rate. Examples of Event signals are LFOs, envelopes, and triggers.

There are three types of ports in REAKTOR:

- Audio ports only accept connections carrying Audio signals; they are colored black.



Fig. 7.16 Audio port

- Event ports only accept connections carrying Event signals; they are colored red.



Fig. 7.17 Event port

- So-called “hybrid” ports accept connections carrying either Audio or Event signals; they are colored green. When a connection is made to a free hybrid port, the color will change depending on the type of connection that was made.



Fig. 7.18 Hybrid port



Please read sections [↑9.1, Audio signals](#) and [↑9.2, Event Signals](#) for more information on Audio and Event signals.



If a red cross appears over your port, it means that the port is muted (no signal arrives at this port). This is usually because of incompatibility between Polyphonic output ports and Monophonic input ports. See section [↑9.3, Polyphony](#) for more details on Polyphony.

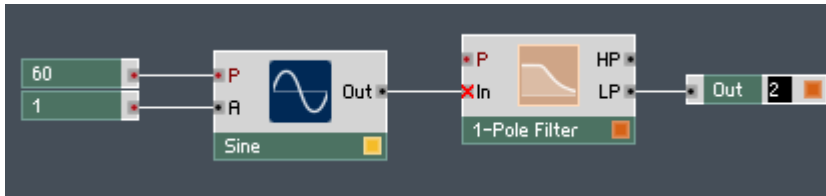


Fig. 7.19 A muted port is denoted by a red cross over said port. No signal arrives.

7.3.3 Dynamic Ports

“Dynamic Ports” is a feature that helps you create additional input or output ports of an Object simply by dragging a wire to the vicinity of the port area. All Instruments, Macros and many of the Modules (Add and Multiply, for example) have dynamic ports. If an Object has dynamic ports it is indicated by three small dots at the bottom of the port area of the Object.

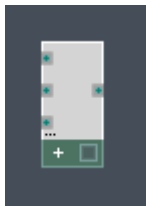


Fig. 7.20 An Add Module with dynamic input ports.

To create additional input ports for a Macro, follow these instructions:

1. Create a new Ensemble as shown in subsection [↑3.3.5, Creating a New Ensemble](#).
2. Insert a new Macro into the top-level Structure of the empty Instrument by right-clicking the Structure background and choosing the *Macro > _New - Empty* menu entry.
3. Now you have a Macro in the Structure without any input or output ports. When a wire is dragged to an empty part of the input port region (the left edge of the Macro) of a dynamic-input Module while holding down the Ctrl key (Cmd key on Mac OS X), a new input port is created automatically. See below for an illustration of this.

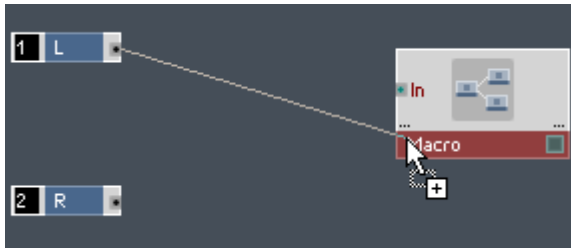


Fig. 7.21 Creating an additional input port using the dynamic ports feature of the Macro.

7.3.4 Port Order

For Instruments, Macros, and some other Objects like the Selector and Distributor Modules REAKTOR allows you to change the order of the input and / or output ports. In this subsection a small tutorial will illustrate this feature with a Macro. Perform the following steps:

Creating a Macro with Four Ordered Output Ports

1. Create a new Ensemble.
2. Insert an empty Macro into the Structure by right-clicking the Structure background and choosing the *Macro > _New - 2in2out* menu entry.
3. Double-click the Macro to see its internal Structure.
4. Select the two Out Modules labeled “L” and “R” (see subsection [↑7.2.1, Selecting Objects](#) for selecting Objects).
5. Duplicate these Modules by using the keyboard shortcut Ctrl+D if you are on Windows (Cmd+D if you are on Mac OS X).

Now you should have four Out Modules. Notice how each Out Module has a number on its right edge. These numbers denote the order of the output port, counted from above, when looking at the Macro from “outside”. Rename the Out Ports so that the uppermost Out Module has the Label “A”, the second the Label “B”, the third the Label “C”, and the fourth the Label “D”. You can rename an Object by double-clicking on it to open its Properties in the Sidebar and then entering the new Label in the Label edit field (shown in figure below).



Fig. 7.22 The Label edit field in the Object Properties lets you change the Label of the Object.

The internal Structure of your Macro should now look something like the one shown in the figure below. Double-click the Structure background to navigate out of the Macro:

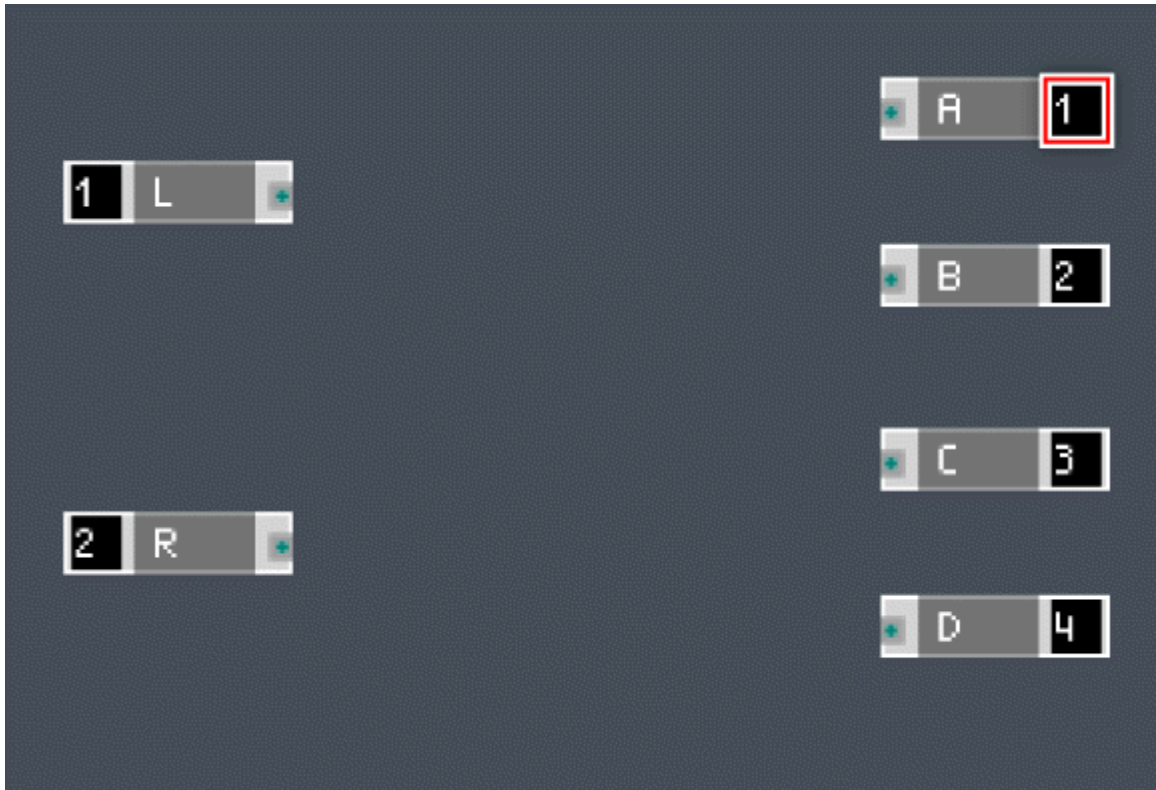


Fig. 7.23 The numbers on the edge of the In and Out Modules denote the order of the corresponding port on the outside of the Instrument or Macro when counted from the top.

Looking from outside, your Macro should look like the one shown in the screenshot below. To change the order of this Macro's output ports proceed to the instruction steps under the next heading.

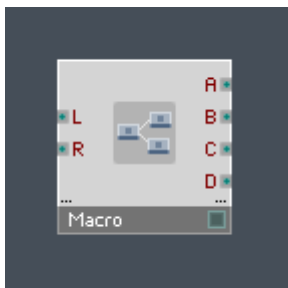
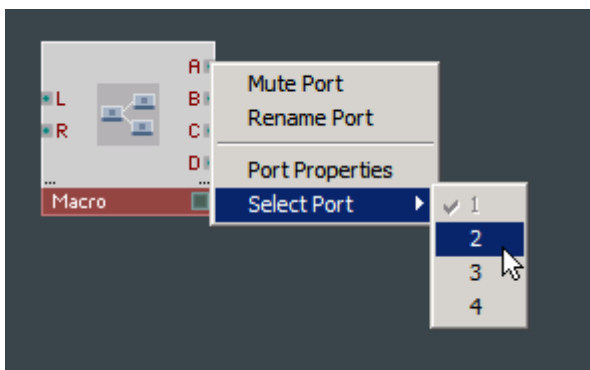


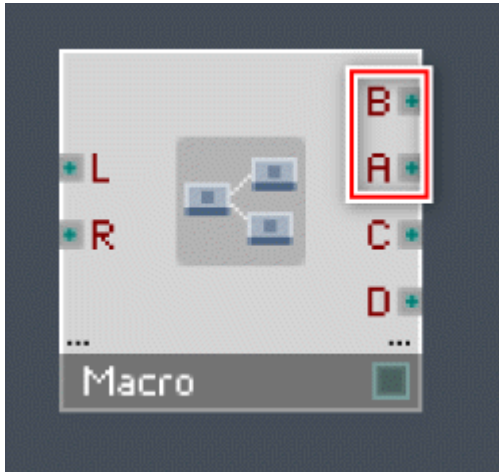
Fig. 7.24 A Macro with ordered output ports “A”, “B”, “C”, and “D”.

Changing the Output Port Order of a Macro

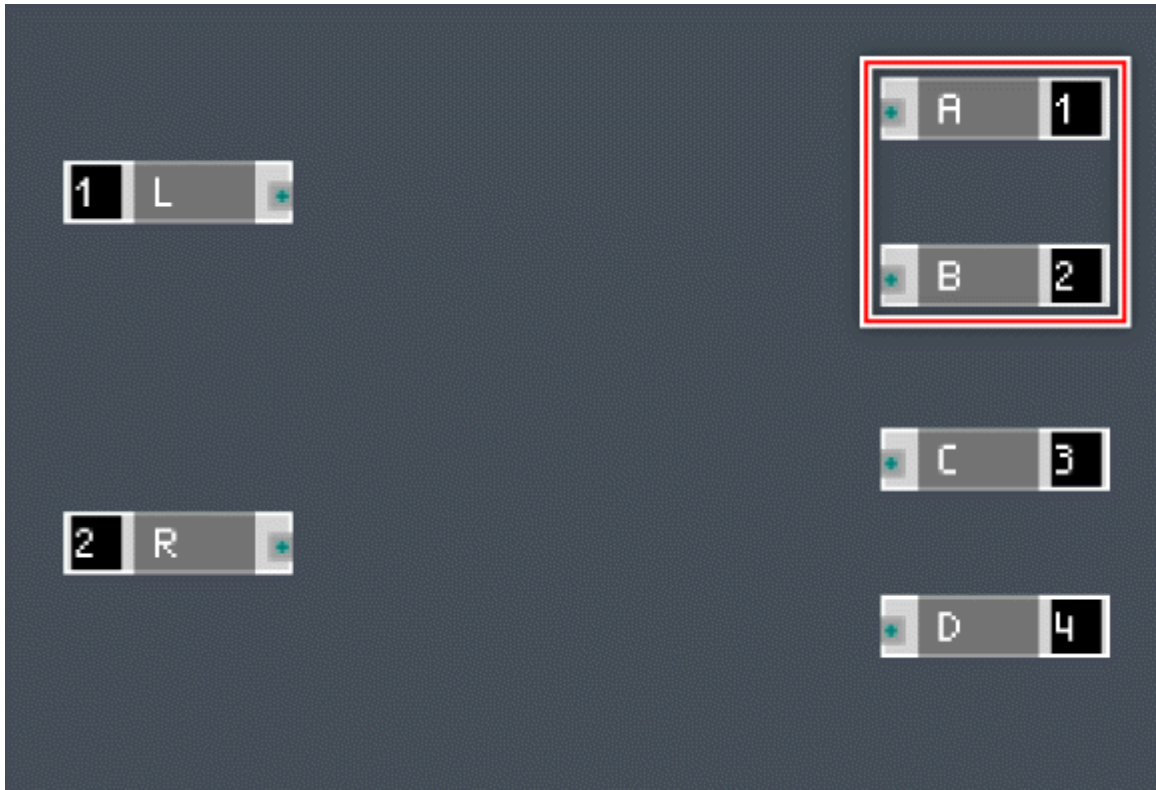
1. Continuing with the Macro you created in the previous heading, let's change the order of its output ports. In particular, let's swap the positions of the “A” and “B” output ports. To do this, right-click on the “A” output port and open the *Select Port* submenu.
2. In the *Select Port* submenu you can choose the position of the corresponding port. Right now you should see that there is a checkmark next to the menu entry *1*. This means that the port is currently in the first position counted from the top of the Macro. To change the position of the “A” output port to the second position counted from the top, click on the *Select Port > 2* menu entry (see figure below).



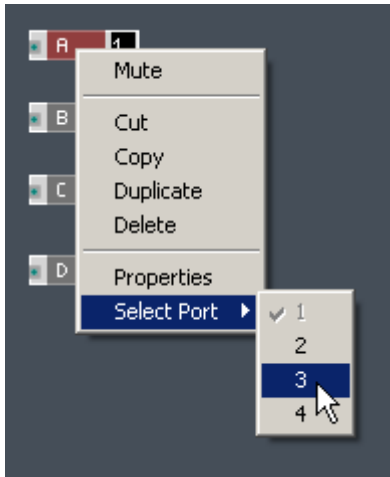
- Now you should see that the “A” output port has moved to the second position, swapping places with output port “B” that was previously at that position. Last, let's look at the situation one more time from inside of the Macro. Double-click the Macro to navigate to its internal Structure.



- The inside of your Macro should now look something like the Structure shown in below figure. The numbers on the right edge of the Out Modules signal that the Module labeled “A” is now in the second position and the Module labeled “B” is in the first position, counting from the top. Last, let's see how to change the port order from inside of the Macro. Right-click on the “A” Out Module.



5. The context-menu of the “A” Module also has the *Select Port* submenu, shown in the screenshot below. You can also use this submenu to change the order of the ports from the inside of the Macro.



7.4 Creating an Active Signal Flow

In REAKTOR, Objects will remain inactive if they are not connected to another active Object. Inactive objects will not send or process any signal. For this purpose, it is important to know how to create a signal flow that comprises active Objects and that is exactly what this section is about.

7.4.1 Activating Objects

Objects only become active once they are connected to a signal path that ultimately ends with the Audio Out Module or with a Module that has the *Always Active* checkbox engaged. This is in part for CPU reasons, but also to show you which Objects are actually contributing to your signal. Each Object in the Primary Level has a lamp in its bottom right corner. This lamp will light up yellow or orange if the Object is active and will be dimmed otherwise.

To activate an object, connect its output to a signal path that ends with an active Module. In the screenshot below, you can see an example of an active and an inactive Module. Some Modules, such as some of the Panel Elements, have the Always Active property set-

ting. Engaging this checkbox, which you find in the Function page of the Module Properties, activates the Module regardless of its connections and activates all Objects lying upstream from it.

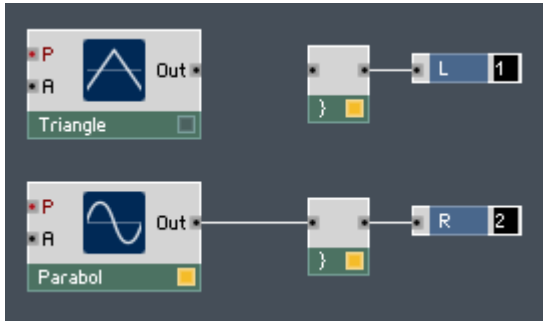


Fig. 7.25 The Triangle Module is inactive whereas the Parabol Module has been connected to the output (that is in turn connected to the Audio Out Module, not pictured) and is therefore active.

7.4.2 Muting Objects

Sometimes you want to deactivate an Object or a number of Objects without removal from the Structure. This could be the case when you want to try out how your Ensemble functions without that particular part of the Structure or when you want to deactivate a part to save CPU resources for another task. To mute an Object, follow the following steps (they apply also when you have several Objects selected):

1. First insert an Object into the Structure. If you don't remember how to do that, see [section ↑7.1, Inserting Objects into the Structure](#) for instructions.
2. Right-click the Object to open the context menu.
3. Click on the *Mute* menu entry.

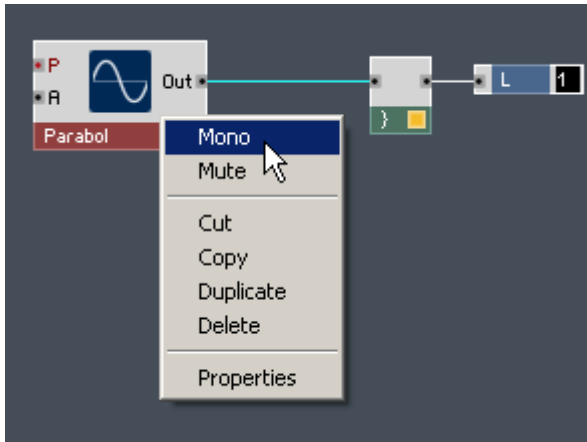


Fig. 7.26 The context menu of an Object in the Structure.

7.4.3 Signal Flows Where REAKTOR Is an Effect Unit

In this subsection you will be shown an example of an active signal flow utilizing the Audio In Module to make REAKTOR act as an effect unit. Follow these instructions to create a chorus effect Instrument:

1. First create a new Ensemble as shown in subsection [↑3.3.5, Creating a New Ensemble](#).
2. Go to the Structure View and delete the empty Instrument.
3. Now insert the Chorus SE-IV Chorus Instrument by clicking the *Instrument > Effects > SE-IV Chorus* menu entry.
4. Connect the “L” and “R” input ports of the Instrument to the “1” and “2” output ports of the Audio In Module. This causes the chorus effect Instrument to take the signal that the soundcard routes to REAKTOR's inputs as its dry signal.
5. Now connect the “L” and “R” output ports of the Instrument to the “1” and “2” input ports of the Audio Out Module. The lamp at the bottom right corner of the Instrument Object in the Structure should light up.
6. If your Audio and MIDI settings are configured correctly, you should now be able to send an Audio signal from your soundcard to REAKTOR, use REAKTOR to add a chorus effect to the signal, and hear the result from your soundcard. You can adjust the

parameters of the chorus effect from the Instrument Panel. If you want, you can add additional effects to your signal flow to create an ensemble comprising a chain of multiple effects.



Fig. 7.27 You can use REAKTOR as an effect unit by using the signal from the Audio In Module as the dry signal for your effect Instrument and then routing the output of the effect to the input ports of the Audio Out Module.



See the Getting Started Manual if you are having problems with configuring your soundcard or MIDI interface in REAKTOR.

7.5 Choosing the Right Objects

When building or modifying Instruments, the types of Objects you decide to insert into your Structure—Instruments, Macros, Modules, or Core Cells—depend on your need at that moment. This section serves to give you rough guidelines in making this decision.

7.5.1 Which Object Type to Choose

Instruments

Let's start with Instruments. The case where you would choose to insert an Instrument into your Structure is quite straight-forward. Instruments are suited for the case when you are building an Ensemble consisting of ready-built Instruments such as synthesizers and effects or when you want to insert a ready made effect into your custom-built synthesizer. The analogy to this would be inserting VST effects behind a synthesizer or adding and mixing together VST synthesizers in your host.

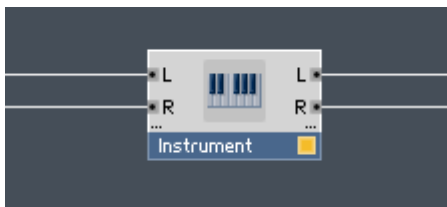


Fig. 7.28 An Instrument

Saving Instruments

If you have modified an Instrument and decide to save it, you can right-click on the Instrument and choose the *Save Instrument As...* menu entry. This opens your operation system's browser that lets you save content as usual. Make sure you choose your User Path as the save path if you want to be able to access the Instrument directly from the “*Insert Instrument*” menu. See the subsection [↑14.4.2, Directories Tab \(Setting File Paths\)](#) in regard to setting up your User Path. You can also do the same thing when in Panel View. In that case you need to right-click on the Instrument Panel to find the *Save Instrument As...* menu entry.

Macros

Macros can be applied in a variety of cases. First off, consider the two Macro libraries that are part of the factory content: Building Blocks and Classic Modular. Common to both libraries is the idea to make these Macros as easy to use as possible. For that purpose, all relevant parameters within these Macros are already equipped with controls and displays that you see and are able to manipulate on the Panel.

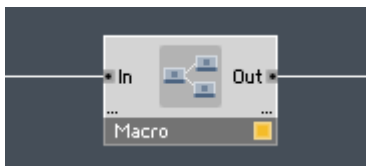


Fig. 7.29 A Macro.

The Classic Modular Macro library consists of Macros that together provide functionality similar to the different components of a modular synthesizer. If you like creating your own modular synthesizers out of basic components without having to think too hard about the

compatibility of the components that you are using then the Classic Modular library is exactly for you. All of the input and output values of the library's Macros are normalized, so within itself the Classic Modular library forms a self-consistent system where all the Macros are compatible with each other.

Whilst the Classic Modular library provides a self-consistent paradigm for the modular synthesizer approach, the Building Blocks Macro library is more focused on supplying ready-made synthesizer parts to speed up your work in a larger variety of situations. For example, you will find a collection of commonly used digital effects ranging from a phaser to a compressor in the Building Blocks library. In contrast, these digital effects are not part of the classic modular synthesizer concept and are therefore not found in the Classic Modular Macro library. Comparing the oscillators of the Classic Modular and the Building Blocks libraries, you see that the Classic Modular library sticks with the oscillators that we know from analogue modular synthesis whereas the Building Block library ventures a bit further with the Formant Osc and Sync+FM Macros. Also, the Building Blocks library gives you quicker access to some of the more complex Modules like the Grain Cloud Module by delivering these Modules in ready-to-use Macros.

Saving Macros

If you have modified a Macro and decide to save it, you can right-click on the Macro and choose the *Save Macro As...* menu entry. This opens your operation system's browser that lets you save content as usual. Make sure you choose your User Path as the save path if you want to be able to access the Macro directly from the “*Insert Macro*” menu. See subsection [↑14.4.2, Directories Tab \(Setting File Paths\)](#) for setting up your User Path. You can also do the same thing when in Panel View. In that case you need to right-click on the Macro frame to find the *Save Macro As...* menu entry.

Modules

Modules in the Primary Level are elementary building blocks out of which you build your Structures. Often the Modules need to be equipped with controls or constant values before they can become functional. For example, an oscillator Module will send a zero-signal unless its “A” (amplitude) input port is connected to a nonzero signal. Depending on the number of input ports and conditions for a particular Module, this makes it more time consuming to work with Modules rather than with Macros from the Macro library that serve the

same purpose. You could insert a Sine Osc Macro instead of a Sine Module into the Structure and basically achieve the same thing. Nevertheless, Modules give you in many cases many more degrees of freedom to explore new ways of building Instruments. See section [↑7.5, Choosing the Right Objects](#) for an overview of the different types of Modules.

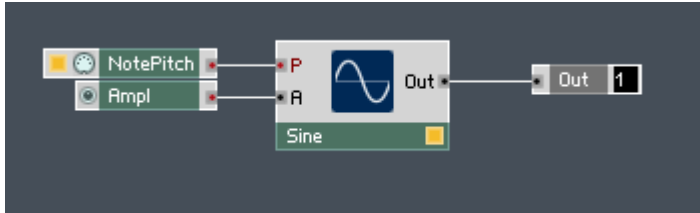


Fig. 7.30 Four Modules.

Core Cells

For most practical purposes other than their ability to be customized, Core Cells are interchangeable with Primary Level Modules. There are some differences between the two, though. One important difference is that there are no panel elements in Core Cells, therefore, you will always need to use Primary Modules to create controls for Core Cells. Another type of Primary Level Module that you will not find among the Core Cells is the sampler. For most other uses you can view the Core Cell as a highly customizable Module. This customization takes place in the Core Level of REAKTOR and is addressed in much detail in the Core Reference. One should be aware though, that there is a slight overhead in CPU load when a Core Cell has been inserted into the Structure. This means that when you are working with the Core Level, it is best to pack as much of your Core Level Structure into as few Core Cells as possible. Furthermore, in the monophonic case or sometimes with a low number of Voices you can expect similar performance for the Primary and Core Level. However, since REAKTOR Core is not SSE optimized, the Primary Level will generally outperform the Core Level when the number of Voices becomes large.

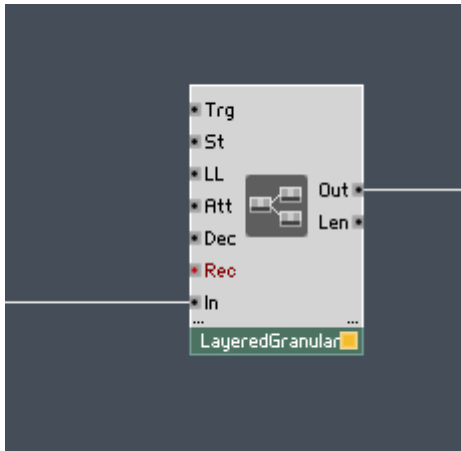


Fig. 7.31 A Core Cell.

7.5.2 Panel Elements

Panel Element Modules add a control or visualization to the Instrument Panel. They allow you to create the Panel interface for your Instrument so that you can control it without opening the Structure. Examples of Panel Elements include faders, knobs, oscilloscopes, level meters, and other powerful interface tools such as the MultiDisplay Module.

7.5.3 Oscillators

Oscillators are the main sound generation tool in synthesizers. Whether you are doing subtractive, FM, or additive synthesis, oscillators are the initial source for your audible signal before you process it into your final sound. From a more technical perspective, oscillators are units that generate a periodic waveform, with a frequency and amplitude that you determine with the values at the corresponding input ports. For some of the more versatile oscillator Modules, the shape of the waveform can also be controlled with external parameters.



In addition to the oscillators you find among the Built-In Modules, there is another set of more customized oscillators in the “Oscillator” folder of the Core Cells factory library.

7.5.4 Samplers

Another common source for Audio signals is the sampler. Sampler Modules create a signal from a sample that you load in the form of a .wav, .aif, or .aiff file into a Sample Map file. Beyond the standard Sampler Module, in this category you will find advanced tools like Beat Loop and Grain Cloud to build anything from sample-based grooveboxes to highly customized granular synths.



For more information regarding Sample Maps, please refer to chapter [↑6, Sample Map Editor](#). For more detailed information addressing the use of sampler Modules, please refer to chapter [↑10, Working with Sampler Modules](#).

7.5.5 Filters

Filters are the basic tools with which you shape the spectrum of a signal. In this category you will find a wide variety of filters, among them also the well known low-pass, band-pass, and high-pass filters. Roughly speaking, low-pass filters only let frequencies below the cutoff frequency pass, band-pass only let frequencies around the cutoff frequency pass, and high-pass filters only let frequencies above the cutoff frequency pass. If you want to build your custom equalizer (EQ), then this is the category that you should turn to. Also, if you are interested in modal synthesis, the Modal Bank Module is found in among the filters too.



In addition to the filters you find among the Built-In Modules, there is another set of more customized EQs and filters in the “EQ” and “VCF” folders of the Core Cells factory library.

7.5.6 Audio Modifiers

Audio Modifiers are Modules which are explicitly used to shape, saturate, or generally modify Audio signals. Shapers and saturators are often used to add or in some cases change the harmonic content of signals. This functionality is especially desirable in distortion type effects. In this category you will also find the Audio Table Module which can have a variety of functions from serving as a custom sample playback unit to being a wavetable oscillator. The Table Modules are discussed in detail in chapter [↑11, Working with Table Modules](#).



In addition to the Audio Modifiers you find among the Built-In Modules, there is another set of shapers in the “Audio Shaper” folder of the Core Cells factory library. Also, the “Granular” folder provides Core Cells that apply a real-time granular effect to an incoming Audio signal.

7.5.7 Delay

The Delay Modules are the essential building blocks for delay-based effects such as delay, chorus, reverb, and comb filters. Delay Modules take an incoming signal and delay it by a certain time interval that can be specified at the corresponding input port. Multi-tap delays allow you to conveniently work with multiple delays simultaneously.



In addition to the delays you find among the Built-In Modules, there is another set of delays in the “Delay” folder of the Core Cells factory library.

7.5.8 Sequencers

Sequencer Modules are the tools that you need to quickly build step sequencer Instruments. Step sequencers send repeating sequences of signals at fixed time intervals, usually synchronized with the time signature of a piece of music. The Sequencer Modules have input ports for the step, so that you can specify the value for each step separately

7.5.9 Envelopes, LFOs (Control)

A widely spread method for modulating synthesizer parameters such as amplitude, pitch, and filter cutoff frequency utilizes envelopes and LFOs.

An envelope is a set of exponential curves connected by breakpoints. The characteristic parameters that determine the shape of the envelope are the times that each exponential curve takes to reach its value at the next breakpoint. An example of such a parameter is the attack time or the time that an envelope takes to go from zero to its maximum value. Envelopes are often applied as modulating signals for the oscillator amplitudes in synthesizers.

An LFO or low frequency oscillator is a generator of a slowly oscillating periodic signal. Musical examples of applying LFOs to amplitude and/or frequency are the tremolo and vibrato.



In addition to the LFO and Envelope Modules you find among the Built-In Modules, there is another set of delays in the “Control” folder of the Core Cells factory library.

7.5.10 Math

Math Modules form the backbone of every REAKTOR Structure. Very simple but essential operations can easily be implemented using Math Modules. For example, mixing two signals together (at equal amplitude) entails adding the two signals using the Add Module. Also, setting the amplitude of a signal can be done by multiplying the signal by a constant value. The multiplication is done with the Multiply Module and the constant value usually is delivered by the Constant Module or a Knob of Fader Module. This is not to say that the amplitude of a signal cannot be determined by something completely different, like the output signal of an envelope.

7.5.11 MIDI In, MIDI Out

The MIDI In and MIDI Out Modules let your Instruments communicate with the outside world, your VST host, or other Instruments using the MIDI protocol. Use the NotePitch Module if you want to retrieve the pitch that you are playing with your MIDI keyboard or that your VST host is sending to REAKTOR. The Gate Module sends an Event when a “note on” message is sent from the MIDI interface that you are using. This can, for example, be used to conveniently control envelopes or trigger samples. Of course you can customize the way that your Instrument communicates using the MIDI protocol and the examples brought here serve to give you a rough idea of the purpose of MIDI Modules. See section [↑13.2, Setting up MIDI Connections](#) for more information on the MIDI interface of REAKTOR Instruments.

7.5.12 Signal Path

Signal Path Modules serve to route your signals in REAKTOR, regardless if they are Audio or Event signals. If you want to create a simple crossfader between two signals, then using the Mixer Modules in this category is the quickest way to achieve that. Another example includes the Distributor Module which can distribute one signal to one or two of several output ports, depending on the value at its “position” input port. Similarly, a Selector Module takes a signal from one or two of several input ports and forwards it to its output port, depending on the value at its “position” input port.

7.5.13 Event Processing

The Event Processing category holds Modules that are necessary to work with the discrete nature of Events and Modules meant to work at the Control Rate. In this category you will, among others find Modules that count Events and their values, Modules that shape control signals, Modules for logic operations, Modules that specifically deal with the order of Events, and Modules that route single Events (as opposed to the Distributor and Selector Modules, which route values). The Event Table Module, which can have a variety of functions from serving as a sequencer unit to being a buffer table for copy/paste operations, is also found in this category. The Table Modules are discussed in detail in chapter [↑11, Working with Table Modules](#).

7.5.14 Auxiliary

The Auxiliary Modules provide miscellaneous functionality such as saving values with Snapshots (Snapshot Module), converting between Event and Audio signals (like Audio and Event Smoother Modules), working with Polyphony (like Voice Combiner Modules), and retrieving information on global settings (like Tempo Info and Voice Info Modules). These Modules are usually, but not always, connected to more advanced operations in RE-AKTOR.

7.5.15 Terminal

Terminal Modules include terminals input and output ports of Instruments and Macros, but also ports to communicate with the OSC and Internal Connection (IC) protocols. See sections [↑14.2, Using the Internal Connection Protocol](#) and [↑13.4, Setting up OSC Connections](#) for information on the IC and OSC protocols, respectively.

7.6 Object Properties

Each Object in REAKTOR has a set of Properties that can be changed in the Properties View. To switch to Properties View for an Object, click on the Properties button in the Sidepane as mentioned in subsection [↑3.1.1, Sidepane](#) or double-click on the Object (but not the Object icon in the case of Instruments, Macros, or Core Cells). When in Properties View, the Properties of the selected Object are displayed and regardless of Object type, there are four Properties pages which can be accessed by clicking on the corresponding

button. These four pages will be described in more detail below however, since each Object has its own unique set of Properties, then the specific content of these pages might vary. For some Objects a few pages might even be empty.



Fig. 7.32 The Properties tab



You can also access an Object's Properties from the Panel View by right-clicking on the Object on the Panel and then left-clicking on the “...” Properties menu entry where the three dots stand for whatever the Object Label is.

7.6.1 Function Page

You can access the Function page by clicking the **Function** button while being in Properties View. This page displays settings pertaining to the functionality of the selected Object. Usually this includes checkboxes for general settings like Mute and Mono but often there are additional options specific to an object. The Function page of a Sampler Module, for example, has the button to select a Sample Map and a menu to select the quality of the sample playback. Other Modules, like the Distribution and Selector Modules, have value fields to choose the minimum number of output or input ports, respectively.



Fig. 7.33 The Function button

7.6.2 Info Page

Access the Info page by clicking on the **Info** button icon when in Properties View. The Info page displays text that gives an overview of the functionality of the Module. It is the text that appears in the Tooltip upon a mouse-over when Hints are activated (see subsection [↑3.5, Other Common Operations](#) on activating Hints). Often the text is editable as is the case for Instruments and Macros. It is convenient to document your Structure or hints regarding the operation of your Instrument using the Info page.



Fig. 7.34 The Info button

7.6.3 View Page

In order to open the View page, click on the **View** button while being in Properties View. If the selected Object has a Panel representation (as is the case for Instruments, Macros, and Panel Element Modules), its appearance is determined by the settings in this page. For Instruments, the settings include Panel colors and background pictures. For Macros you can specify Macro visibility on the Panel including the appearance of frames and for Panel Element Modules like knobs and faders you can specify size, control type, and even custom skins. Most of these settings can be separately specified for the A and B Panel Views so you can create two user interfaces for different ways of using the Instrument. Furthermore, for Instruments and Macros you can specify the Structure Icon in this page.



Fig. 7.35 The View button

7.6.4 Connections Page

The Connections page can be viewed by clicking on the **Connect** button while being in Properties View. This page holds Properties regarding connections to the external world and within the Instrument. Here you can specify MIDI settings like assigning MIDI CC values to controls. Also, the Automation ID Number for Panel Elements is set here so that this control can be automated by your host. Please read section [↑14.3, Automation ID Management](#) for more information on Automation ID Numbers. In addition, you find the settings for the selected Object's connections using the IC and OSC protocols in this Tab. See sections [↑14.2, Using the Internal Connection Protocol](#) and [↑13.4, Setting up OSC Connections](#) for information on the IC and OSC protocols, respectively.



Fig. 7.36 The Connect button

8 Panel Editing

The Instrument Panel is the point at which you will most likely interact with your Instrument when making music. In order to successfully modify and build Instruments, you need a good grasp of how to change the Instrument Panel and the Panel Elements. In this chapter you will learn all about the Panel; the topics discussed range from changing the Panel layout to specifying the appearance of your controls to importing custom skins for them.

8.1 Setting Panel Layout Properties

The first step to modifying the appearance of your Instruments is changing the general layout of your Instruments and Panel Elements. In this section, you will become familiar not only with changing the arrangement of Instruments and Panel Elements, but also with the A and B Panel Views which let you simultaneously specify two different layouts for your Instrument Panel.

8.1.1 The Panel and Instrument Layouts

Changing the Panel Layout

You can change the location of the Instruments on the Panel just by clicking and dragging the Instrument Header. If the *Settings > Automatic Layout* menu entry has been selected, the Instruments snap into place so that they are aligned with the Panel borders and neighboring Instruments, if they are present. If the automatic layout option is deactivated, the Instruments can hold arbitrary positions; they can even be placed on top of each other.

Changing the Instrument Layout

In order to change the layout of the Panel Elements on the Instrument Panel, you need to first unlock the Panel. You can achieve this by clicking on the **Panel Lock** button or with the help of the keyboard shortcut Ctrl+P in Windows (Cmd+P for Mac OS X). Once you have unlocked the Panel, you can move the various Panel Elements around by clicking and dragging them with the mouse. Objects that are in Macros that have a visible frame around them can be moved around together by clicking and dragging the Macro's frame. You can

also select several Instruments by clicking on an empty area of the Instrument Panel and dragging a box around the Objects you want to select. Then, click on a selected Object and drag it (along with the other selected Objects) to the desired position on the Panel. When you are done with your arrangement, you must remember to lock the Panel again before you can use the Panel Elements as controls. This is done the same way that you unlocked the Panel.



Fig. 8.1 The Panel Lock button



You can quickly select all Objects on the Panel by right-clicking on the Instrument Panel and then left-clicking on the *Select All* menu entry. To access this menu entry, the Panel needs to be unlocked.

Protecting the Instrument Layout

If you want to keep the Panel layout fixed, click on the *Settings > Protected* menu entry. This causes the *Panel Lock* button to gray out. Also, you cannot change the locations of the Instruments relative to each other. Clicking on the same menu entry again deactivates the Protected Mode.

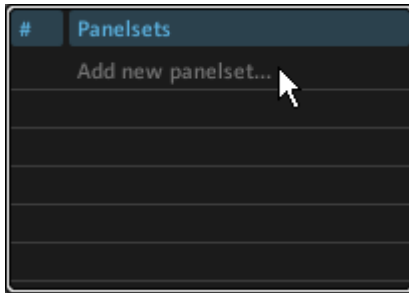
8.1.2 Working with Panelsets

Panelsets allow you to create layouts of Instruments. You can choose which Instruments are shown in a particular Panelset. To get a feel for what you can do with Panelsets, follow these instructions:

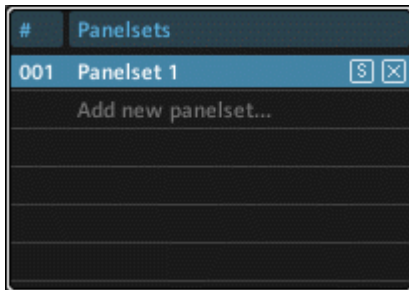
1. Load an Ensemble with several Instruments, like Junatik, as shown in subsection 3.3.1. You'll find Junatik in the folder *Factory Library > Classics > Synthesizer*.
2. Next, click on the *Panelsets* tab in the Sidepane.



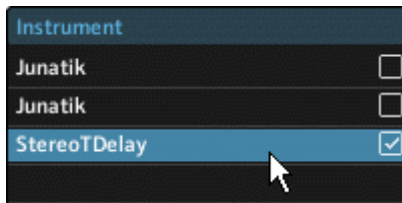
3. To create a Panelset out of the current layout of the two Instruments, click on the *Add New panelset* button.



4. You should now see the first Panelset appear in the following form:



5. Now rearrange your Instruments on the Panel. In the case of Junatik, place the delay Instrument above the Synthesizer Instrument.
6. Create a second Panelset by clicking on the Add New panelset... button.
7. With the second Panelset selected (it should be light blue), click on the first Panelset button. The Panel should change back to how it appeared when you first loaded the Ensemble.
8. Click on the second Panelset button again to show the rearranged Panel. Note that creating a new Panelset saves the Panel arrangement as that new Panelset.
9. Now click once more on the Add new panelset... button to create a third Panelset. The entry corresponding to the third Panelset should be light blue, indicating that the third Panelset has been selected.
10. With the Panelset Instrument buttons on the bottom of the screen you can determine which Instruments are visible on the Panel. Click on the StereoTDelay Panel Instrument button.







11. Now only the StereoTDelay Instrument is visible on the Panel. Press the StereoTDelay Panel Instrument button again to show the previous Instrument layout.
12. You can hide individual Instruments by disengaging the checkmark next to the corresponding Instrument's entry in the Panelsets list. Do this for the delay Instrument.
13. Now the StereoTDelay Instrument should have disappeared from the Panel. You can show and hide as many Instruments as you like, even an empty Panel is allowed. Engage the checkbox next to the StereoTDelay entry. This shows the delay Instrument again.
14. Hide the Junatik synthesizer by clicking on the corresponding checkbox. Now you'll learn how to apply this layout to the first Panelset.
15. To save the current Panel layout as the first Panelset, click on the Save Panelset button next to the corresponding Panelset entry (labeled with an "S").



16. To delete a Panelset, click on the Delete Panelset button (labeled with an “x”) next to the corresponding Panelset. This is it, you have just become familiar with all the functions regarding Panelsets in REAKTOR!



-  Use the keyboard shortcut Ctrl+Shift+# in Windows (Cmd+Shift+# in Mac OS X), **where # represents a number**, to store a Panelset to that Panelset slot. Note that you can also store a Panelset using the menu entry corresponding to the target Panelset slot in the *Settings > Store Panelset* submenu.
-  Use the keyboard shortcut Ctrl+# in Windows (Cmd+# in Mac OS X), **where # represents a number**, to recall a Panelset to that Panelset slot. Note that you can also recall a Panelset using the menu entry corresponding to the desired Panelset in the *Settings > Recall Panelset* submenu.
-  Use the keyboard shortcut Ctrl+Shift+# in Windows (Cmd+Shift+# in Mac OS X), **where # represents a number**, to store a Panelset to that Panelset slot. Note that you can also store a Panelset using the menu entry corresponding to the target Panelset slot in the *Settings > Store Panelset* submenu.
-  Use the keyboard shortcut Ctrl+# in Windows (Cmd+# in Mac OS X), **where # represents a number**, to recall a Panelset to that Panelset slot. Note that you can also recall a Panelset using the menu entry corresponding to the desired Panelset in the *Settings > Recall Panelset* submenu.

8.1.3 Working with Panel Views

A and B Panel Views and Minimizing

REAKTOR allows for each Instrument to have two separate Panel layouts, A and B. Changing the layout of the Panel while in one Panel View will leave the other layout unchanged. You can switch between the A and B Panel Views by clicking on the A View and B View buttons in the Instrument Header or by right-clicking on the Instrument Panel and clicking on the *View B* or *View A* menu entry. The A View and B View buttons in the Instrument Header are labeled with an “A” and “B”, respectively.



Fig. 8.2 The Instrument Panel View buttons

To minimize the whole Instrument Panel leaving only the Instrument Header visible, click on the Minimize View button (labeled with a “-”) in the Instrument Header. Alternatively you can right-click on the Instrument Panel and then click on the *Minimize View* menu entry.

Different Appearance Properties for A and B Panel Views

For most Objects which have settings in the View page of the Object Properties you can specify these settings separately for the A and B Panel Views. For example, you can choose a Control Module to appear as a knob in the A Panel View but as a fader in the B Panel View. To try this out by performing the following steps:

1. First create a new Ensemble as shown in subsection [↑3.3.5, Creating a New Ensemble](#).
2. Unlock the Instrument Panel by clicking on the *Panel Lock* button.



3. Right-click on the Instrument Panel background and click on the *Insert Built-In Module > Fader* menu entry.
4. Lock the Instrument Panel by clicking on the *Panel Lock* button.
5. Open the Properties by double-clicking on the fader.

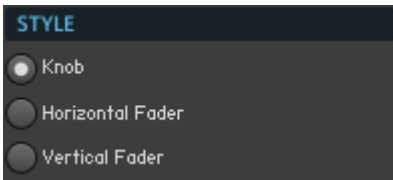
6. Click on the **View** button.



7. You should now see the **Panel View** tabs. Click on the **A Panel View** tab.



8. Adjust the appearance settings of the fader by turning it into a knob for the **A Panel View**. Do this by clicking on the **Knob** radio button.



9. If you are in the **A Panel View**, you should see on the **Instrument Panel** that the fader turned into a knob.
10. Click on the **B Panel View** button. You should now see that the knob has turned into a fader. This is exactly the point of this feature: to allow you to specify two different ways the **Instrument Panel** appears on your screen.
11. If you want the appearance settings that you are applying to an **Object** to be the same for both **Panel Views**, then click on the **AB Panel View** tab (labeled as “AB”) before you do your editing.



Fig. 8.3 AB Panel View tab

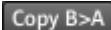
Copying Appearance Properties between A and B Panel Views

Let's say that you have specified different appearance settings for a **Panel Element** in the **A** and **B Panel Views**. After a bit of consideration, you decide that you want to discard the settings for the **B View** and use the settings that you specified for the **A View** also in the **B View**. So that you don't have to reproduce all of the settings for the **B View** manually, RE-AKTOR allows you to copy them from one **View** to the other. Here's how to do this:

1. Make sure you have a setup as in step 10 in the previous instruction list: a control is a knob in the A View and a fader in the B View and the View page of the corresponding Panel Element is opened in the Sidebar.
2. If you want the same settings for the B View as for the A View, that is, both Views to display a knob, just click on the **Copy A>B** button. Now all the appearance settings for the control that are in A View have been applied to the B View; both are knobs. Note that not only the control Style setting, but all the other settings like control Size have been copied from the A View.

A dark rectangular button with the text "Copy A>B" in white.

3. If you remember that you actually wanted the settings from the B View applied to the A View, then just undo by pressing Ctrl+Z for Windows (or Cmd +Z for Mac OS X) and then clicking on the **Copy B>A** button instead. Now all the appearance settings for the control that are in B View have been applied to the A View; both are faders.

A dark rectangular button with the text "Copy B>A" in white.

Copying Whole Layouts between A and B Panel Views

1. Often when you are modifying or building Instruments, you are only thinking about one Panel View, say, the A Panel View. After you have finished your layout for the A Panel View you remember that you have another View to work with but since you ignored setting the layout for the B Panel View, it is probably a mess by then, sometimes all the Panel Elements are stacked on top of each other. In that case it is easier to start off with the same layout for the B Panel View as for the A Panel View and start modifying it from there. REAKTOR lets you do this very easily without having to manually duplicate the layout and all the appearance settings for each set of Panel Elements. Here's how:
2. First create a new Ensemble as shown in subsection [↑3.3.5, Creating a New Ensemble](#).
3. Unlock the Instrument Panel by clicking on the **Panel Lock** button.



4. Add a few Panel Elements to your Instrument by right-clicking on the Instrument Panel background and choosing from the *Insert Built-In Module* menu.
5. Now, being in the A Panel View, rearrange the Panel Elements into a definite order by clicking and dragging them to their respective places.
6. Now click on the B Panel View button in the Instrument Header. You should now see that while ordering the A Panel layout, the B Panel View remained unordered.



7. Right-click on the Instrument Panel and choose the *Copy Controls From Other Panel* menu entry. The B Panel View should now look like the A Panel View that you ordered. Note that the panel has to be unlocked in order for you to be able to access that menu entry.



You could have also copied the layout from the A Panel View to the B Panel View without entering the B Panel View. While in the A Panel View, right-clicking on the Instrument Panel and clicking on the *Copy Controls To Other Panel* menu entry would have had the same effect.

8.2 Inserting and Modifying Panel Controls

This section will discuss the Modules that have a Panel representation and which you will be using the most to control your Instrument from the Panel. These Modules include faders, knobs, buttons, lists, and switches. In addition, you will become familiar with the different Properties that are relevant to the appearance and functionality of these Modules.

8.2.1 Faders and Knobs

Faders and knobs are Panel Elements which let you control the values that are sent from the Module output port in the Structure. The screenshot below shows the appearance of faders and knobs on the Instrument Panel.



Although the fader and the knob are essentially the same Module, there are separate menu entries for them in the *Built-In Module > Panel* menu.

Setting the Range and Resolution of Output Values

Often when you are modifying a Structure, the process entails that you create new controls which appear on the Instrument Panel. Since these Modules are meant to let you control new aspects of the Structure, it naturally leads to the question, what values are delivered from the output port of the control and at which resolution. To specify these values for a control, follow the following instructions:



You can conveniently create a fader or a knob, depending on the situation, that is already connected to an input port of your choice. For this you must right-click on the input port and choose the *Create Control* menu entry.

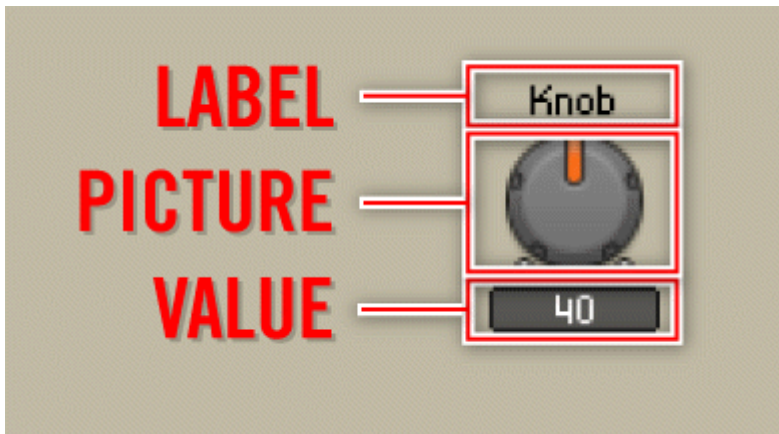
Choosing the Control Appearance

The fader and the knob are essentially identical; you can switch between them and modify other visual properties of the Module in its View page. For a brief tutorial on what you can do, perform the following steps:

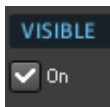
1. As in the previous list, make sure you have an Instrument open that has conventional knobs on its Panel (like the one at the beginning of this chapter). Remember that you can add knobs and faders directly to ports.
2. Double-click on a knob to open up its Properties.
3. Click on the *View* button.



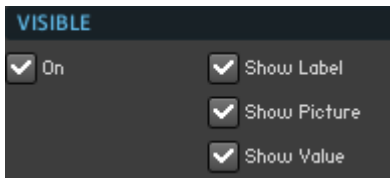
4. First, let's look at how to change the visibility of the different parts of a knob or fader. Controls are split up into three parts: the label, the picture, and the value.



5. The whole control can be made invisible on the Panel by engaging the `On` checkbox. Disengage the checkbox and see the control disappear. Engage the checkbox again to make the control appear (there's more to show in the steps to come).

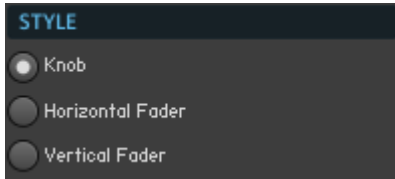


6. Now let's turn our attention to hiding and showing the three parts of the control separately. This can be done with the help of the checkboxes (see screenshot below). Disengage the `Show Label` checkbox to see the control Label disappear.

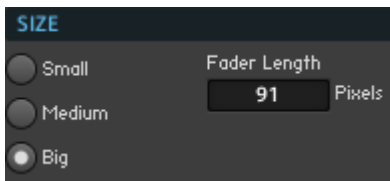


7. Now, disengage the `Show Picture` checkbox to see the animated Picture of the control disappear. Notice that you can still control the fader or knob by clicking and dragging its Value field.
8. Last, you can even make the Value field disappear by disengaging the `Show Value` checkbox. Now disengaging the `On` checkbox has no visible effect.

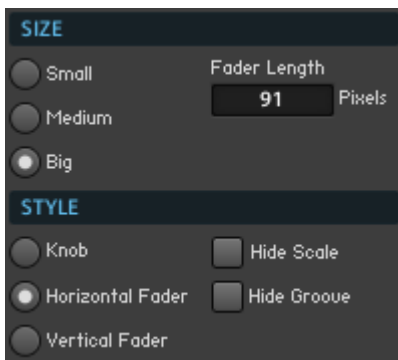
9. Now you will see how to change the control type using the Style selector. Click on the different radio buttons to see the three different control styles. In the end, though, click on the **Knob** radio button.



10. Get familiar with the different sizes for the controls by clicking on the different selector boxes of the Size selector (see below). Make sure you check out how the Size setting affects the appearance of both knobs and faders.



11. Switch to a fader control type using the Style radio button selector.
12. You can change the length of the fader in pixels using the **Fader Length** edit field (see below). Experiment with different values for Fader Length to get a feel for this parameter.



13. You can also hide the scale and the grooves of the fader by using the `Hide Scale` and `Hide Groove` checkboxes. Try it out and see what suits your taste for Instrument interfaces best.
14. Last it is worth mentioning that you can use custom skins for your controls. You can browse for the skin files with the `Skin Bitmap` drop-down menu (see below).



8.2.2 Buttons

A button is a Panel Element which lets you send Events in the form of toggle, trigger, and gate signals. An Event is sent each time the button is pressed. The figure below shows the appearance of a button on the Instrument Panel.



Fig. 8.4 A button on the Instrument Panel.

Setting the Button Mode and Output Values

You can specify two values for buttons, the `On Value` and the `Off Value`. When each of these values is sent depends on the button's `Mode` setting. The `Mode` radio button selector in the `Function` page offers three possibilities: `Toggle`, `Trigger`, and `Gate`.

- **Toggle:** the button has two states, “On” and “Off”. When you turn the button on, an Event with the value specified by the `On Value` edit field is sent. When you turn the button off, an Event with the value specified by the `Off Value` edit field is sent. The `Toggle Mode` is useful for controlling signal flow. A “bypass” button that turns an effect on and off is most likely in `Toggle Mode`.
- **Trigger:** when you click the button, a trigger signal in the form of an Event is sent. The Event carries the value specified by the `On Value` edit field. This `Mode` is useful for triggering samples.

- **Gate:** similar to the Toggle Mode, the button has two states, “On” and “Off”. When you press (and hold) the button, an Event with the value specified by the On Value edit field is sent. The moment that you release the button, an Event with the value specified by the Off Value edit field is sent. With a button in this Mode you could send gate signals to envelopes.

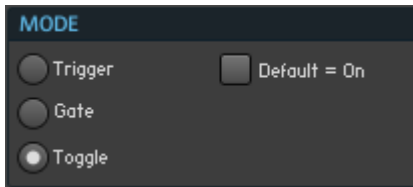


Fig. 8.1 The Mode radio button selector in a button's Function page

To specify the On and Off Values and the Mode of a button, read the following instructions:

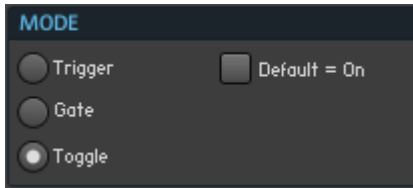
1. Double-click on a button to open up its Properties.
2. Click on the **Function** button.



3. To change the On Value and Off Values, enter the desired numbers in the **On Value** and **Off Value** edit fields, respectively. The figure below shows these edit fields. You just need to double-click the corresponding edit field and type in the new value. Note that the On Value can be less than the Off Value.



4. Change the Mode of the button to Trigger by clicking on the **Trigger** radio button of the Mode radio button selector shown below.



5. If the Default = On checkbox is disengaged, then the default state of the button is “Off”. This means that if you click on the *Recall default values* menu entry of the Edit Bank drop-down menu in the Snapshot tab, the button is set to its “Off” state. Also, when you right-click the button in Panel View and then left-click the *Set to Default* menu entry, the button is set to its “Off” state. The setting of the Default = On checkbox is only significant for the Toggle Mode.

Choosing the Button Appearance

Like for the fader and knob controls, you can change the visibility and size of the button. To see how to do this, follow these instructions:



If you have already read the text under the header “Choosing the Control Appearance”, then you can skip this part.

8.2.3 Switches

The Switch Module is a special type of Panel Element. Its main functionality lies in the Structure as an element that forwards the signal from one of its input ports to its output port. Which signal it forwards, is chosen on the Instrument Panel by the Switch Module's Panel representation. This subsection explores the different aspects of the Structure functionality and Panel appearance of the Switch Module.

Routing Signals Using a Switch

In the following example you are shown how to use the Switch Module to choose which signals are forwarded to an Output Terminal. Let's say we have a Macro that is applied to a signal as an effect. We call the processed signal the “wet” signal and the unprocessed signal the “dry” signal. See the screenshot below for an exemplary structure.

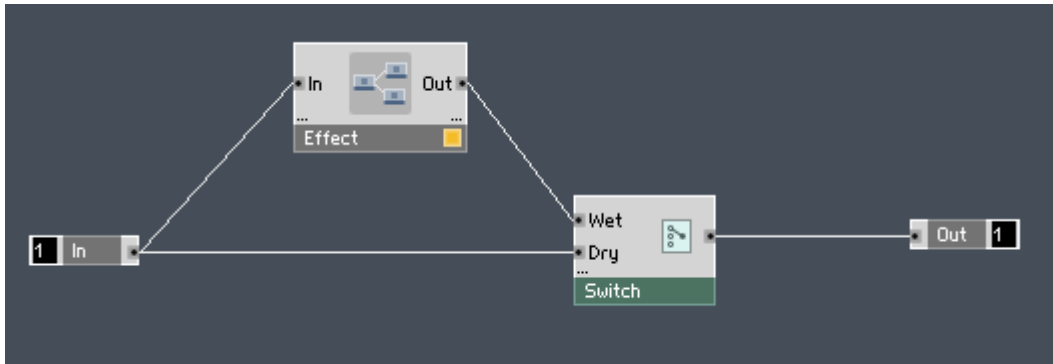


Fig. 8.5 Here a Switch Module is used to forward either the “wet” signal or the “dry” signal to the Output Terminal.

If we go to the Instrument Panel, the Switch can be set to have a Panel representation as shown in the figure below. The Switch Module lets you choose between the different signals arriving at its input ports. Here the input ports have been labeled “Wet” and “Dry” and these names also appear on the Instrument Panel. If you click on the “Wet” button, then the signal arriving at the “Wet” input port of the Switch Module is routed to its output port. If you press the “Dry” button, then the signal arriving at the “Dry” input port of the Switch Module is routed to its output port.



Fig. 8.6 A possible Panel representation of the Switch Module.



An important case is when you have Event signals connected to the input ports of the Switch Module. “Switching” to one of these signals causes the value that was sent last to that particular input port to be sent from the output port of the Switch Module in the form of an Event.

Switch vs. Router vs. Selector

In REAKTOR, Switches, Routers, and Selectors can all be used to select which signals are forwarded to another part of the Structure. What is the difference between these Modules?

- The two most important differences between these Modules are the following. The first one is a Module's compatibility with the Audio and Event signal types. Whereas the Selector and Switch Modules are compatible with either Audio or Event signals, the Router Module is only meant for Event signals. Second, a property that is unique to the Switch Module is its ability to mute all Objects lying upstream from its “inactive” input ports. However, these Objects will not be muted if they are part of other signal flows that would make them active. This feature of muting the Objects that are connected to inactive input ports can be useful for saving CPU resources, but can also be disabled. See the “Activating All Input Ports” header below on how to deactivate this feature.
- The next difference between the Switch, Router, and Selector Modules is their behavior in response to switching from one Event input signal to another. For Switch and Selector Modules, switching input ports triggers an Event at the output port. At the moment of the switch, the value that was last sent to the “new” input port is sent from the Module's output port. When you switch between input signals of a Router Module, nothing happens unless an Event arrives at the selected input port. The Router Module only forwards incoming Events.
- Another aspect that differentiates the Switch Module from the other two Modules is the fact that the routing of the Switch Module can only be controlled from its Panel representation whereas the Selector and Router Modules have a “Pos” (position) input port which allows you to control their routing by sending a value to that input port. However, functionality and Panel representation similar to the Switch Module can be achieved by connecting a List Module to the “Pos” input port of these Modules.

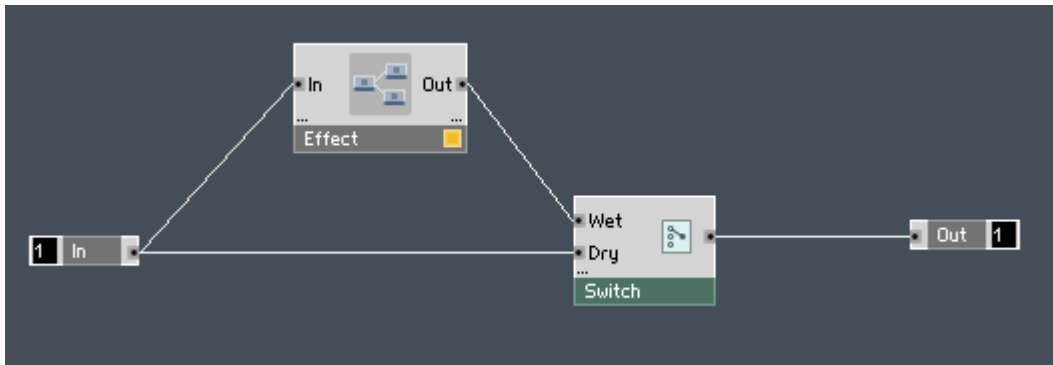
Choosing the Appearance for Lists and Switches

The View pages for the Switch and List Modules are identical. Although subsection [↑8.2.4, Lists](#) constitutes a separate subsection for the List Module, the following text serves as a guide for changing the Panel representation of the List Module as well. For the sake of brevity, the contents of its View page will not be repeated in subsection [↑8.2.4, Lists](#).

! Note that even though the steps to changing the Panel representation of the Switch Module also apply for the List Module, there is one important difference. Instead of the Panel representation deriving its list of elements (menu entries or buttons) from the list of input ports, as is the case for the Switch Module, the List Module's list of elements is created in the Module's Function page. See subsection [8.2.4, Lists](#) for more information on this.

The List and Switch Modules have several variations for their appearance on the Panel. In addition to changing the visibility and size of the Label you can also choose between different Panel representations including buttons, drop-down menus, and text panels. The following list of instructions guides you through a small tutorial on the Panel representation of the Switch (and List) Module:

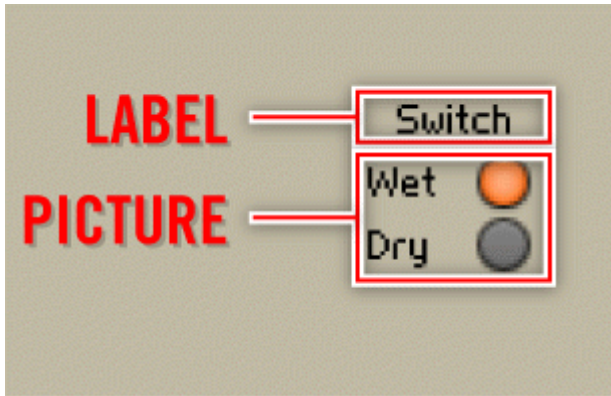
1. Create a simple Instrument that implements a Switch Module in its Structure.



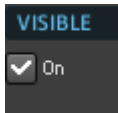
2. Double-click on the Switch Module to open up its Properties.
3. Click on the **View** button. Make sure you can see the Panel representation of the Switch Module.

VIEW

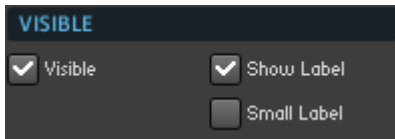
4. Switches are split up into two parts: the Label and the Picture.



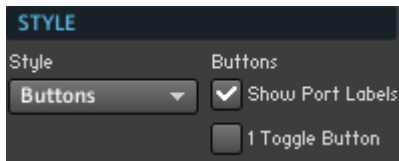
5. The whole representation can be made invisible on the Panel by engaging the On checkbox. Disengage the checkbox to see the Panel representation disappear. Engage the checkbox again to make the Panel representation appear.



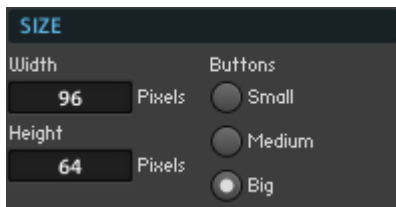
6. Similar to the previous step, the Label can be made invisible by disengaging the Show Label checkbox. Disengage the Show Label checkbox to see the Label disappear.



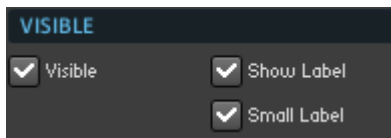
7. Engage the Show Label checkbox to make the Label appear again on the Panel. Next make the Labels for the ports "Wet" and "Dry" disappear by disengaging the Show Port Labels checkbox. This checkbox only has an effect when the Style of the Switch Module is set to Buttons.



8. Similar to the Button Module, you can change the size of the buttons of the Switch Module using the Size selector (see figure below). Click on the different selector boxes to see the three different sizes of the buttons Panel representation of the Switch Module.



9. Next, reduce the size of the Label by engaging the **Small Label** checkbox (see figure below). Please note that for the Buttons Panel representation the size of the buttons is automatically changed when using this checkbox. For the other Panel representations of the Switch Module, only the size of the Label is affected.



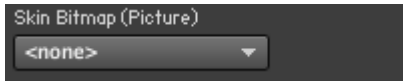
10. Now let's look at the different Panel representations of the Switch Module. REAKTOR gives you four different possibilities (see screenshot below).



11. Change the Panel representation to Text Panel by choosing the *Text Panel* menu entry from the Style menu.



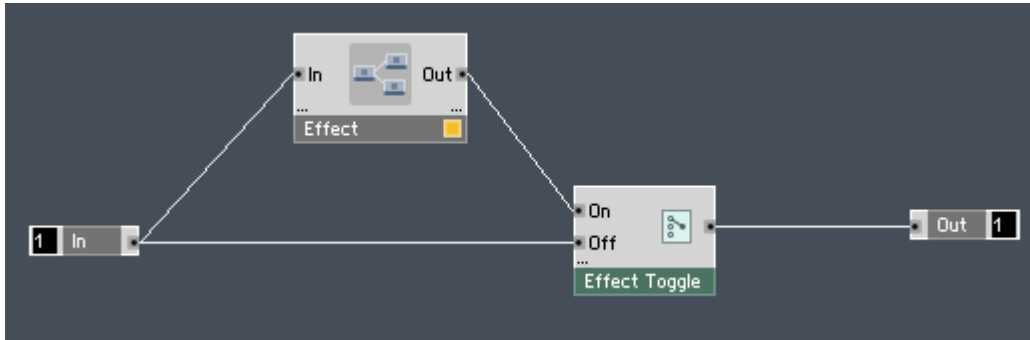
12. Next, change the size of the Text Panel representation of the Switch Module by double-clicking on the Width and Height edit fields and typing in new values. Note that the size dimensions have no effect on the Buttons representation. For the Spin and Menu representations only the Width edit field has an effect.
13. Lastly, you can use custom skins for your Switch. You can browser for the skin files with the Skin Bitmap drop-down menu.



Using the Switch as an On/Off Button

A common use for the Switch Module is to toggle certain parts of the Structure on and off. Here's how to get this functionality from a Switch Module:

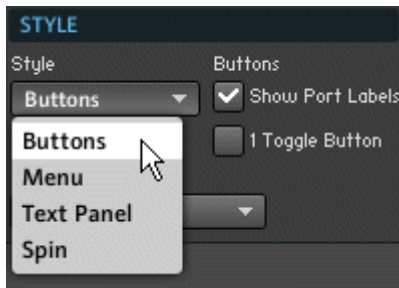
1. Create a Structure such as that shown in the screenshot below. It is basically the same Structure we have been looking at in the previous subsections, but this time the input ports of the Switch Module have been named “On” and “Off” and the Switch itself is called “Effect Toggle”.



2. Double-click on the Switch Module to open up its Properties.
3. Click on the **View** button. Make sure you can see the Panel representation of the Switch Module.



4. Change the Panel representation of the Switch Module to Buttons using the **Style** drop-down menu.



- When looking at the Panel, you should see two buttons: one labeled “On” and the other labeled “Off”. Switching between two buttons to turn the effect on and off is often inconvenient and takes up space. Let's make the two buttons into one button that you can toggle on and off. For this, engage the 1 Toggle Button checkbox.



- Now the Panel representation of your Switch Module should look like in the screenshot below. Toggling the button on and off switches between the signals from the first and second input ports (counting from the top).



Fig. 8.7 With the 1 Toggle Button checkbox engaged, the Buttons Panel representation of a Switch Module becomes just one toggle button.

Activating All Input Ports

The Switch Module has the unique property to mute all Objects lying upstream from its “inactive” input ports. For example, if a Sine Module is connected to the Switch Module's first input port and a Triangle Module is connected to its second input port, choosing the first input port will deactivate the Triangle Module. However, if the Triangle Module's output is connected to another Module that is active, the Triangle Module will not be muted. This feature of muting the Objects that are connected to inactive input ports can be useful for saving CPU resources — in this case the Triangle Module would not use up CPU re-

sources when the output signal of the Object is not used. Nevertheless, sometimes it is desirable to have the Objects turned on regardless of the state of the Switch Module. To achieve this, engage the *Always Active* checkbox (see below). Note that this will activate all Objects lying upstream from the Switch Module, including the Switch Module itself, regardless if the Switch Module's output port is actually connected to any active Objects or not.

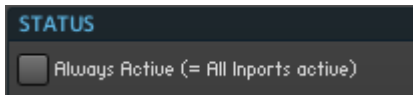


Fig. 8.8 To activate all Modules lying upstream from the Switch Module, engage the Always Active checkbox. You will find it in the Function page of the Switch Module's Properties.

Menu Entry for Selecting No Input Ports

Sometimes you don't want any of the input ports of the Switch Module to be forwarded to its output port. This could be the case when you are mixing a signal that has been processed with an insert effect to the dry signal and decide not to use the processed signal. In this case you should engage the *Enable Switch Off* checkbox in the Function page of the Switch Module's Properties (see below). This adds the *Off* menu entry to the Menu, Text Panel, and Spin Panel representations of the Switch Module. Choosing the *Off* menu entry causes none of the input signals of the Switch Module to be forwarded to its output port.

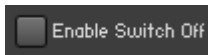


Fig. 8.9 Engaging the Enable Switch Off checkbox adds the Off menu entry to the Panel representation of your Switch Module. With this you can choose not to forward any input signals to the output port of the Switch Module.

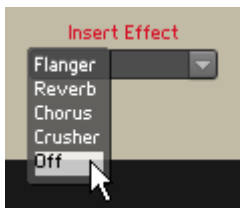


Fig. 8.10 When the Enable Switch Off checkbox is engaged, the Off menu entry is added to the menu.

Setting the Default Value and Mouse Resolution for the Switch and List Modules

If you have chosen the Spin Panel representation for the Switch or List Module, you can switch between input signals or values, respectively, by clicking and dragging the Spin menu. For this, you might want to change the sensitivity of the Spin menu in respect to your mouse movements. For very low sensitivity, set the **Mouse Reso** edit field in the Function page of the Switch Module's Properties to a high value, perhaps "100". For high sensitivity, you might set the **Mouse Reso** edit field to "1". See figure below for a screenshot of the **Mouse Reso** edit field.

You can also set the Default input port of the Switch Module which corresponds to the Default Value of the other control types. This is done using the **Default** edit field in the Function page of the Switch Module's properties, shown in the figure below. The Default Value corresponds to the input port that is selected when you right-click the Panel representation of the Switch and then left-click the *Set to Default* menu entry or when you choose the *Recall default values* menu entry from the **Edit Bank** drop-down menu in the Snapshot tab..



Fig. 8.11 The **Mouse Reso** edit field determines how sensitive the Spin menu is to mouse movements. With the **Default** edit field you can set the default input for the Switch Module. This is the input port that is selected when you right-click the control in Panel View.

Setting the Minimum Number of Input Ports

For your convenience when building more complex Structures, you can set the minimum number of input ports of the Switch Module. This means that the number of input ports will always be at least the number that you enter in the **Min Num Port Groups** edit field, regardless of how many input ports are connected. The **Min Num Port Groups** edit field is found in the Function page of the Switch Module's Properties.

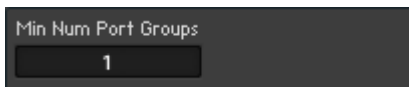


Fig. 8.12 You can set the minimum number of input ports that the Switch Module has by using the **Min Num Port Groups** edit field.

8.2.4 Lists

The List Module is a Panel Element that lets you create a list of values and use the Panel representation to select an entry from this list. Each list entry corresponds to a value being sent as an Event from the output of the List Module. A common use for the List Module is illustrated in the next paragraph, where it has a similar functionality to a Switch Module. Using this as an example, you will learn how to create a list in the Function page of the List Module. We will take a look at the other settings offered in the Function page as well. For information regarding the View page of the List Module, please refer to “Changing the Appearance of Lists and Switches” in subsection [↑8.2.3, Switches](#). If you are interested in the Default Value and Mouse Resolution settings in the Function page of the List Module, please refer to the “Setting the Default Value and Mouse Resolution for the Switch and List Modules” in subsection [↑8.2.3, Switches](#).

Routing Signals Using a List and Selector Module

The List Module can be used in conjunction with a Selector Module to provide functionality similar to the Switch Module (see screenshot below). The Selector Module “selects” which signal to forward to its output port depending on the value at the “Pos” (position) input port. For example, a “Pos” value of “0” will cause the Selector Module to forward only the signal at the “0” input port to its output port. Similarly, a “Pos” value of “1” will cause the Selector Module to forward only the signal at the “1” input port to its output port. By connecting a List Module to the “Pos” input port you can use the Panel representation of the List Module to control the routing of the signal from the Instrument Panel.

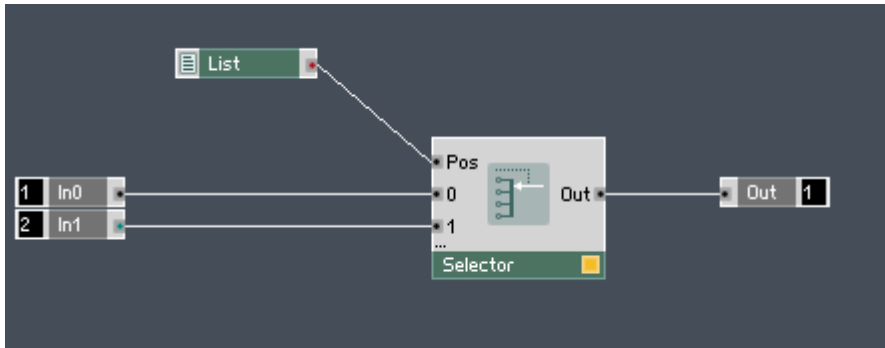


Fig. 8.13 Here you see a common Structure for routing signals. Here, a List Module is used to supply the “position” value the “Pos” input port of a Selector Module.

See below for a Panel representation of the List Module.



Fig. 8.14 A possible Panel representation of the Switch Module.

Creating and Modifying Lists in the Function page

To make the Structure from the previous paragraph work as described, you need to create a list that sends the correct values from the List Module to the “Pos” input port of the Selector Module. The following instructions guide you through this process:

1. Create a Structure such as that shown in the previous paragraph.
2. Double-click on the List Module to open up its Properties.
3. Click on the **Function** button.



4. See below the area of the Function page that you will use to create the list.

ENTRIES

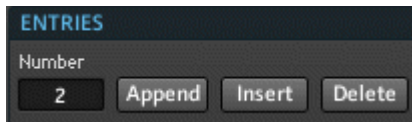
Number
2 Append Insert Delete

#	Label	Value
1	Signal 0	0
2	Signal 1	1

Default 0 Mouse Reso 127

☐ Enable Off Entry

5. You can add entries to the list by either using the Num Entries edit field or by clicking on the Append Entry or Insert Entry buttons. Double-click on the Num Entries edit field and enter the value “2”.
6. Your list should now have two entries with the names “Entry 1” and “Entry 2” in the Label column and the values “1” and “2” in the Value column. Now click on the Append Entry button.
7. An additional entry should have appeared at the bottom of your list. This one should be called “Entry 3” and have the value “3”. You can also insert entries in the list. Click on “Entry 3” and click on the Insert Entry button.
8. A new entry should have appeared between “Entry 3” and “Entry 2”. It is also labeled “Entry 3”.
9. Now click on the higher “Entry 3” and click on the Delete Entry button twice.



10. You should now see only two entries. Double-click on the **Label** edit field of "Entry 1" and rename it to "Signal 0". Do the same with the second entry, renaming it to "Signal 1".
11. Now you need to change the values of the corresponding entries. Since the Selector Module starts counting its input ports from zero and you have two input ports, you address the first input port with a "Pos" value of "0" and the second input port with a "Pos" value of "1". Double-click on the **Value** edit field of the first entry and type "0".
12. Double-click on the **Value** edit field of the second entry and type "1". Now when you go to the Panel representation (see Function page screenshot above) you can select between "Signal 0" and "Signal 1". The respective values which are sent from the List output port to the "Pos" input port of the Selector Module cause the signals at the "0" and "1" input ports to be forwarded to the Module's output.
13. You can also automatically populate the list with values that change by a given increment. This is done using the Value Generator (see below). First, double-click on the **Num Entries** edit field and type in "10".
14. Now that you have a list with 10 entries, you might want the list to start out with the value "0" and go up to "9". This might be the case when you want to create a list that controls the "Pos" input port of a Selector Module (since it is zero-based). Make sure the **Min Value** edit field is "0" and change the **Max Value** edit field to "9". The **Stepsize** edit field automatically adjusts to the Min and Max Values.
15. Now press the **Apply** button.
16. Your list should now have 10 entries with values ranging from "0" to "9" with increments of "1". Note that negative increments are also allowed.

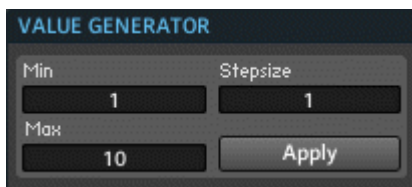


Fig. 8.15 With the Value Generator you can automatically create a list which has values that change by the increment specified in the Stepsize edit field.

8.3 Changing the Instrument Appearance

REAKTOR allows you to build your own Instruments with custom interfaces. Although a lot of what makes up your style in REAKTOR goes on under the hood, the appearance and interface also has an effect on the experience playing the Instrument. For this purpose, this section will teach you how to edit the Instrument Panel colors and how to change the background picture.

8.3.1 Editing the Instrument Panel Colors

The colors of an Instrument Panel are edited all in one place: in the View page of the Instrument Properties. Here's how to edit the different colors:

1. First, open the “SynthOne.ens” Ensemble in the “Tutorial Ensembles” folder of the factory library. Make sure you see the Instrument Panel (see below).

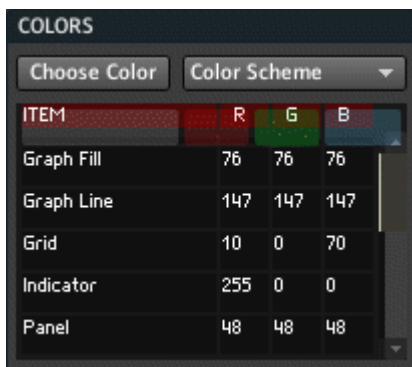


2. Click on the Instrument Header and then on the Properties button.

- Next, click on the View button.

VIEW

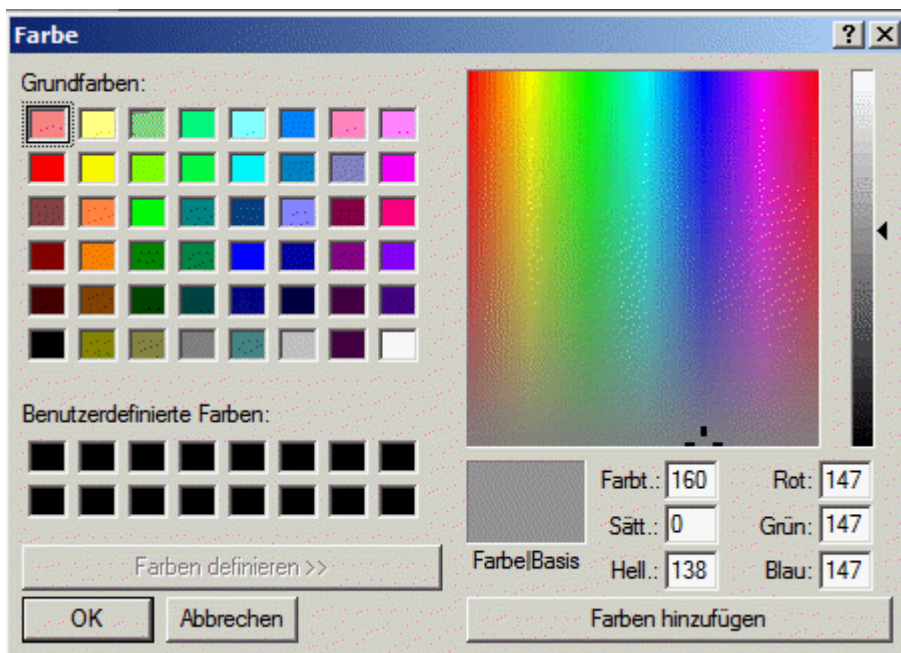
- You should now see the Color Scheme table.



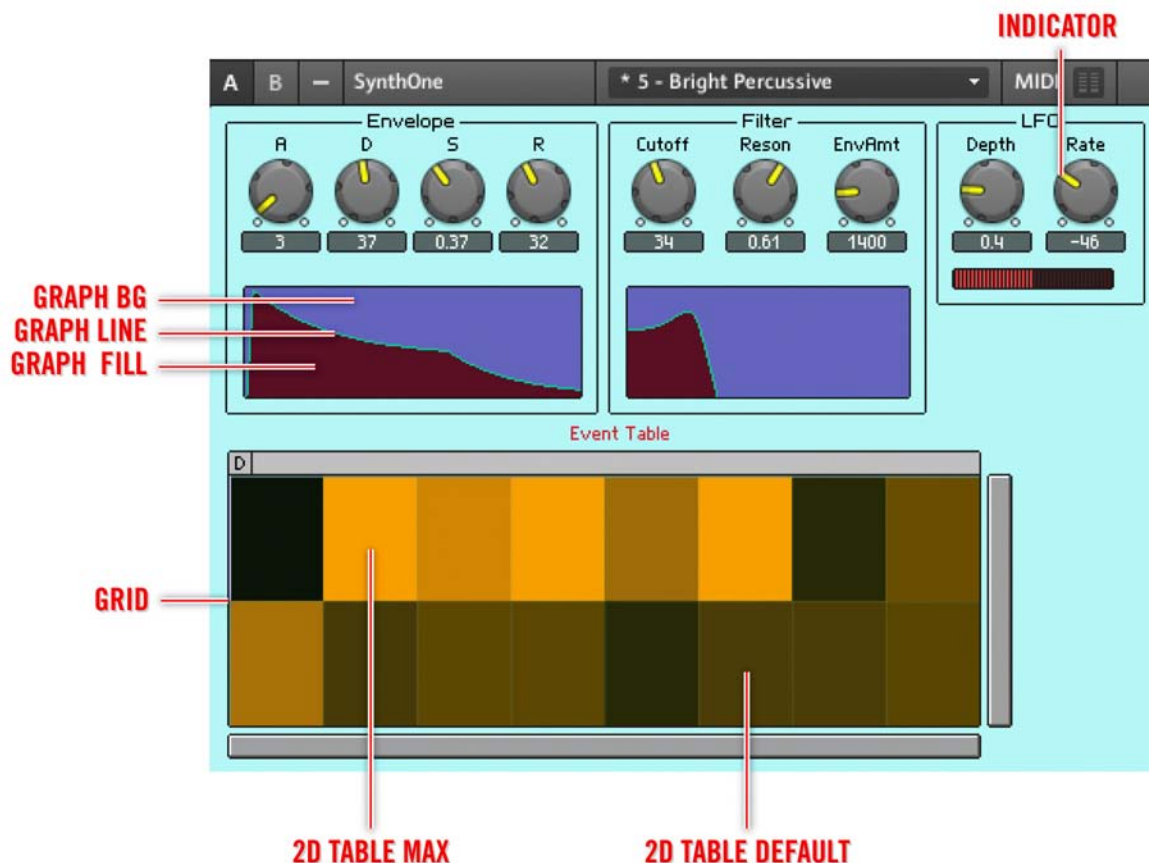
The screenshot shows a dialog box titled 'COLORS'. At the top, there are two buttons: 'Choose Color' and 'Color Scheme' with a dropdown arrow. Below these buttons is a table with the following data:

ITEM	R	G	B
Graph Fill	76	76	76
Graph Line	147	147	147
Grid	10	0	70
Indicator	255	0	0
Panel	48	48	48

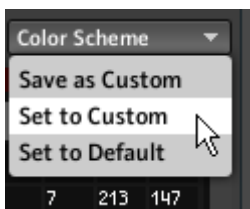
- In the Color Scheme table you can either set the color of each component by explicitly specifying the desired color's "RGB" in the corresponding "R", "G", and "B" columns. Most of the time, though, this is rather inconvenient. Click on the Panel row of the table and then on the Choose Color button above the table.
- The Color dialog box should have popped up (see below). Choose a color that you would like for the Instrument Panel background and click "OK".



7. You should now see how your Panel background has changed to the color that you chose. See the screenshot below for the names of all the components for which you can change the color in the Color Scheme table. Next, click on the Color Scheme drop-down menu.



8. From the Color Scheme drop-down menu in REAKTOR, you can globally choose one Custom Color Scheme. If you would like to set the current Color Scheme as the global Custom Color Scheme, select the *Save as Custom* menu entry. Beware though, that any Color Schemes that you saved in other sessions will be overwritten. If you have a Custom Color Scheme already saved, you can click on the *Set to Custom* menu entry. The current Color Scheme will be then replaced by the Custom Color Scheme. If you would just like to revert to REAKTOR's default Color Scheme, choose the *Set to Default* menu entry.



8.3.2 Choosing the Background Picture

Beyond just choosing what colors you would like your Instrument to have, you can further give your Instrument unique character by applying a background picture. This is done in the following steps:

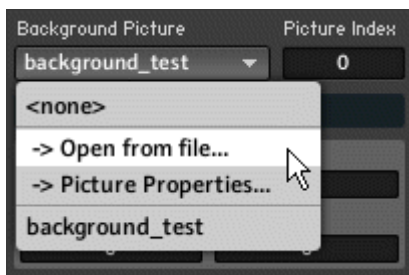
1. First, open a simple Ensemble like “SynthOne.ens” in the “Tutorial Ensembles” folder of the factory library. Make sure you see the Instrument Panel.



2. Click on the Instrument Header and then on the Properties button.
3. Next, click on the View button.



4. To browse for a .tga or .bmp picture file to become your Instrument background, click on the -> *Open from file...* menu entry in the Background Picture drop-down menu.



5. Once you have chosen a .tga or .bmp file in the browser and clicked “OK”, the Picture Properties dialog box (see screenshot below) should pop up on your screen. For now, click on the “OK” button. For more information on the Picture Properties dialog, please refer to section [↑8.5, Customized Panel Elements](#).



6. With default settings, the Instrument Panel takes up the area required by all visible Panel Elements. If the Panel Elements take up an area that is smaller than the background picture, then the background picture gets clipped. Also, the top-leftmost Panel Element will always appear in the top-left corner of the background picture. Often you have some border for your background picture where you don't necessarily want the Panel Elements to appear. With the Picture Borders edit fields (see screenshot below) you can specify the border area where the Panel Elements should not appear.

By increasing the border either on the left or right, or top or bottom, you can increase the effective area of the Instrument such that the whole background picture is shown, regardless of the area that the Panel Elements take up.

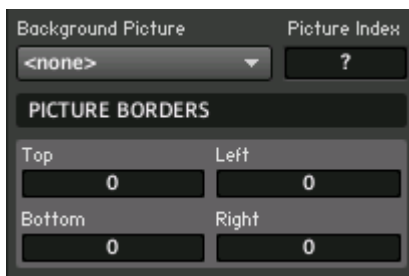


Fig. 8.16 The Picture Borders edit fields let you specify the borders for your background picture in pixels. The Picture Index edit field lets you choose which animation frame is used for your Instrument background. This is only valid if you have configured the animation in the pictures properties dialog.



The Picture Borders edit fields also let you create a border for your Instrument Panel even if you have no background picture loaded.



All changes made to a picture file in the Picture Properties dialog are applied to all instances of this picture in the Ensemble.

If you loaded and configured an animation in the Picture Properties dialog box of your background picture (see section [↑8.5, Customized Panel Elements](#) for this), you can also choose which animation frame should be taken as your Instrument background. This is done with the Picture Index edit field.

8.4 Choosing the Macro Appearance

Similar to Instruments, you can also customize the appearance of your Macros. You can select between different types of frames and background pictures that appear on the area on the Instrument Panel occupied by the Macro and even make them appear and disappear. Follow these instructions for a small walkthrough regarding the Appearance Properties of a Macro:

1. First, open a simple Ensemble like “SynthOne.ens” in the “Tutorial Ensembles” folder of the factory library. Make sure you see the Instrument Panel.



2. Click on the frame of the macro labeled "LFO" and then on the Properties button.
3. Next, click on the View button.



4. Let's first look at the appearance Properties.

The screenshot shows a dark-themed dialog box for editing a panel. It has several sections: 'VISIBLE' with a checked 'Visible' checkbox; 'SIZE' which is currently empty; 'STYLE' with three radio buttons for 'Frame Style' (3D Frame, Line Frame, No Frame) and a checked 'Frame with Label' checkbox; a 'Background Bitmap' dropdown set to '<none>' and a 'Picture Index' button with a question mark; a checked 'Includes Frame' checkbox; and 'PICTURE BORDERS' with four numeric input fields for Top, Left, Bottom, and Right, all set to '0'.

VISIBLE	
<input checked="" type="checkbox"/>	Visible

SIZE	

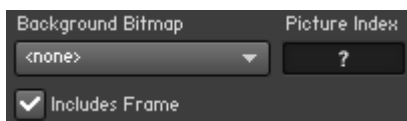
STYLE	
Frame Style	
<input type="radio"/>	3D Frame
<input checked="" type="radio"/>	Frame with Label
<input type="radio"/>	Line Frame
<input type="radio"/>	No Frame
Background Bitmap	Picture Index
<input type="text" value="<none>"/>	<input type="button" value="?"/>
<input checked="" type="checkbox"/>	Includes Frame

PICTURE BORDERS	
Top	Left
<input type="text" value="0"/>	<input type="text" value="0"/>
Bottom	Right
<input type="text" value="0"/>	<input type="text" value="0"/>

5. Engage the ☐ checkbox to make the Macro disappear from the Panel.
6. Disengage the ☐ checkbox to make the Macro reappear.
7. Now let's look at the Frame Style selector. Clicking on a selector box causes the frame of the Macro to adapt the respective style. Click on the selector boxes to see the different frame styles.



8. You can also make the Label of the Macro, in this case the text “LFO”, disappear by disengaging the *Frame with Label* checkbox. Do this now.
9. Browsing for and loading a background picture for a Macro is exactly the same as in the case of an Instrument. Load a background picture for your macro. If you don't know how to do this, please refer to subsection [↑8.3.2, Choosing the Background Picture](#) for instructions.



10. Once you have loaded a background picture for the Macro, you can choose if you want the frame to appear in addition to the background picture. You can make the frame disappear by disengaging the *Includes Frame* checkbox (see screenshot above). The screenshot below shows the Line Frame style overlaying the background picture when the *Includes Frame* checkbox has been disengaged. Try out different combinations of the Frame Style and the Includes Frame settings to see what combination you like most.

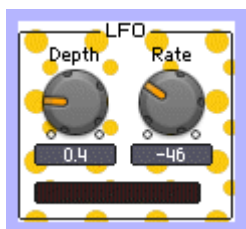


Fig. 8.17 A Macro with a custom background picture. Here the Includes Frame checkbox has been disabled.

8.5 Customized Panel Elements

It is possible to create fully customized Instrument Panels in REAKTOR. In addition to background pictures, discussed in the previous two sections, you can also apply your own skins to Panel Elements. Furthermore, you can create bitmap animations that react to user input, going so far as to build your own custom controls.

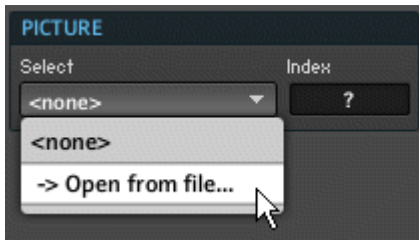
8.5.1 Supported Formats and Loading Pictures

For Modules that support pictures, you can load them in the View page or the Function page of the Module Properties. This is done using a designated drop-down menu. REAKTOR allows you to load 24-bit Bitmap (*.bmp) and 32-bit uncompressed Targa (*.tga) images in many places: as Instrument and Macro backgrounds, as Structure Icons, and in Panel Elements such as the Fader, Multi Display, Picture, and Multi Picture Modules. The advantage of using the Targa format is that it enables you to add a transparency layer to the image (see the instructions below). To see how to load a picture and place it on the Instrument Panel using a Picture Module, follow these instructions:

1. Create a new Instrument and add a Picture Module to it. You'll find the Picture Module in the *Built-In Module > Panel* submenu.
2. Double-click on the Picture Module to open up its Properties.
3. Click on the Function button.

FUNCTION

4. To browse for a .tga or .bmp picture file to represent the Picture Module on the Panel, click on the -> *Open from file...* menu entry in the Select Picture drop-down menu, shown in the figure below.



- One you have chosen a .tga or .bmp file in the browser and clicked “OK”, the Picture Properties dialog box (see below) should pop up on your screen. Whenever you want to go back to change the Picture Properties, you can access the dialog from the -> *Picture Properties...* menu entry in the *Select Picture* drop-down menu (see screenshot above). For now, click on the “OK” button. We will come back to the Picture Properties dialog in the subsections below.



Fig. 8.18 The Picture Properties dialog box.



All of the pictures that are loaded in an Ensemble are available to all Objects in that Ensemble (in their drop-down picture menus) that can display pictures. Using the same picture more than once does not appreciably increase an Ensemble's memory needs. Hence, as long as you work with a relatively small number of picture files, you can use them as much as you want without dire RAM consequences.

8.5.2 Transparency

Transparency allows you to place pictures with arbitrary shapes on the Panel. The transparency feature is only applicable for Targa picture files that have an alpha channel. For example, in the below screenshot, the picture has transparency outside of the logo's borders. Engaging the `Has Alpha Channel` checkbox illustrates this fact in the Preview area of the Picture Properties dialog. You can see that the previously white area has been replaced by the background color.



Fig. 8.19 You can make certain areas of a Targa picture file transparent by the use of an alpha channel. On REAKTOR's side, the Has Alpha Channel checkbox has to be engaged in order for the alpha channel to be recognized.

- ! All changes made to a picture file in the Picture Properties dialog are applied to all instances of this picture in the Ensemble.

8.5.3 Knob Skins and Animations

Pictures that react to user input have to be loaded in the form of a collection of animation frames. In this subsection you will learn how to configure such a picture file so that it can become an animated skin for a knob. Follow these instructions:

1. Open up a new Instrument and add a knob to its Panel.
2. Double-click on the knob open up its Properties.
3. Click on the *View* button.

VIEW

4. To browse for a .tga or .bmp picture file to represent the Picture Module on the Panel, click on the -> *Open from file...* menu entry in the *Skin Bitmap* drop-down menu.
5. Note that all changes made to a picture file in the Picture Properties dialog are applied to all instances of this picture in the Ensemble. This means that you might want to load the same picture twice to have independent versions to apply the picture properties to. So that you don't confuse them in the *Select Picture* or *Skin Bitmap* drop-down menus, change the name of the picture file in the *Name* edit field in the Picture Properties dialog (screenshot below).
6. For this example, we are using the Targa picture file labeled "citrus" (see figure below) as the skin for the knob. Notice that the picture consists of a column of animation frames. Currently, the knob would appear on the Panel with a white rectangular background. Since this would look awkward on most interfaces with non-white backgrounds, an (invisible) alpha channel has been laid over the white areas of the picture file. To activate the effect of the transparency, click on the *Has Alpha Channel* checkbox.



7. With the transparency activated, the white parts of the skin which do not belong to the knob itself should have turned transparent (see figure below). Right now the picture file is still recognized by REAKTOR as one very tall picture. With the Num Animations edit field we can tell REAKTOR to slice the picture into a number of animation frames of equal size. The Num Animations, Animation Height and Animation Width edit fields are interdependent, along with the Horizontal checkbox. The Animation Height and Animation Width edit fields specify the dimensions of one animation frame. If you have a picture file that has the animation frames set in a row, engage the Horizontal checkbox.



8. Next, type in the number of animation frames in the Num Animations edit field. The Animation Height and Animation Width values should be adjusted automatically to suit the size of the loaded picture file. Conversely, if you don't know the number of animation frames but the dimensions of one animation frame, specify those in the respective edit fields.
9. Now that the picture has been recognized as a series of animation frames in REAKTOR, the Preview area of the Picture Properties dialog should now look similar to the screenshot below.

Name: citrus

TRANSPARENCY

☒ Has Alpha Channel

ANIMATION

Numbers: 31 ☐ Horizontal

Animation Height: 32 Animation Width: 32

RESIZABILITY

☐ Vertical ☐ Horizontal

BORDERS

Border Top: 0 Border Left: 0

Border Bottom: 0 Border Right: 0

PREVIEW

OK CANCEL

10. Now you can click on the OK button and try out your new control.



Fig. 8.20 A knob with a custom skin.



All changes made to a picture file in the Picture Properties dialog are applied to all instances of this picture in the Ensemble.

8.5.4 Resizability and a Custom Control

The “Resizability” feature in the Picture Properties dialog serves two main purposes. It reduces the size of Ensemble files by enabling small pictures to be tiled so that they can fill large panel areas, as is the case for background picture files. Also, it uses scaling to enable pictures such as fader skins to fit their Panel Elements. In this subsection you will learn about these two applications.

Resizing Background Pictures

If the Panel area is larger than the background picture file, the picture will be tiled until it fills up that Panel area. On the other hand, if your background picture should include custom borders for the Panel, then simply tiling the picture will not solve your problem: your borders would also be tiled. The Resizability area of the Picture Properties dialog serves to achieve borders along with tiling in a more effective manner. Instead of having to load a large picture file that has the exact size of the respective Panel into your Ensemble and having that bloat the size of your Ensemble file, you can load a small picture that includes the borders and the part of the background that you would like to tile. The previous chapter showed how a small background picture for a Macro can be made to separately account for borders and the “main area.” In the Picture Properties dialog for the background picture, the Border Top, Left, Bottom, and Right edit fields have been set to the values shown in blue (see below).

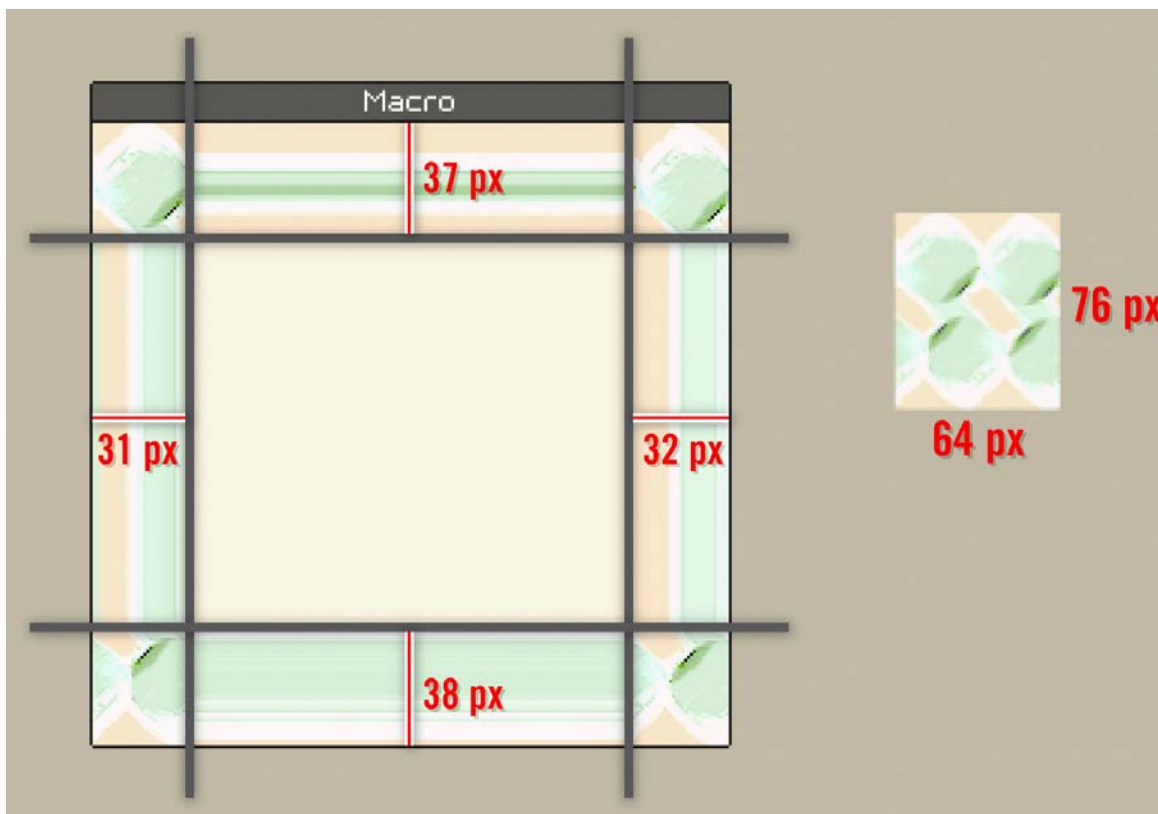


Fig. 8.21 The Macro on the left uses the Bitmap shown on the right as its background picture.

These are the parts of the picture that are not tiled. In this case, the borders include all but the pixel with the coordinates 38 px from the top and 32 px from the left. This pixel is then taken as one tile and repeated both vertically and horizontally over the whole Macro Panel background excluding the borders. The tile can be made to repeat only in the vertical or only in the horizontal direction, depending on which of the [Horizontal](#) or [Vertical](#) checkboxes is engaged. The settings for the Resizability area in this particular case are shown below.



Fig. 8.22 The settings for the Resizability area for the case of a background picture that is illustrated by figure 9.62.

Using Resizability in Custom Controls

Say you have an Instrument Panel that utilizes the style of horizontal fader (see screenshot below). Furthermore, you would like to have controls that have areas of different lengths for the control label. In this case, the “Resizability” feature saves you from having to load a separate skin for each control that has a different area for its label.



Fig. 8.23 Using the Border Left edit field in the Picture Properties dialog, the original skin in a) is resized to include a longer text fields on the right for b) and c).

In order to create controls with areas of different length for the label, but using only one picture file, follow the instructions below. You will also learn how to create a custom control using a Multi Picture and Mouse Area Module.

1. First, load a new Ensemble.
2. Second, insert a Multi Picture Module into the Structure.
3. Double-click on the Multi Picture Module to go to the Module Properties.
4. Click on the Function button.

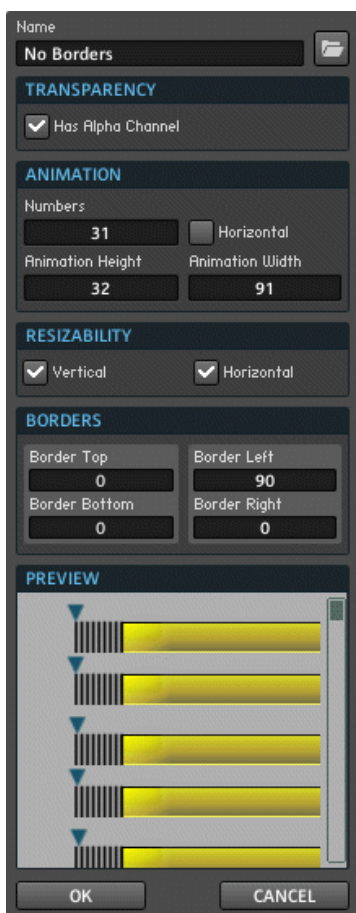
FUNCTION

5. Now, a picture with the -> *Open from file...* menu entry in the Select Picture drop-down menu. Make sure your skin has a form where one side is designated for the control label.

6. If applicable, engage the Has Alpha Channel checkbox. Also, enter the specifications for your animation, as shown in subsection [↑8.5.3, Knob Skins and Animations](#).
7. Now that the animation frames have been configured, let's turn our attention to the resizing of the skin. In our case, we want to extend the right side of the control. This means that the Horizontal checkbox in the Picture Properties dialog should be engaged. The whole picture starts tiling itself in the horizontal direction, as shown in the screenshot below. If you have a label that comes on top or below the control, engage the vertical checkbox.



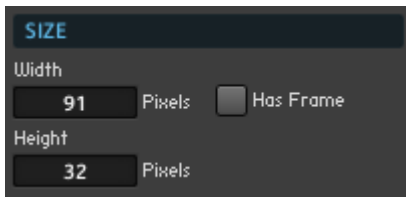
8. To extend the right side, we have to think in terms of borders. This means that since the left side of the control should not be repeated to fill up the full desired length of the control, it should be treated as a border. By setting the size of the left border to 90 px, just one pixel short of the full horizontal length of the picture we ensure that only the rightmost column of pixels is horizontally tiled (see the previous header “Resizing Background Pictures” for more information on the geometry of the borders and tiling). The result should be something like the figure below.



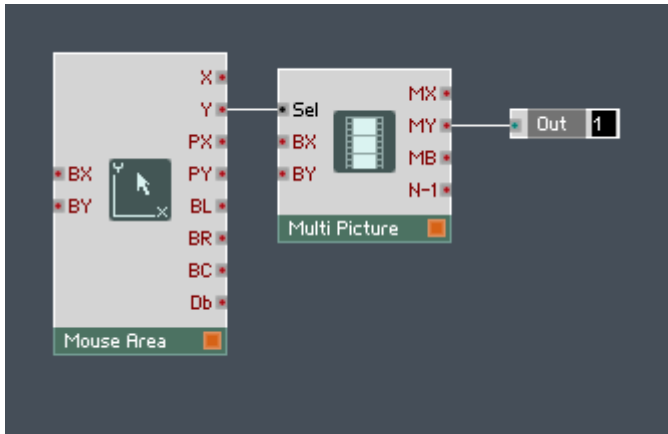
9. Now the resizing behavior of the picture has been configured. Click on the “OK” button.
10. Next you actually need to control the size of the picture on the Instrument Panel. For this, click on the View button of the Multi Picture Module.



11. Use the Width and Height edit fields (see below) to change the size of the picture on the Instrument Panel. Notice that in the case where one has made the right side of the picture horizontally resizable, increasing the Width edit field causes the new space to fill up with the tiles specified in the Resizability area of the Picture Properties dialog.



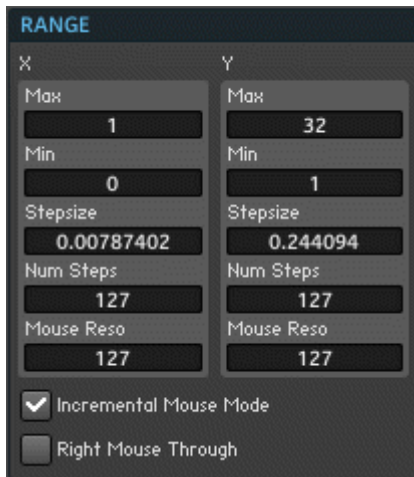
12. To make a usable control out of this, add a Mouse Area Module to the Structure and connect its “Y” output to the “Sel” input port of the Multi Picture Module (see below). The Mouse Area Module is a Panel Element that allows you, among other things, to click and drag the mouse on it, much like a conventional control. While the mouse button is pressed, the absolute Y coordinate of the mouse in respect to the mouse area on the Panel is sent to the “Y” output port of the Mouse Area Module.



13. The “Sel” input port of the Multi Picture Module is used to tell the Module which animation frame to display on the Instrument Panel. Hence, the range of the values entering this input port should be from 1 to the $n+1$ where “n” denotes the number of animation frames of the skin you loaded. In the current case, the range should be 1 to 32 since the number of animation frames is 31. This affects the range of the output values of Mouse Area Module that we want to have. To change this range, double-click on the Mouse Area Module to open up its Properties.
14. Click on the Function button.

FUNCTION

15. Now change the Min Y and Max Y edit fields to “1” and “32”, respectively (see screenshot below).
16. For conventional controls, clicking and dragging the mouse over it changes the position of the control instead of setting it to the location where you clicked the mouse. This means that it doesn’t matter where the mouse is on an absolute scale when you click it, but how far you move it. In step 12 we said that the Mouse Area Module sends the absolute coordinate of the mouse in respect to the mouse area on the Panel to its output. Luckily, the Mouse Area Module can also track a relative change of the mouse. This makes it possible for us to use the Mouse Area Module like a normal control. To activate this mode, engage the Incremental Mouse Mode checkbox (see screenshot below).



17. Next, you need to set the size of the Panel representation of the Mouse Area Module such that the mouse area can cover the picture of your control. This is done by first clicking on the View button.



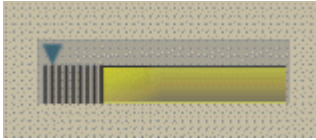
18. Now set the Width and Height edit fields to be about the same size, perhaps a bit larger, than the Multi Picture's size on the Panel. This is done by using the edit fields (see screenshot below).



19. Finally, you need to place the mouse area on the Instrument Panel over the picture of our control to make it feel like a real fader. Click on the Panel Lock button.



20. Click and drag the mouse area over your control.



21. Lock the Panel again and try out your new custom control! If you want you can now add a label of any length to your control with the help of a Text Module.

8.5.5 General Guidelines for Skins

REAKTOR enables you to customize the appearance of several Panel Elements by applying skins to them: faders, knobs, buttons, lists, switches, Receive Modules, lamps, and meters. This subsection describes some of the particularities that are associated with different Control Types. Read the previous sections for more general information on how to load and edit pictures files.

Fader Skins

There are two types of fader skins: skins consisting of a single picture and skins that have a number of animation frames. In skins with a single picture, the picture is used as the handle (not the body) of the fader. If the picture has been made resizable by engaging either the *Horizontal* or *Vertical* checkbox in the Picture Properties, it is resized to the horizontal or vertical size of the original REAKTOR fader handle. If not, the handle size is the same as the picture size.

You can use the *Head (Pixels)* edit field in the View page of the Module Properties to add a head to your custom fader handle. The *Head (Pixels)* edit field specifies how many pixels are taken up by the Head, it has a range [0 ... 20]. If you set the *Head (Pixels)* edit field to “0”, the handle is placed exactly within the fader groove and no head appears. Setting the *Head* edit field to “20” creates a handle head that is 20 pixels wide.

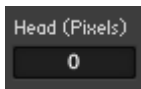


Fig. 8.24 The Head (Pixels) edit field

With a fader skin that consists of animation frames, the entire fader (not just the handle) is replaced by the picture; thus the fader size is determined by the picture size, and the Resizability area in the Picture Properties dialog and the **Length** edit field in the View page are ignored. The number of fader states is equal to the number of animation frames in the picture.

For all fader skin modes — single-picture, animation, and none — the **Hide Scale** and **Hide Groove** checkboxes in the View page hide the fader scale graphics (tick marks) and groove, respectively.

Knob Skins

A knob skin is always treated as an animation. The entire knob is replaced by the animation picture. As a result, the knob size is determined by the size of the animation frame and the in the Resizability area in the Picture Properties dialog and the **Length** edit field in the View page are ignored. The number of knob states is equal to the number of animation frames in the picture. For a more detailed description on applying skins to knobs, see subsection [↑8.5.3, Knob Skins and Animations](#), “Knob Skins and Animations”.

Button Skins: Button, List, Switch, and Receive

A button skin uses a picture that consists of four animation frames to define its four states. The states should be represented by the respective animation frames in the following order: “Off” state “Up”, “On” state “Up”, “Off” state “Down”, and “On” state “Down”. If the picture has been made horizontally or vertically resizable (by engaging the **Horizontal** or **Vertical** checkboxes in the Picture Properties, respectively), it is resized to the horizontal or vertical size of the original button. If not, the button size is the same as the picture size. The List, Switch, and Receive Modules can all have button-type skins if their style is set to Buttons using the Style menu in the View page. See below for a List and Switch Module employing the same custom skin.

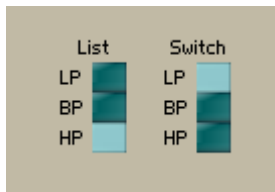


Fig. 8.25 A List and Switch Module employing the same button skin

Lamp Skins

A lamp skin uses a two-frame animation picture to define its two states. The picture should have the animation frames for the two states in the following order: “Off” state, and “On” state. If the picture has been made horizontally or vertically resizable (by engaging the *Horizontal* or *Vertical* checkboxes in the *Picture Properties*, respectively), it is resized to the horizontal or vertical size determined by the *Width* and *Height* edit fields in the *View* page of the *Lamp Module's Properties*. If not, the lamp size is the same as the picture size.

Meter Skins

There are two types of meter skins: skins with an “Off” and an “On” state and skins that consist of animation frames. A skin that has only the “Off” and “On” states should have the corresponding animation frames in that order.

An animation skin uses a multiple-frame animation picture to define the meter's states. The number of meter states is equal to the number of animation frames in the picture. The number of segments is also determined by the number animation frames, and the *Number of Segments* edit field in the *View* page is ignored. If the picture has been made horizontally or vertically resizable (by engaging the *Horizontal* or *Vertical* checkboxes in the *Picture Properties*, respectively), it is resized to the horizontal or vertical size determined by the *Width (Segments)* and *Height (Segments)* edit fields in the *View* page of the *meter Module's Properties*. If not, the meter segment size is the same as the picture size.

9 The Primary Level Paradigm

This chapter will teach you about the fundamental concepts behind building Structures in the Primary Level. All Structures contain wires that carry signals. For that reason the two signal types in Primary Level, Event and Audio, will be discussed first. When playing several notes at the same time, your Instrument can make use of several Voices. To learn how this feature, called “Polyphony”, is treated in REAKTOR, read the section called “Polyphony”. Finally, when you load an Instrument or Ensemble, the initial flow of Events and Audio signals through the Structure is called Initialization. Understanding this behavior is crucial to building Instruments that behave the way you want them to even after closing and reopening the Instrument.



Note that you should have read the REAKTOR Getting Started Manual before reading Part II and III of the REAKTOR Main Manual. By reading the Getting Started Manual, you will get an intuitive understanding for building functioning Primary Level using Instruments, Macros, and Modules. This will help you grasp many of the concepts described in these Parts of the Main Manual.

9.1 Audio signals

An Audio signal is a signal that is updated at the sampling rate and can be routed to your sound card. Think of it as something that you could play from your speakers. Note that Audio signals can be used strictly for modulating other signals without ever sending them to the Audio Out Module. An example of an Audio signal is shown in the screenshot below, the output of a Sine Module. Audio and Event signals can be combined, as shown in [↑9.2.1, An Event Is Tied to a Point in Time](#). Note that because Audio signals are updated at the sampling rate, then for a sampling rate of 44,1 kHz the values are calculated 44100 times per second. This can be a considerable load on your computer's CPU resources if the calculations are exceedingly complex. When creating an Audio signal in the REAKTOR Structure, always consider if the same goal could not be achieved using the less CPU-hungry Event signal type.

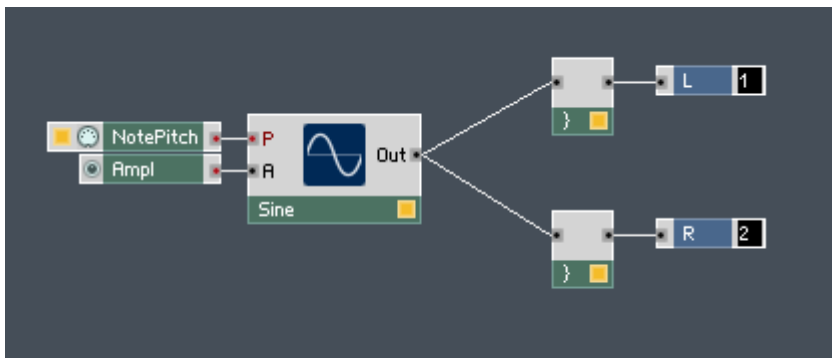


Fig. 9.1 The output of a Sine Module, a sine wave, is an example of an Audio signal. This can be deduced from the black coloring of the output port.



To change the rate at which Audio signals are updated you need to change the Sample Rate, as shown in section [3.5, Other Common Operations](#).

9.2 Event Signals

Event signals are made up of values sent at fixed points in time. You can send single Events to trigger samples or envelopes, for example, but you can also have a Module (such as the LFO Module) send a constant stream of Events to serve as a control signal. Even if you are sending a constant stream of Events, they are updated at a much slower sub-audio rate, the so-called Control Rate. The reason for the Event signal type is that Modules process Event input ports only when Events arrive at them. This saves CPU resources for all signals that do not need to be updated at the Sample Rate. Therefore, Event signals usually control the flow and properties of your Audio signal. The figure below shows the LFO Module; its output is an Event signal.

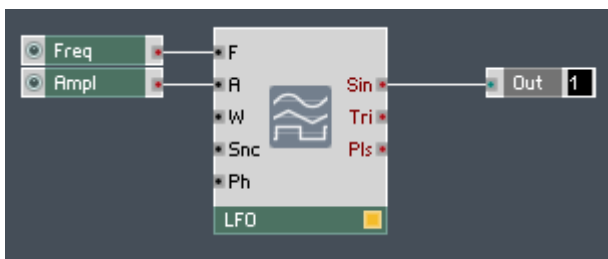


Fig. 9.2 The output of an LFO Module is an Event signal, as seen by the red color of the output ports.



Some Modules, like the LFO Module, have a constant Event stream at their output ports. This stream is sent at the Control Rate. To learn how to change the Control Rate, please refer to section [3.4, Navigating with the Browser](#).

9.2.1 An Event Is Tied to a Point in Time

Event signals have one fundamental property that differentiates them from Audio signals. Each value carried by an Event signal also has the point in time that it was sent from its source, relative to other Events, tied to it. An example of an Event for which the timing is important is the Event sent from the output of the Gate Module as a response to a key being pressed on your MIDI or computer keyboard. This Event is usually used to trigger *when* a note is being played by the Instrument. A simple Structure where the Event signal from the output of the Gate Module is used to trigger an envelope that in turn controls the output of a sine oscillator is given in the screenshot below. One often refers to “Events” rather than the Event signal. An “Event” then means a value that is sent at a specific point in time relative to other “Events”.

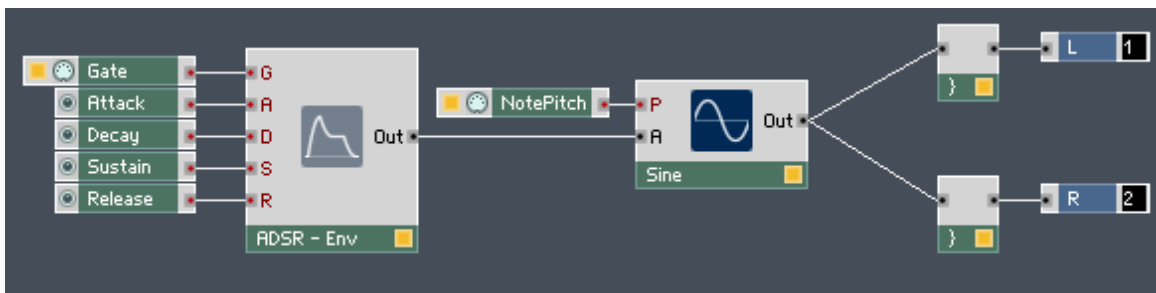


Fig. 9.3 Event signal triggering an envelope which again controls the output of an oscillator.

In the screenshot above, the Event (signal) from the Gate Module is sent when a key is pressed on your MIDI or computer keyboard. This triggers the envelope that in turn controls the amplitude of the sine oscillator with an Audio signal. The result at the output port of the Sine Module is a sine wave that has an amplitude envelope applied to it. In whole, the sine wave is heard at the Instrument output when an Event arrives from the Gate Module.

The below illustration shows an example of combining Audio and Event signals.

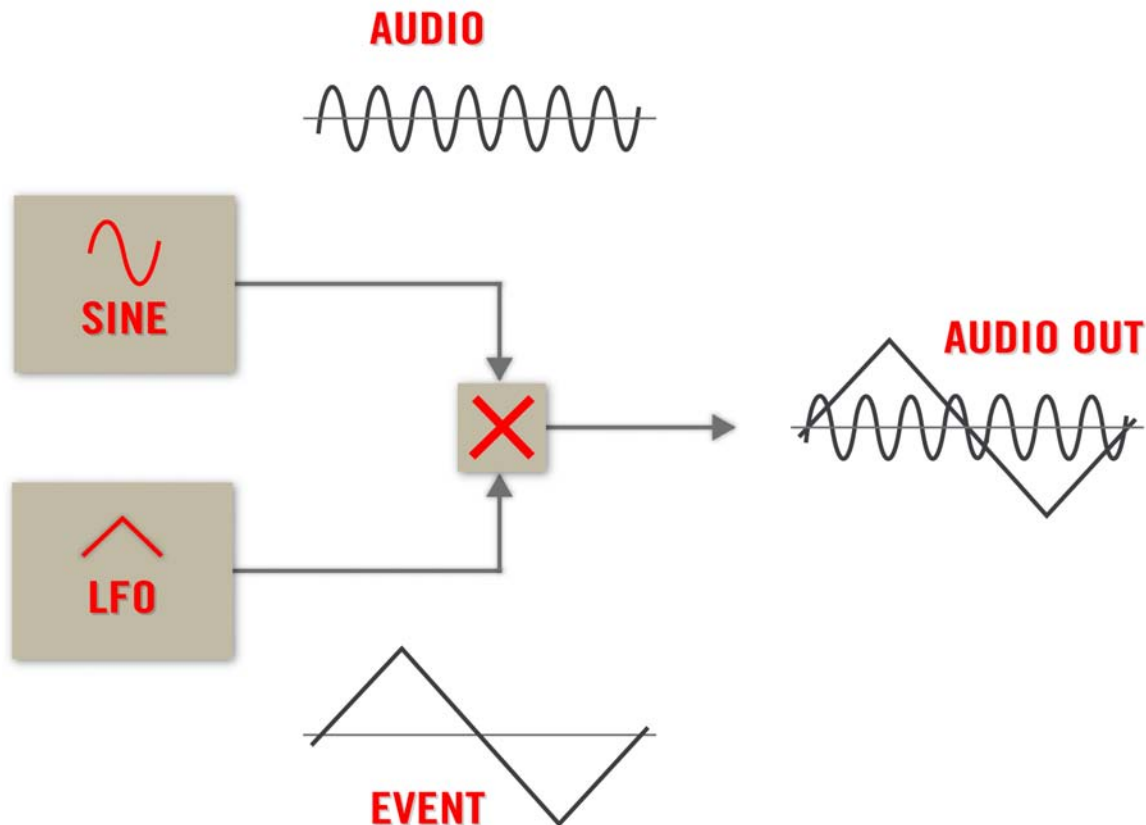


Fig. 9.4 Combined Audio/Event signal

A simple case where Audio and Event signals are combined is when the amplitude of a sine wave is modulated by an LFO. For this purpose, the (Audio) output signal of the Sine Module is multiplied with the (Event) output signal of the LFO Module using a Multiply Module. The resulting output signal of the Multiply Module is the Audio signal carrying a sine wave that has its amplitude modulated by an LFO.

9.2.2 The Order of Events

There are no simultaneous Events in REAKTOR's Primary Level. This means that all the Events in the Primary Level have a strict order. Furthermore, Events can trigger Modules to send other Events, as shown in the screenshot below. In this example an Event at any input port of the Add Module triggers an Event to be sent from its output port. Since we need two Events at the input to create a sum of two values, two Events are in turn produced at the output. Referring to the below screenshot, when the first button is pressed on the Instrument Panel, an Event with the value "1" is sent from the corresponding Button Module in the Structure. The Add Module adds the value "2", which lies at its bottom input port, (if the second Button Module has previously been pressed) to the value of the Event sent from the first Button Module. Then a new Event with the sum as its value is sent from the output port of the Add Module. A new Event sent from either Button Module, regardless if the value is the same as the old value or not, triggers an Event from the output port of the Add Module.

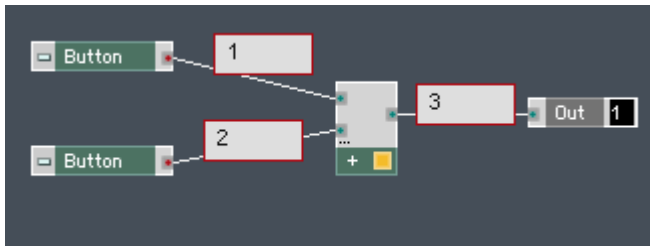


Fig. 9.5 This Structure portrays a case where an Event sent from either one of the Button Modules triggers another Event at the output port of the Add Module.

9.2.3 Events from the Same Source

What if the two input ports of the Add Module are connected to the same source, as shown in the figure below? Then the Events sent from the same source will arrive at the input ports of the Add Module at separate times and trigger two Events. In the below screenshot, the Button Module is in “Trigger” mode with “0” and “1” for the Off and On values, respectively (see subsection [↑8.2.2, Buttons](#) for more information on Button Modules). Let's say that the Events from the Button Module are first sent to the top input port and then to the bottom input port (they cannot arrive simultaneously and in a case like the one in the screenshot below, the order of Events depends on the order in which the wires were created). Right after Initialization, both input ports have the value “0”. When the button is pressed on the Instrument Panel, the Event with value “1” first arrives at the top input port and as a result the Add Module sends the sum of “1” and “0” as an Event with the value “1” from its output port. Next the value “1” arrives at the bottom input port and the Add Module sends another Event carrying the sum of “1” and “1” as its value from its output port. All in all, if we would be sitting at the output port of the Add Module we would see two Events, one carrying the value “1” and another carrying the value “2” when the button is pressed once.

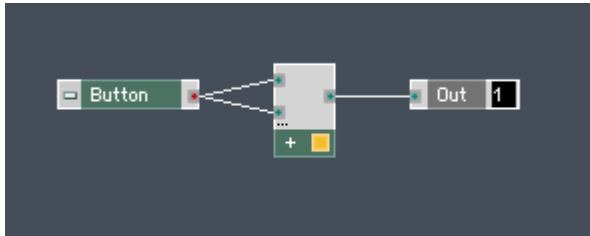


Fig. 9.6 When an Event source is connected to two target ports, the order of the Events arriving at the ports is arbitrary. In the example shown, one press of the button on the Panel causes two Events to be sent from the output port of the Add Module.

9.2.4 Controlling the Event Order

Looking at the example from the previous chapter, it is clear that having one Event which has only one operation performed on it trigger two Events can be disturbing to our signal flow. For one, if there are further operations down the line, having two Events instead of

one increases the CPU load (since all operations are performed on both Events). Second, if the result of the operation should trigger something like an envelope, then the envelope would be triggered twice — not necessarily a desirable characteristic. For this purpose, the Order Module stands available to you.



Fig. 9.7 The Order Module lets you distribute an Event in a particular order. You can find it in the Built-In Module > Event Processing submenu.

The Order Module

The Order Module takes an Event and sends it to its output ports in the following order: “1”, “2”, and then “3”. There is one additional aspect to the timing of the output Events. After the Event at the input port is sent from the “1” output port, the Module “waits” until the signal stemming from the “1” output port has been fully processed by all the Modules downstream before the next Event is sent from the “2” output port. Similarly, the Order Module sends the final Event from the “3” output port only when the signal from the “2” output port has reached all terminuses. How do we apply this knowledge in the case shown in [↑9.2.3, Events from the Same Source](#) example shown at the end of the previous subsection? There we want to add the value of the Event from the output port of the Button Module to itself and the sum to be carried by only one Event. Let's think about this in the context of the Order Module, that is, we connect the output of the Button Module to the input port of the Order Module.

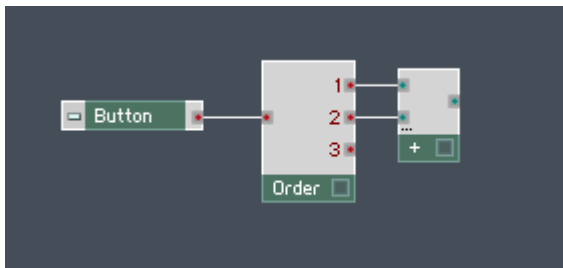


Fig. 9.8 The output port of the Button Module is connected to the input port of the Order Module.

The Order Module should send the first Event, from its “1” output port, to the first input port of the Add Module and the result of the add operation should only be sent from the Add Module once the second Event, from the “2” output port of the Order Module, arrives at its second input port. A signal that is sent only after the second Event has been processed by the Add Module, comes from the “3” output port of the Order Module. This Event then should trigger the sum of the two values to be sent downstream.

The Value Module

For triggering values (or results of operations) with Events, REAKTOR supplies you with the Value Module (see below). The Value Module lets you use an Event at the “Trig” input port to trigger an Event from the output port. The output Event carries the value that lies at the bottom input port.



Fig. 9.9 The Value Module lets you trigger the value that lies at the lower input port to be sent as an Event from its output port. The triggering signal is an Event at the “Trig” input port. You can find it in the Built-In Module > Event Processing submenu.

The combination of the Order Module with the Value Module is one that you will use frequently when working in the Primary Level. The figure below shows you an example where this combination works to extract only one Event carrying the sum of the Button's output Event with itself. The Order Module sends the Event from the Button Module from its “1” and “2” output ports to the Add Module. The output of the Add Module reaches the bottom input port of the Value Module, but since the triggering Event hasn't arrived yet to the “Trig” input, no Events are sent to the Out Module. Only when the final result has been calculated (that is, the signal from the “2” output port of the Order Module has reached its terminus), does the Order Module send the Event from the “3” output port. This Event then triggers the result of the calculation, the value at the bottom input port of the Value Module, to be sent as a single Event to the Out Module.

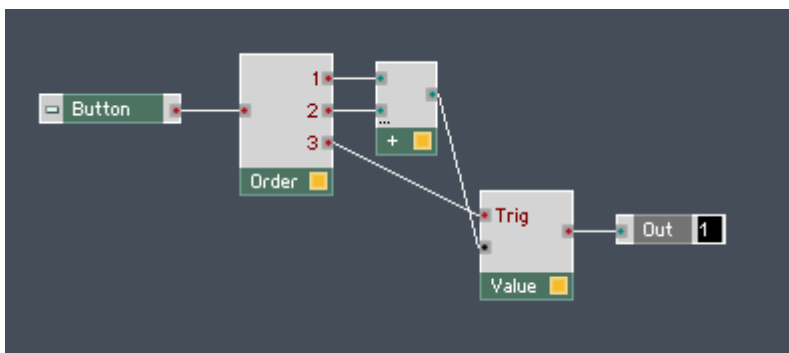


Fig. 9.10 A common combination of the Order and Value Modules. The Order Module distributes the incoming Event in such a way that the last Event triggers the output of the finished operation to be sent as an Event to the Out Module.

9.2.5 Initialization

Initialization is the initial signal flow through your Instrument Structure when you open that Instrument. Modules react to Events slightly differently during Initialization than during normal Instrument operation. During this procedure, the Modules are initialized in a certain order and each only once. The Initialization order usually, but not always, depends on the order in which the Modules were placed into the Structure. At the moment of its Initialization, the Module looks at all values that have arrived at its input port(s). This means that in the simple case of an Add Module that calculates the sum of two Constant Modules, the Add Module always sends only one Event during Initialization, namely the sum of the two values at its input port. If the Constants are initialized before the Add Module, then the output of the Add Module during Initialization is the sum of the two constants. If no value has arrived at an input port of a Module that is being initialized, then this value is taken to be “0”.



To see the Initialization order of your Modules in the Structure, either press Ctrl+Alt+E if you are in Windows (Cmd+Alt+E in Mac OS X) or click on the *Options > Debug Structure > Show Event Initialization Order* menu entry. See subsection [14.1.2, REAKTOR's Debugging Tools](#) for more information on REAKTOR's Debugging Tools.

One should take special care with Modules for which the order of Events arriving at its input port is relevant. For example, the Multi Display, Poly Display, Modal Bank, and Sine Bank Modules (among others) need to receive the set of values at their input ports in a specific order before they can fulfill their intended function. In the case of the Multi Display Module, the index of the object needs to be received before the coordinates, color, transparency, and object type values for that object. If there are several objects, the event processing becomes highly non-trivial. Therefore, for more complex Modules it can easily happen that a Module is initialized before all the Events have arrived (in the correct order) to the Module's input ports, causing undesired behavior.



Please refer to chapter [↑12, Working with the Sine and Modal Bank Modules](#) for a more detailed explanation of the Sine Bank and Modal Bank Modules.

Each Module has a specific behavior at Initialization with which you should become familiar. Some Modules send Events at Initialization, others don't. The Merge and Order Modules exhibit special behavior during Initialization, described below. Also, Modules often send the Events from their output ports in a specific order. In the previous subsection you learned the first techniques in controlling Event order. Some of the Modules that pop up when controlling the Structure Initialization are:

- Order Module (*Built-In Module > Event Processing*)
- Value Module (*Built-In Module > Event Processing*)
- Separator Module (*Built-In Module > Event Processing*)
- Merge Module (*Built-In Module > Event Processing*)
- Step Filter Module (*Built-In Module > Event Processing*)
- Iteration Module (*Built-In Module > Event Processing*)
- Snap Value Module (*Built-In Module > Auxiliary*)
- Snap Value Array Module (*Built-In Module > Auxiliary*)
- Bottom line: be mindful of the Initialization behavior of your Structures.

Initialization of the Order Module

One of the Modules that exhibit special behavior during Initialization is the Order Module. The Order Module, as described in the previous subsection, has three output ports, labeled “1”, “2”, and “3”. The Event arriving at the input port of the Order Module is first sent to the “1” output port, and only when the signal connected to that output port has reached

its terminus, does the Order Module send another Event (with the same value) from its “2” output port. How does this work during Initialization? The Order Module only sends an Event from its “1” output port during the actual Initialization procedure. The Events from the “2” and “3” output ports come later.

Let's consider a practical example (see screenshot below). We are looking at the Initialization of three Structures that in principle add together two values, “1” and “2”, which come from two Constant Modules.

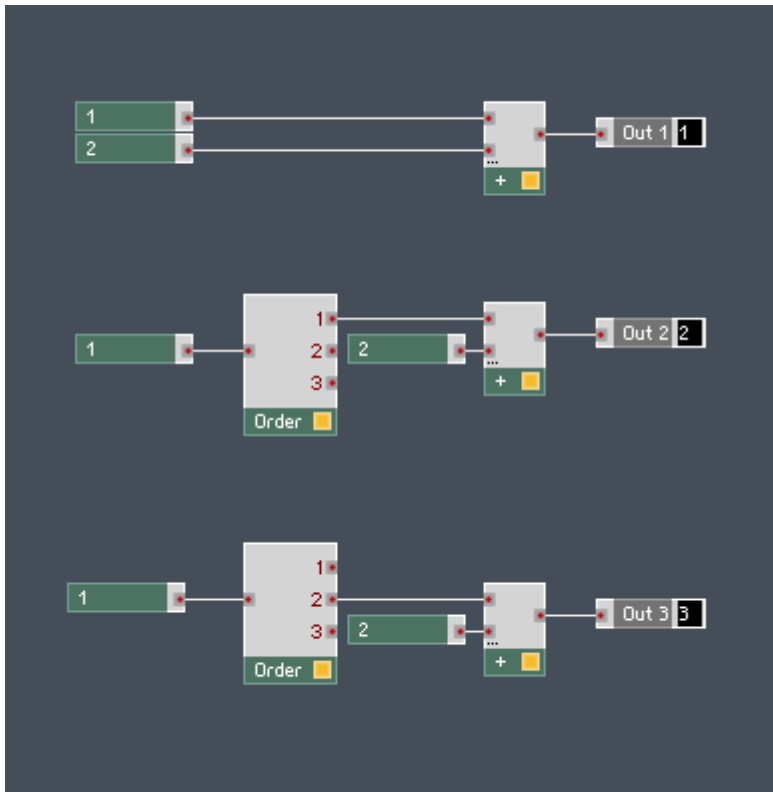


Fig. 9.11 If the input port of the Add Module is connected to a Constant Module or the “1” output port of the Order Module, the Add module sends one Event during Initialization. On the other hand, if the input port is connected to the “2” output port, two Events are sent during Initialization.

Before we can know for sure how this Structure behaves during Initialization, we need to find out the Initialization order of the Modules. To see the Initialization order of your Modules in the Structure, either press Ctrl+Alt+E if you are in Windows (Cmd+Alt+E in Mac OS X) or click on the *Options > Debug Structure > Show Event Initialization Order* menu entry (see subsection [↑14.1.2, REAKTOR's Debugging Tools](#) for more information on REAKTOR's Debugging Tools). If active, your Modules should instead of their Label show their order in the Initialization procedure. The Module with the Initialization number “1” is initialized first, the one with the Initialization number “2” second, and so on. We see in the screenshot below that the Modules lying upstream from the Add Modules are initialized before the Add Modules. This means that for the uppermost Structure, the values “1” and “2” arrive at the Add Module's input ports before the latter initializes and that at Initialization, the Add Module sends one value, namely “3”, the sum of “1” and “2”.

Now let's look at the second Structure in the above and below screenshot. Now there is an Order Module between the Constant “1” and the upper input port of the Add Module. As mentioned above, during Initialization the Order Module forwards the incoming Event at its input port to the “1” output port with no delay. This forwarded Event arrives at the upper input port of the Add Module before the Initialization of the Constant “2” (as can be seen by the Initialization order number “9” of the Order Module versus the “10” of the second Constant Module). As a result, both constants arrive at the two input ports of the Add Module before the latter's Initialization. At the Initialization of the Add Module, it sends one Event from its output, again carrying the value “3”.

Last, let's look at what happens with the bottommost Structure shown in the above and below screenshots. Again there is an Order Module between the “1” Constant Module and the Add Module but this time the “2” output port is connected to the Add Module's input port. Remembering what was said above, we know that the Order Module does not forward any Events to its “2” and “3” output ports but immediately after Initialization. Therefore during Initialization the value at the upper input port of the Add Module stays “0” whereas the bottom input port receives the value “2” and the Add Module sends an Event with the value “2” to its output port. However, the Event from the “2” output port of the Order Module is sent immediately after the Initialization and this in turn triggers another Event to be sent from the output port of the Add Module, this time carrying the value “3”.

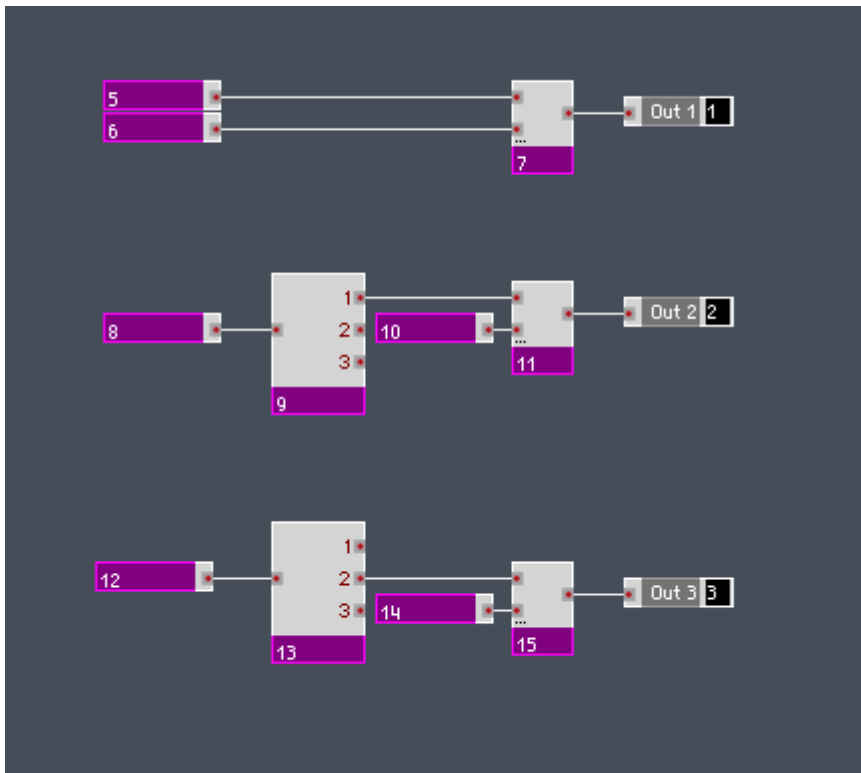


Fig. 9.12 To see the Initialization order of your Modules in the Structure, either press Ctrl+Alt+E if you are in Windows (Cmd +Alt+E in Mac OS X) or click on the Options > Debug Structure > Show Event Initialization Order menu entry.

Initialization of the Merge Module

Another Module that has special behavior at Initialization is the Merge Module. As with all Modules that have one Event output port, it sends only one Event during Initialization. As a result, the Merge Module cannot behave like it does during normal operation, that is, to forward all Events to its output. It only forwards the Event from the bottommost input port to its output port. In the Structure shown below, the Merge Module sends the value “3” to its output during Initialization.

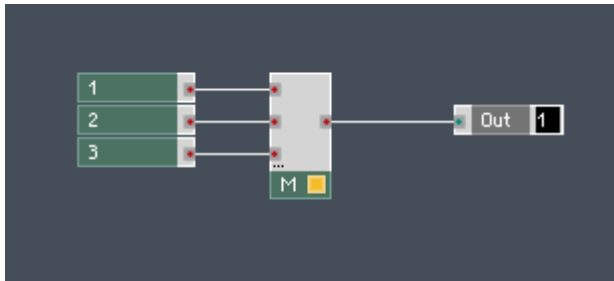


Fig. 9.13 When the Merge Module is initialized, only the Event from the lowest input port is forwarded to the output.

Another special property of the Merge Module is that its ports determine the Initialization order of the Modules connected to them. The rule is that the Modules are always initialized from top to bottom. In the figure below, we see that the Constant Module connected to the top input port of the Merge Module is initialized first, then the Constant Module connected to the second input port, and finally the Constant Module connected to the lowest input port. Remember that the Merge Module only lets the Event from the bottommost input port through. Nevertheless, the Initialization order can affect other parts of the Structure as well if the Modules lying upstream of the Merge Module have other connections as well.

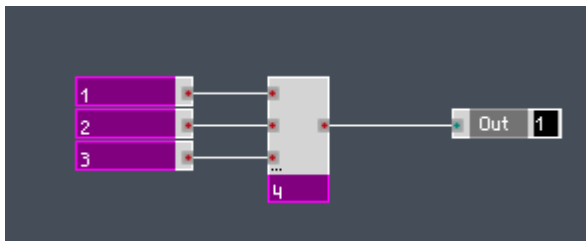


Fig. 9.14 The Initialization order of Modules lying upstream from the Merge Module is determined by the position of the Merge Module's input port that they are connected to.

9.3 Polyphony

In REAKTOR, Polyphony enables you to play several notes at the same time or in Unison Mode, or even use the Voices to carry a number of parallel signals using only one wire. More generally, Polyphony determines how many Voices are processed in parallel by Mod-

ules that operate in Polyphonic Mode. The number of Polyphonic Voices is determined by the Instrument to which the Module belongs. In this section you will learn how to differentiate between Polyphonic and Monophonic signals, how to change the Polyphony settings of your Instrument, and how to utilize the Polyphonic Voices as carriers of parallel signals.

9.3.1 Polyphonic and Monophonic Signals

Most Modules can function either in Polyphonic or Monophonic Mode. Which Mode the Module is operating in, is indicated by the color of the lamp on the bottom right of the Module. A yellow lamp indicates Polyphonic Mode. This means that the output signal of the Module is Polyphonic and the Module accepts Polyphonic input signals. An orange lamp indicates that the Module is Monophonic, meaning that its output signal is Monophonic and that it doesn't accept Polyphonic input signals. All Panel Elements except the Poly Display Module only operate in Monophonic Mode. If a Module can fulfill the same task in Monophonic Mode as it does in Polyphonic Mode, the Module should be set to Monophonic Mode in order to save CPU resources (see "Setting Modules to Mono or Poly" below).

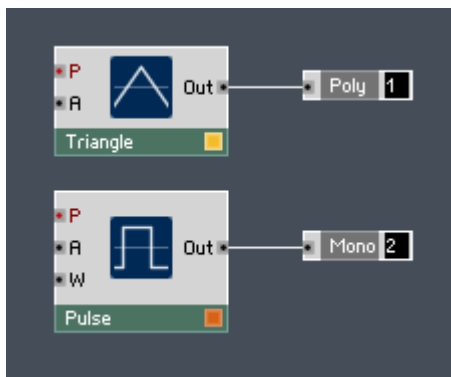


Fig. 9.15 Here the Triangle Module is set to Polyphonic Mode, as seen by the yellow lamp on the bottom right of the Module. The Pulse Module is set to Monophonic Mode. This is indicated by the orange lamp.

Polyphonic Voices

The number of Polyphonic Voices determines how many notes you can simultaneously play with your computer keyboard or MIDI device. For example, if you have set the number of Voices for your Instrument to “4” and the Unison value is set to “1”, then you can play a chord consisting of four notes. If you press a fifth key, one of the Voices from the chord is reassigned to play the new note. See subsection [↑9.3.2, Changing the Number of Polyphonic and Unison Voices](#) for information on Unison playing. To learn how to configure which notes the Voices are distributed to when more keys are pressed than is the Polyphony of the Instrument, please refer to subsection [↑9.3.3, Changing an Instrument's Voice Allocation Settings and Tuning](#).

In next paragraph’s structure, you can see that the output signal of the top Polyphonic Add Module carries four Voices. In this case each Voice carries the value “2”. Also you can see the info on the current notes being played. In the figure, no notes are played, as indicated by the “Note: Off” status of all four Voices. If a note has been played, then for the Voice that the note occupies, the Note Number (in the MIDI scale) is indicated instead of the “Off”. For example, if you have the “Q” key on your computer keyboard pressed, one Voice should have its status change to “Note: 60” with “60” being the Note Number in the MIDI scale.

Polyphony in Modules

For Modules, Polyphony refers to the number of parallel signals with which the Modules operates. These parallel signals are called (Polyphonic) Voices. The number of Voices is determined by the Instrument to which the Module belongs. In the below screenshot, a Structure of an Instrument is shown that has the number of Voices set to “4”. The bottom Monophonic Add Module sends a signal that carries only one Voice but the top Polyphonic Add Module sends a signal from its output port that consists of four parallel signals or Voices. You can process Polyphonic signals in a way that each Voice has different values (see subsection [↑9.3.4, Using Voices for Parallel Processing](#) for more information on this). When connecting a Monophonic signal to a Polyphonic input port, all Polyphonic Voices receive the same value (from the Monophonic signal). For pitch signals this means the Voices in effect play in unison.

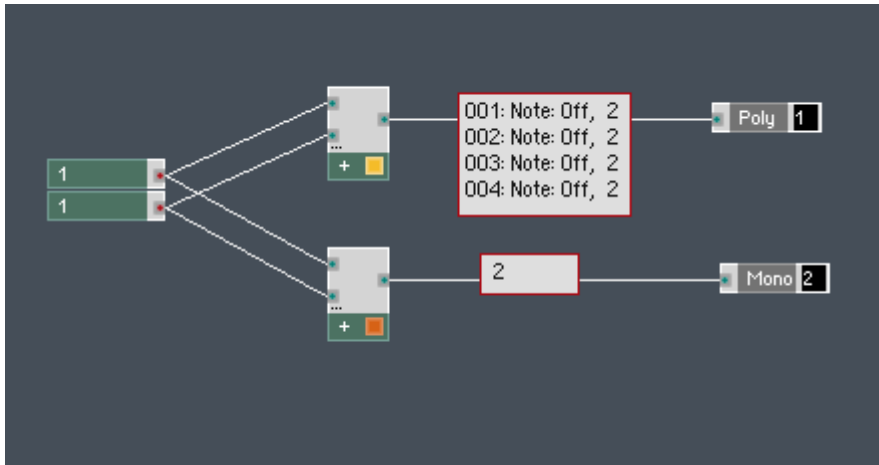


Fig. 9.16 A Structure that belongs to an Instrument which has the number of Voices set to “4”. The upper, Polyphonic Add Module has as its output a signal with four Voices, each carrying a value. In this case all the values are the same, namely “2”. The bottom, Monophonic Add Module has a Monophonic signal as its output; it consists of one Voice carrying the value “2”.

Setting Modules to Mono or Poly

You can switch the operation Mode for a Module either to Mono (Monophonic) or Poly (Polyphonic). There are two ways to do this. The first way, as illustrated below, is by right-clicking the Module (or one of a number of selected Modules, as is the case in the figure) in Structure View and choosing the *Mono* (or *Set Selection to Mono*) or *Poly* (or *Set Selection to Poly*) menu entries, respectively.

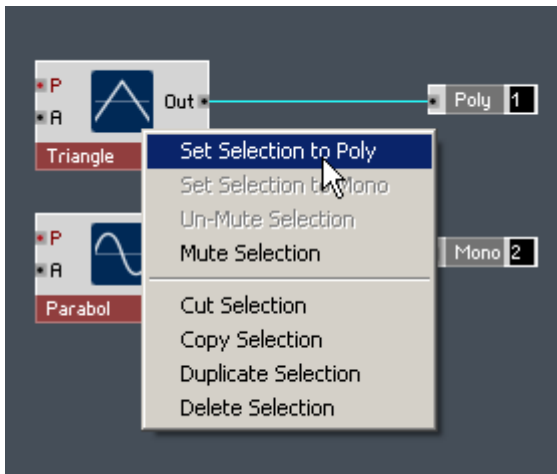


Fig. 9.17 To change a number of Modules in the Structure View to Polyphonic Mode, first select the Modules, then right-click on of them, and then choose the Set Selection to Poly menu entry.

Another way to switch between Polyphonic and Monophonic operation of a Module is by going to the Module's Properties and then going to the Function page. There you will find the **Mono** checkbox which, when engaged, makes the Module Monophonic. A disengaged checkbox means that the Module is Polyphonic.



Fig. 9.18 Engaging the Mono checkbox in the Function Page of the Module Properties makes the Module operate in Monophonic Mode.

Active and Inactive Input Ports

Modules that are set to Mono (operate in Monophonic Mode) cannot accept Polyphonic signals. This means that input ports of Monophonic Modules that are connected to output ports of Polyphonic Modules, with the exception of the Voice Combiner Modules, are muted (become inactive). This is indicated by a red "X" covering the input port.

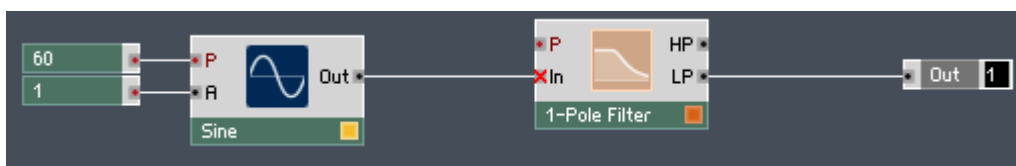


Fig. 9.19 The Polyphonic output signal of the Sine Module is connected to the input port of the Monophonic 1-Pole Filter Module. Since Polyphonic signals are not allowed to enter the input ports of Monophonic Modules, the corresponding input port is muted, as indicated by the red “x” over the port.

To solve the problem of the muted port you can either place an Audio Voice Combiner Module for Audio signals or any of the Event V.C. Modules, depending on your need, between the output and input ports. This is shown in the illustration below. You can find the Voice Combiner Modules in the *Built-In Module > Auxiliary* submenu.

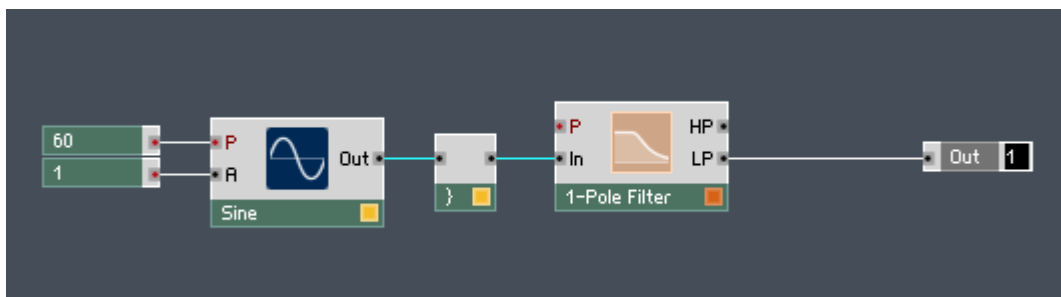


Fig. 9.20 You can turn a Polyphonic Audio signal into a Monophonic Audio signal by using the Audio Voice Combiner Module. Here the Audio Voice Combiner Module adds the signals of all the Polyphonic Voices and uses the sum as a Monophonic signal at its output port

! Instruments can send and receive only Monophonic signals regardless if they are Polyphonic or not. When building a Structure inside an Instrument and routing a signal to its output ports you have to make sure that the signal is Monophonic. This is usually achieved by placing an Audio Voice Combiner Module in front of the Out Terminal.

Muting Ports

You can also mute individual ports in REAKTOR. This is simply done by right-clicking a port and selecting the *Mute Port* menu entry, as shown below. To unmute a port, just follow the same procedure.

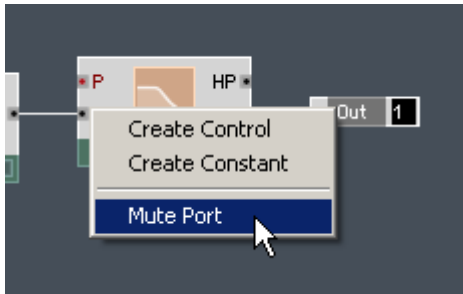


Fig. 9.21 To mute a port, right-click the port and choose the Mute Port menu entry

9.3.2 Changing the Number of Polyphonic and Unison Voices

The number of Polyphonic Voices determines how many notes you can simultaneously play with your computer keyboard or MIDI device. It is equal to the number of parallel Voices processed by Polyphonic Modules. You can also make several Polyphonic Voices be allocated to one note and, by slightly detuning each Voice, fatten up the sound. This does not change the number of Voices processed by the Polyphonic Modules.

When the number of Unison Voices is set to “1”, the number of different notes that you can play equals the number of Polyphonic Voices. When set to “2”, each note has two Voices and with a Polyphonic Voice number of “8” you can in all play $8 \div 2 = 4$ different notes simultaneously. The number of Polyphonic and Unison Voices can be set in the Function page of the Instrument Properties. This subsection covers how to change these values, including the detuning of the Unison Voices.

! Increasing the number of Polyphonic Voices generally also increases CPU load.

Changing the Number of Polyphonic Voices

1. In the Function page of the Instrument Properties, double-click the Voices edit field.

FUNCTION

2. Enter the desired number of Polyphonic Voices.

VOICES
4

Changing the Number of Unison Voices

The behavior of Unison playing of the Instrument is specified in the Function page of the Instrument Properties. The maximum and minimum number of Unison Voices can be changed by double-clicking the **Max Voices** and **Min Voices** edit fields in the Unison area (shown in the figure below) and entering the desired value. The entered values have the following effect:

- **Max Voices:** Specifies the maximum number of Unison Voices that an Instrument can play per note. If, for example, you set the **Max Voices** edit field to the value “3”, the Instrument will play a maximum of 3 Unison Voices per note. The amount of detuning between the Voices is set with the **Spread** edit field (see the next header).
- **Min Voices:** Specifies the minimum number of Unison Voices that an Instrument can play per note. If, for example, you set the **Min Voices** edit field to the value “2”, the Instrument will play a minimum of 2 Unison Voices per note.

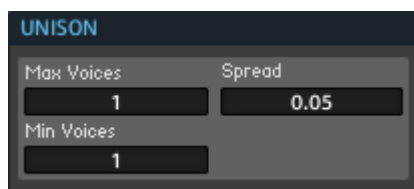


Fig. 9.22 The Unison area of the Instrument's Function page lets you specify the minimum and maximum number of Unison Voices and the detuning of these Unison Voices.

The Polyphonic Voices, Max Voices, and Min Voices values are interdependent. The Polyphonic Voices value specifies the total number of Polyphonic Voices that the Instrument can play (described under the previous header). If the Instrument does not make use of Unison playing, the **Max Voices** and **Min Voices** edit fields are both set to “1”. If the Instrument does make use of Unison playing, then these edit fields specify the maximum and minimum number of Unison Voices that the Instrument can play per note.

For example, say the Polyphonic Voices, Max Voices, and Min Voices settings are “24”, “1”, and “1”, respectively. Then the Instrument could play up to 24 notes at the same time, with no Unison effect. If the settings were “24”, “3”, and “3”, then the Instrument could play up to 8 notes at the same time with 3 Unison Voices per note ($24 \div 3 = 8$).

On the other hand, if the settings were “24”, “3”, and “2”, then the Instrument could play up to 8 notes at the same time with 3 Unison Voices per note ($24 \div 3 = 8$), or up to 12 notes at the same time with 2 Unison Voices per note ($24 \div 2 = 12$).

If the Max Voices and Min Voices are different as in the last part of the example above, the REAKTOR automatically switches between the maximum and minimum number of Unison Voices depending on how many notes are being played at the same time.

Voice Settings in Instrument Header

The Voices settings of an Instrument are displayed in the Instrument Header, as shown in the screenshot below. The first number after the label “Voices” signifies the number of Polyphonic Voices. The second number displays the Max Voices value. It appears only if it is greater than the value “1”.



Fig. 9.1 The Voices display in the Instrument Header displays the number of Polyphonic Voices (here: 6) and, if set to a value greater than “1”, the Max Voices value (here: 3).

Changing the Unison Spread

The “Unison Spread” feature can be used in conjunction with Unison playing to create fatter sounds. The Spread edit field determines the amount of detuning between Voices that are playing in Unison. It is only effective when the number of Unison Voices playing is greater than one. To change the Spread value, go to the Function page of the Instrument and double-click the Spread edit field, and enter the new value.

Locking the Voice Settings

You can lock the Voice Settings of the Instrument so that the Voices, Max Voices, and Min Voices edit fields in the Function page are not editable. This can be useful when your Instrument's signal processing explicitly relies on a fix number of Voices and you want to keep potential Instrument users from messing with the Voice settings.

- To lock the Voice settings, go to the Function page of the Instrument and engage the Lock Voices checkbox.

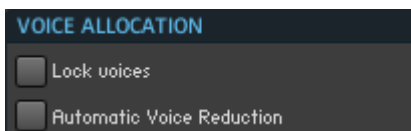


Fig. 9.23 In the top part of the Voice Allocation area you can find checkboxes to lock the Voice settings of an Instrument and to activate the “Automatic Voice Reduction” feature.

Automatic Voice Reduction

Increasing the number of Polyphonic Voices also increases the CPU load. If you plan to share your Instrument with other users it might be a good idea to keep inexperienced users from increasing the Voice number beyond what their CPU can handle. The “Automatic Voice Reduction” feature reduces the number of Voices if the CPU load crosses a certain threshold. REAKTOR compares its CPU usage with the Maximum CPU Usage value set in the CPU Usage Tab of the REAKTOR Preferences dialog box and if that threshold is crossed, it searches for loaded Instruments for which the “Automatic Voice Reduction” feature has been activated. If such an Instrument is found, REAKTOR automatically reduces its Voice number until the maximum CPU usage is below the threshold. Setting the Maximum CPU Usage edit field to “100” globally disables this feature. To activate the “Automatic Voice Reduction” feature, perform the following steps:

1. First, define the CPU threshold beyond which Automatic Voice Reduction kicks in. Choose the *File > Preferences ...* menu entry to enter the REAKTOR Preferences dialog box.
2. Click on the CPU Usage tab.



3. Enter the CPU threshold (in percent) into the Max CPU Usage edit field.
4. Press the Close button to exit the REAKTOR Preferences dialog box.
5. Go to the Instrument's Function page for which you wish to activate the “Automatic Voice Reduction” feature.



6. Engage the Automatic Voice Reduction checkbox.

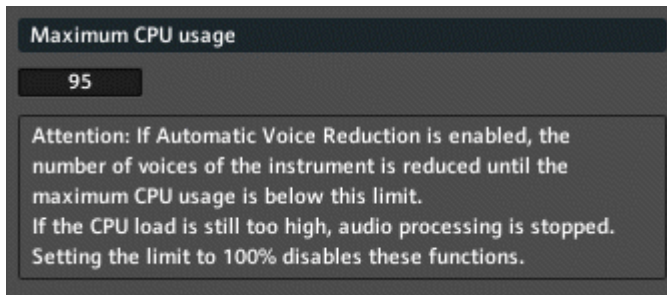


Fig. 9.24 Use the Maximum CPU Usage edit field in the REAKTOR Preferences dialog box to set the CPU threshold beyond which the “Automatic Voice Reduction” feature kicks in.

9.3.3 Changing an Instrument's Voice Allocation Settings and Tuning

This subsection discusses how to change your Instruments tuning in respect to other Instruments, how to define to which notes available Polyphonic Voices are allocated, and how to set your Instrument to be a Voice & MIDI Slave to another Instrument.

Instrument Tuning

By changing the tuning of your Instrument you can adjust its pitch in respect to the Ensemble. The value is specified in fractional semitone units (where 12 semitones comprise an octave). Positive values raise the pitch; negative values lower it. A typical application is to detune two Instruments to get a fatter sound. A good value for this is “0.05” which is equivalent to 5 cents or 1 / 20th of a semitone. For the top-level Instrument (Ensemble) the tuning is only in respect to the incoming MIDI pitch value (after it has been processed according to the Instrument's MIDI In settings).

! The tuning of an Instrument is applied after the incoming MIDI pitch signal has been processed according to the settings in the MIDI In area of the Instrument's Connections page, in particular the Note Shift value. Please refer to subsection [↑13.2.1, MIDI In](#) for more information on configuring an Instrument's MIDI In settings.

- To change the tuning of an Instrument, go to its Function page and enter the new value into the Tune edit field.

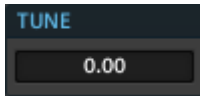


Fig. 9.25 The tuning of an Instrument can be changed using the Tune edit field in the Instrument's Function page.

Voice Assignment Settings

For each Instrument you can specify how it assigns its Polyphonic Voices to incoming MIDI notes (when a new note has been played). This is done using the Assign radio button selector and Reassign checkbox. The different Assignment Modes function as follows:

- **Oldest:** When the Oldest radio button has been selected, the Voice which has been held longest is stopped and assigned to the new note. This is the most common Voice Assignment strategy.
- **Newest:** With the Newest radio button selected, the most recently played Voice is stopped and assigned to the new note. This can be useful for playing a melody over held notes because none of the held Voices are stopped and reassigned.
- **Nearest:** When the Nearest radio button has been selected, the Voice assigned to a note closest in pitch to the new note is stopped and assigned to the new note. This is good when playing an Instrument that utilizes the polyphonic portamento (glide) way of playing.
- The Reassign checkbox determines what happens when the same note is played again. Either the Voice already assigned to the note is reused or another Voice is used. The Reassign Mode is good for making efficient use of a limited number of Voices and it is also what you are used to from playing the piano.

To activate the Reassign Mode, engage the Reassign checkbox in the Instrument's Function page.



Fig. 9.26 The Voice Assignment settings of an Instrument are configured in the Voice Allocation area of the Instrument's Function page

Setting Instrument to Voice and MIDI Slave

In some cases you have an Ensemble that consists of several polyphonic synthesizers that are meant to be played in a stacked manner. For example, one synthesizer provides the warm virtual analogue fundament while a second Instrument is an FM synthesizer that adds a more digital element to the sound. In such a case you might want the Polyphonic Voices and the incoming MIDI messages to be handled in the same way (since you want to play the two Instruments as one). For this purpose REAKTOR allows you to choose one Instrument to have its Voice and MIDI In settings overridden by the settings of another Instrument. . These certain settings are overridden and cannot be edited. Furthermore, incoming MIDI Notes are assigned to the exact same Voices for the Slave Instruments as for the Master Instrument. These things are achieved by activating the “Voice & MIDI Slave” feature for that Instrument

- To activate the “Voice & MIDI Slave” feature for an Instrument, choose its Master Instrument from the Voice & MIDI Slave to drop-down menu.

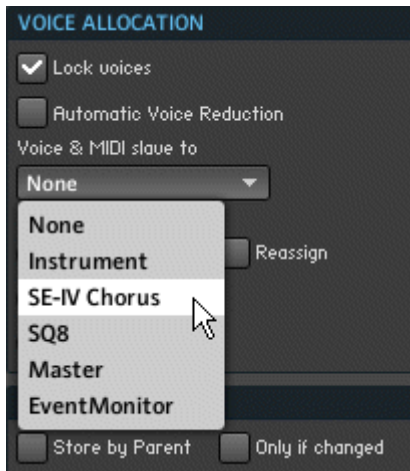


Fig. 9.27 Select the Master Instrument from the Voice & MIDI Slave to drop-down menu

The Voice Settings that are acquired from the Master Instrument are:

- Voices, Max Voices, and Min Voices edit fields.
- Lock Voices and Automatic Voice Reduction
- The Assign radio button selector and the Reassign checkbox

The MIDI In Settings that are overridden for the Slave Instrument are:

- External MIDI In Device (replaced by the entry “*Slave Mode*”)
- MIDI In Channel
- Upper Limit and Lower Limit edit fields
- Sustain Ctrl and Hold Ctrl edit fields

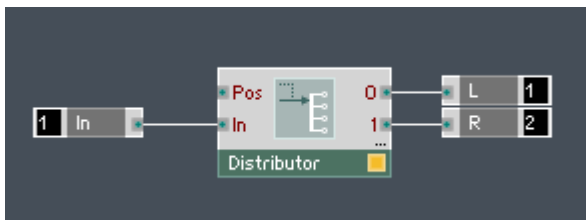
9.3.4 Using Voices for Parallel Processing

Polyphonic signals in REAKTOR consist of a number of parallel Voices, as mentioned in subsection [↑9.3.1, Polyphonic and Monophonic Signals](#). These signals can be used to specify different values for all the different Voices. For example, you could position each Voice differently in the stereo (or even 5.1 surround sound) panorama field. This subsection is a short tutorial on how to build a Macro that does just that: a stereo spread structure that takes a Polyphonic Audio or Event signal and spreads it out in the stereo panorama. Just follow these instructions:

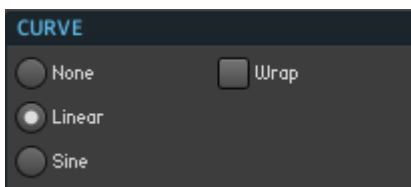
1. To start building this structure, create a Macro with an “In” input port and two output ports labeled “L” and “R” (for the left and right stereo channels, respectively).
2. To give the Macro a name, go to the Properties of the Macro and change the Label edit field to “Stereo Spread”. The outside of the Macro is shown below.



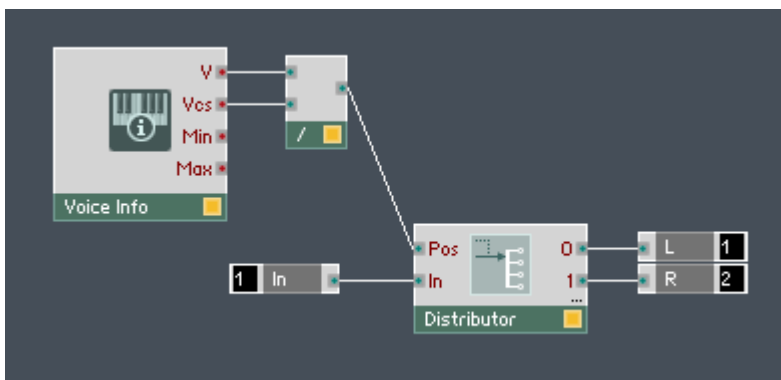
3. We will use the Distributor Module to control the panning of the Voices. Place a Distributor Module into the structure.
4. Route the “O” output of the Distributor Module to the “L” Out Port.
5. Hold down Ctrl if you are using Windows (Cmd for Mac OS X) while dragging a wire from the “R” Out Port to the three dots on the bottom right of the Distributor Module. This will create an additional output port for the right stereo channel.
6. Connect the “In” input port of the Distributor Module to the “In” In Port. What you have now should be comparable to the screenshot below. The Structure now distributes the signal at the “In” In Port between the “L” and “R” Out Ports, depending on the value that is sent to the “Pos” input port of the Distributor Module. If Pos = 0, the signal is exclusively routed to the “L” Out Port and if Pos = 1, the signal is exclusively routed to the “R” Out Port. For intermediary values between “0” and “1” for “Pos” (for example, Pos = 0.24) the distribution depends on the interpolation curve set with the Curve selector in the Function page of the Distributor Module.



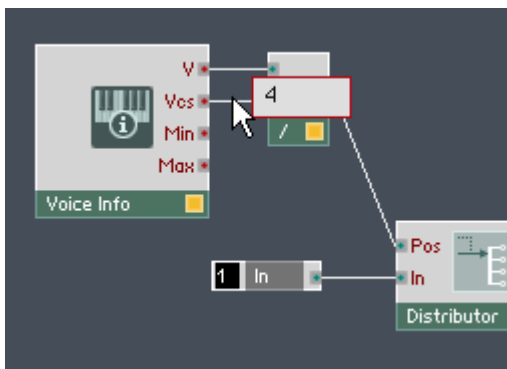
7. For the Voices to be spread out more towards the edge of the panorama, click on the Sine Interpolation box of the Curve selector in the Function page of the Distributor Module.



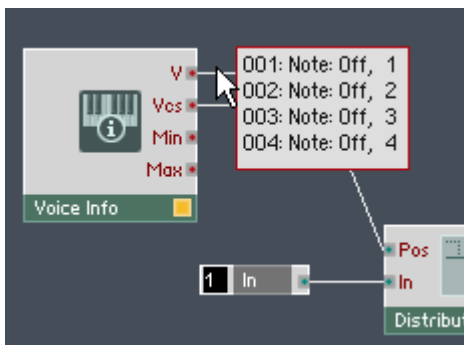
8. Now you need to assign each Voice a different pan setting so that the Voices are spread out in the stereo panorama. For you this means that the value arriving at the “Pos” input port of the Distributor Module needs to be different for each Voice. Furthermore, we want the mechanism that assigns each Voice its “Pos” value to be independent of the number of Voices that the Instrument is using. For this purpose, insert the Voice Info Module and a Divide Module into the Structure.
9. Connect the “V” output port of the Voice Info Module to the upper input port of the Divide Module.
10. Next connect the “Vcs” output port of the Voice Info Module to the lower input port of the Divide Module.
11. Now connect the output port of the Divide Module to the “Pos” input port of the Distributor Module. Your structure should look similar to the screenshot below. The value sent from the “V” output port of the Voice Info Module is different for each Voice. Each Voice carries the value of its Voice number. Furthermore, the “Vcs” output port of the Voice Info Module sends the number of Voices of the Instrument as its value (this is value is carried by all Voices).



12. Let us look at this information for each Voice separately. Say you have an Instrument with the Polyphony, or number of Voices, set to “4”. This means that at “Vcs” output port of the Voice Info Module all Voices carry the value “4”.



13. Look at the “V” output port of the Voice Info Module. The first Voice carries the value “1”, the second Voice carries the value “2”, the third Voice carries the value “3”, and the fourth Voice carries the value “4”.



14. Now look at the output of the Divide Module. The first Voice carries the value “ $1 \div 4$ ” or “0.25”, the second Voice carries “ $2 \div 4$ ” or “0.5”, the third Voice carries “ $3 \div 4$ ” or “0.75”, and the fourth Voice carries “ $4 \div 4$ ” or “1”. Since each Voice receives a different value between “0” and “1” at the “Pos” input port of the Distributor Mod-

ule, each Voice is panned to a different coordinate in the stereo panorama. Clicking on the Linear Interpolation selector in the Function page of the Distributor Module you can move the panoramic distribution more to the center.

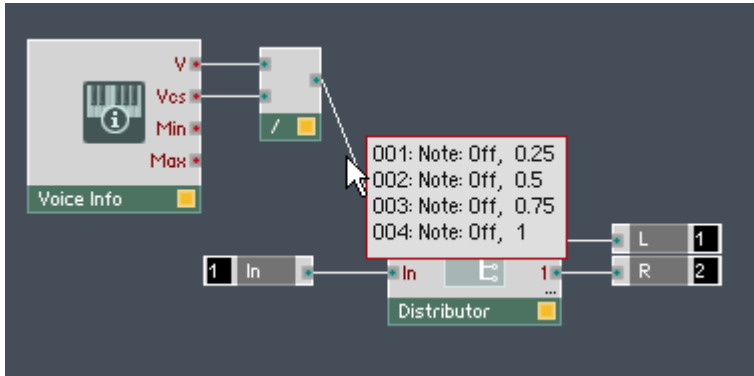


Fig. 9.28 Resulting structure

The final result should look like the Structure shown above. You can see that the Polyphonic signal going from the output of the Divide Module to the “Pos” input port has each Voice carrying different values between “0” and “1”. This means that each Voice receives a different pan setting and is distributed differently between the “L” and “R” Out Ports of the Stereo Spread Macro.

9.4 Different Value Ranges and Scales in REAKTOR

The most common parameters such as pitch, amplitude, and envelope times are specified using certain value ranges and scales. For example, you might want to specify the frequency of an oscillator in Hertz in the case of an LFO and in the logarithmic pitch scale when the oscillator has to follow the pitch being played on the MIDI (or computer) keyboard. This section will give an overview of the different value scales and ranges and how they correspond to the standard MIDI scale [0 ... 127].

9.4.1 The MIDI Scale

MIDI is a communication protocol that operates with integers in the value range [0 ... 127]. Several features in REAKTOR are compatible with this range. For example, you can recall Snapshots (see chapter [↑4, Snapshots](#)) and set the pitch and amplitude levels of oscillators via MIDI messages from remote devices. MIDI values are sent to the Structure through the output ports of so-called MIDI Source Modules. An example of such a Module is the Note Pitch Module (found in the *Built-In Module > MIDI In* submenu). It outputs the pitch values in MIDI scale that are played either on a remote MIDI device or your computer keyboard. The screenshot below shows the Note Pitch Module sending a pitch value to a Sine Module.



For more information on the MIDI interface of REAKTOR Instruments, see section [↑13.2, Setting up MIDI Connections](#).

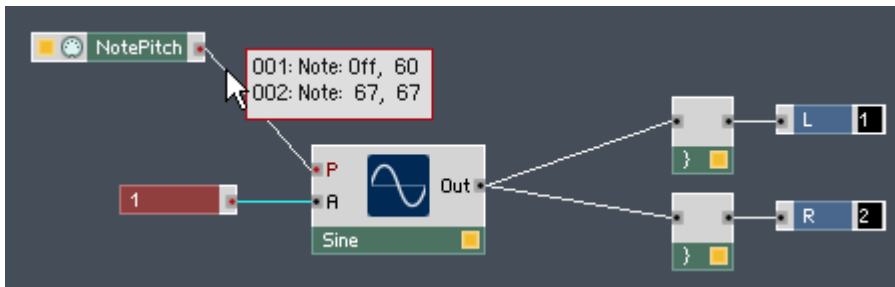


Fig. 9.29 Note Pitch module sending a pitch value

See above, the key corresponding to the MIDI note A4 is pressed on a MIDI device or on the computer keyboard. The result is that the value “69” is sent from the output port of the Note Pitch Module. This value can then be used by the Sine Module at the “P” input port to generate a Sine signal of 440 Hz.

Range of Values

For controls and MIDI Source Modules the range of the output signal is scaled to the range [Min ... Max]. The values for Min and Max are set with the corresponding edit fields in the Function page of the Module's Properties.



Fig. 9.30 The Function button



Fig. 9.31 The “Min”, “Max”, “Lower Limit”, and “Upper Limit” values can be set in the corresponding edit fields of the MIDI Source Module's Function page.

For MIDI Source Modules the range can also be limited with the *Lower Limit* and *Upper Limit* edit fields, also in the Function page (shown above). The output value of the MIDI Source Module is limited to “Min” for incoming MIDI values below “Lower Limit” and to “Max” for incoming MIDI values above “Upper Limit”. The range between the two limits is interpolated linearly between “Min” and “Max” (see below).

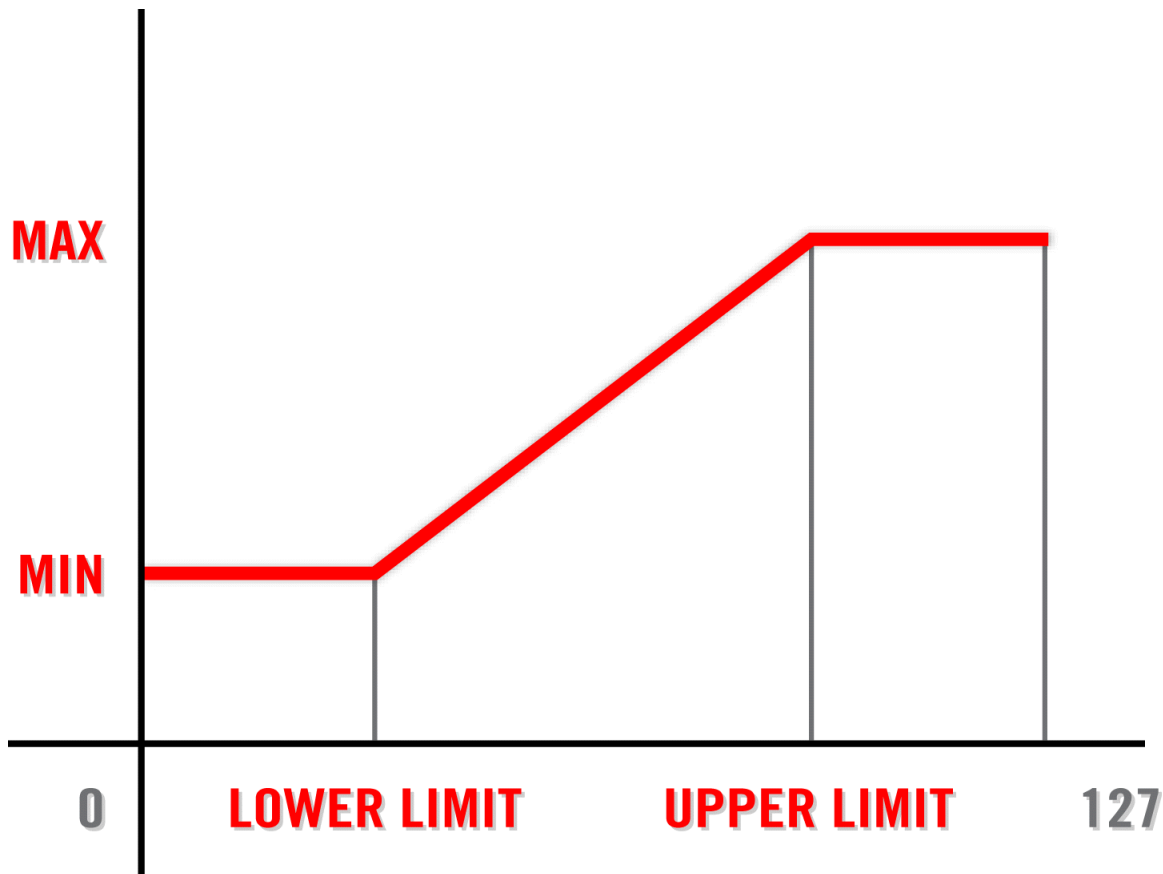


Fig. 9.32 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing values from the MIDI source Module. The values are interpolated linearly between “Min” and “Max”.

The values for the Lower Limit and Upper Limit edit fields lie in the range [0 ... 127]. Also, the value for Upper Limit edit field must be greater than that set in the Lower Limit edit field. However, the value in the Max edit field can be smaller than the value in the Min edit field to achieve inverted operation. If opposite characteristics are set for two sources, a crossfade effect can be programmed (see below).

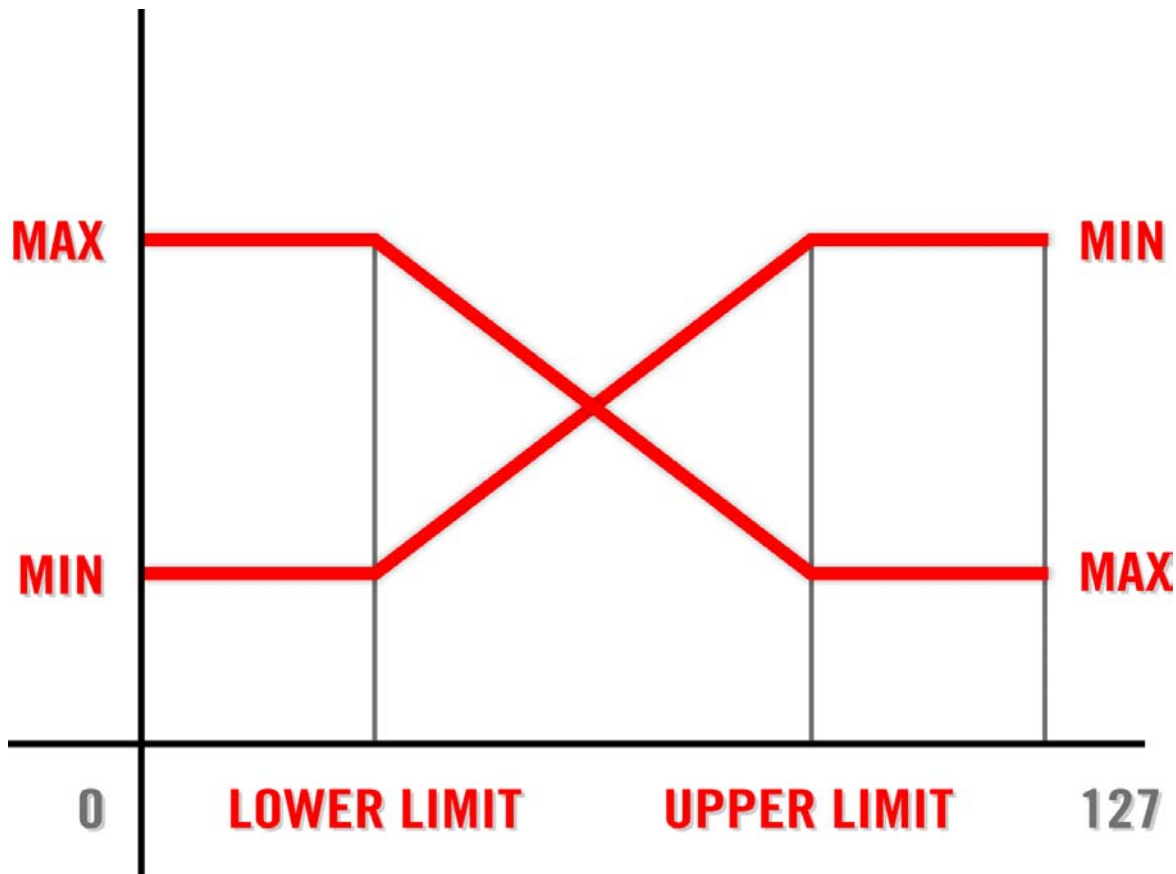


Fig. 9.33 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing values from the MIDI source Module. Setting opposite characteristics for two sources creates a crossfade effect between them, going from low to high MIDI note values.

A switch with an adjustable threshold level can be emulated by setting the **Lower Limit** and **Upper Limit** edit fields to neighboring MIDI values, for example, “63” and “64”. When the input value to such a source is “64”, the value in the **Min** edit field will be output. Otherwise, the output value is the value in the **Max** edit field.

9.4.2 The Linear Frequency and Logarithmic Pitch Scales

There are two scales with which frequencies are specified in REAKTOR: the linear frequency scale and the logarithmic pitch scale. You can convert signals in the logarithmic pitch scale to the linear frequency scale by using the Exp (P-to-F) Module. Signals in the linear frequency scale can be converted to the logarithmic pitch scale using the Log (F-to-P) Modules.

Linear Frequency Scale

In the linear frequency scale the frequency is directly determined by a value “F”. This means that an “F” value of “5000” corresponds to 5000 Hz where Hz (read as “Hertz”) is the unit for cycles per second. A frequency value of 440 Hz corresponds to the MIDI note A4 which a logarithmic pitch value of “69”. Input ports of Modules that accept values in the linear frequency scale are labeled “F” (as in “frequency”). The LFO Module, frequency modulation input ports of filter and oscillators, and Standard Core Macros work in the linear frequency scale.

Logarithmic Pitch Scale

The second and more musical scale for frequencies is the logarithmic pitch scale. This scale is also used by the MIDI protocol for pitch values, with the units being semitones. There are 12 semitones in an octave and an octave corresponds to a doubling of frequency. For example, the value “69” in the logarithmic pitch scale is sent by the MIDI note A4 and corresponds to a frequency of 440 Hz. The MIDI note an octave higher, A5, has twice the frequency (880 Hz) and has the value “81” (12 semitones higher than “69”). The input ports of Modules that accept values in the logarithmic pitch scale are labeled “P” (as in “pitch”). The Oscillator Modules, Filter Modules, and Sampler Modules work in the logarithmic pitch scale. To roughly calculate the frequency “F” (in Hz) from the logarithmic pitch value “P”, use the following formula: $F = 2^{(p / 12 + 3.031)}$.

9.4.3 The Linear Amplitude and Logarithmic Decibel Scales

Similar to frequencies, there are two scales with which amplitude factors are specified in REAKTOR: the linear amplitude scale and the logarithmic amplitude scale. You can convert signals in the logarithmic amplitude scale to the linear amplitude scale by using the Exp (Lvl-to-A) Module. Signals in the linear amplitude scale can be converted to the logarithmic amplitude scale using the Log (A-to-Lvl) Modules.

Linear Amplitude Scale

In the linear amplitude scale the frequency is directly determined by a value “A”. This means that an “A” value of “0.5” corresponds to an amplitude factor of “0.5”. Input ports of Modules that accept values in the linear amplitude scale are labeled “A” (as in “amplitude”). An Oscillator Module that has the value “0.5” at its “A” input port, sends an oscillator waveform with amplitude “0.5” from its output. The Oscillator Modules, Sampler Modules, and Envelopes work in the linear amplitude scale.

Logarithmic Amplitude Scale

The scale that best corresponds to human amplitude perception is the logarithmic amplitude scale. The units for the logarithmic amplitude scale are decibel. It should be noted that the decibel scale is relative scale. For example, a reduction in amplitude corresponding to -6 dB in the logarithmic amplitude scale corresponds to a halving of the amplitude, that is, a multiplication by a factor of “0.5”. The input ports of Modules that accept values in the logarithmic amplitude scale are labeled “Lvl” (as in “amplitude level”). The Mixer Modules and some Standard Core Macros work in the logarithmic amplitude scale. To calculate the amplitude level “L” (in decibel) from the ratio of amplitudes “A1” and “A2”, use the following formula: $L = 20 \text{ Log} [A1 / A2]$ where “Log” is the logarithm with base “10”.

10 Working with Sampler Modules

If the source of your Audio signal in REAKTOR is not an oscillator or an external audio input, then it must be a Sampler Module. This chapter will give you an introduction to the Sampler Modules which you find in the Primary Level. You will see that these Modules go well beyond basic Sample playback. Exemplary more advanced functionalities are granular synthesis and wavetable synthesis. First you will get an overview of loading and managing samples for Sampler Modules. Then the basic differences between the different Modules are highlighted and after that you will get a more in-depth description about the Function and View pages of these Modules.



For more information on specific Sampler Modules, please consult the Module Reference manual that is available as a PDF file in the subfolder “Documentation” of REAKTOR's installation folder.

10.1 Managing Your Samples and RAM

Sample files are external to REAKTOR. This means that before a Sampler Module can use a sample file, the entire file needs to be loaded into RAM. Since RAM is a finite resource of your computer, managing your samples with regard to it becomes important. Additionally, you have the option to store samples and their settings with the Ensemble or sample. The following subsections will discuss these issues and options in more detail.



For more general information regarding sample management in the Sample Map Editor, see the chapter 7.

10.1.1 Sample Files and RAM

REAKTOR's Sampler Modules can use mono or stereo .wav, .aif, or .aiff sample files of any Sample Rate and bit depth. If a sample that is being loaded contains any loop or key-board-allocation information, REAKTOR finds and processes it.

As mentioned in the introduction to this section, samples must be loaded into RAM (Random Access Memory) before they can be played back by a Sampler Module. This RAM memory is actual, not virtual (see below). Regardless of the bit depth of the sample file (8-

bit, 16-bit, and so on), REAKTOR converts its audio information to 32-bit for internal processing. To get an idea of the memory needs for audio files, look at the following figures: one minute of stereo 32-bit audio at a typical Sample Rate of 44'100 Hertz uses 20 MB of RAM (see the header below for a more detailed calculation). Thus, large sample files can use huge amounts of RAM.

Systems that use virtual RAM (the default in Windows, but not in OS X) can allocate a great deal more RAM than is physically available; hence the term "virtual". In such systems, REAKTOR does not display an error message when you load a sample file which is so large that it requires more actual RAM than is physically available. Nevertheless, a message might appear later, informing you that the CPU is overloaded. This error message appears because REAKTOR cannot access the sample audio data on the hard drive fast enough (after first having tried to access it in RAM and not having found it there). Frequently the message is preceded by an unintentional granular sound artifact caused by interruptions of the audio data stream. This situation, particularly undesirable during live performances, can only be avoided by adding more RAM to your system.

Calculating RAM Usage of Sample Files

To calculate the amount of RAM that a sample will use, do the following math. First you need to know the length of your sample and the Sample Rate at which the sample was recorded at. A Sample Rate of 44'100 Hertz means that each second of a sample file contains 44'100 samples. Since for internal processing the audio information contained in the sample files is converted to 32-bit, each sample uses up 4 bytes of RAM. Hence, the amount of RAM that a one second long sample file uses up at a sampling rate of 44'100 Hertz is $44'100 * 4 = 176'400$ bytes, which is a little bit more than 172 Kb (a kilobyte has 1024 bytes). It follows, that for a rough estimate of the total RAM usage of your sample at 44'100 Hertz Sample Rate, you should multiply the length of your sample in seconds by 172 Kb / sec.

10.1.2 Multiple Use of Identical Samples

If one sample is used by several Sampler Modules in an Ensemble, all the Modules access the same sample as it is stored in RAM. This means that you can load the same sample in as many Sampler Modules as you like without increasing the RAM usage by any appreciable amount. Similarly, the number of Polyphonic Voices used by a Sampler Module is irrelevant as far as RAM is concerned.

REAKTOR identifies a sample using the pathname (directory and filename) of the sample file from which the sample was loaded. When a new sample is loaded, REAKTOR searches through the list of already loaded sample files. If it finds one with the same pathname, REAKTOR reuses it instead of loading the new sample file.

10.1.3 Missing Samples

By default, REAKTOR saves the pathnames of the samples that you load into a Sampler Module, not the actual sample files. If any of these sample files are deleted, renamed, or moved after the Ensemble has been saved, they will not be available when the Ensemble is reopened. If this happens, you will get an error message upon opening the Ensemble informing you that the sample files are missing. These samples are labeled as “missing” in the File column of the Sample Map Editor (see chapter [↑6, Sample Map Editor](#) for more information on the Sample Map Editor). To relocate or replace missing samples, select the sample in the Sample Map Editor and click on the *Replace* menu entry in the Edit drop-down menu.

10.1.4 Storing Samples with Modules

You can avoid the potential problem of missing samples by telling REAKTOR to store the samples with the Modules when you save the Ensemble. To do this, follow these instructions:

1. First, make sure you have a Module from the *Built-In Module > Sampler* menu placed into the Structure.
2. Next, double-click on the Module to open the Module Properties.
3. Click on the *Function* button.

FUNCTION

4. Last, engage the *Store Map with Module* checkbox.
5. Storing the Sample Map with the Module saves a copy of all the sample files loaded in the Module's Sample Map with the Module and hence the Ensemble. When you reopen the Ensemble, REAKTOR will load the Module's sample files directly from the Ensemble file rather than loading the original files from their individual locations on your hard disk.

6. The main advantage of storing samples with a Sampler Module is that you can be sure that these samples will still be there when you open the Ensemble or when another user, on a different computer, opens the Ensemble. However, the disadvantage to activating this feature is that depending on the sizes of your sample files, you might end up with a very large Ensemble file size.

! A common mistake is to save a sampler Instrument without having engaged the *Store Map with Module* checkbox in the Function page of the Sampler Module(s), and then to share this Instrument with other REAKTOR users (for example, in the REAKTOR User Library). When another user opens the Instrument, the sample files will be missing. Bottom line: remember to engage the *Store map with Module* checkbox in all Sampler Modules before saving (and sharing) a “sampler” Instrument.

10.1.5 Sample Analysis

Some Sampler Modules like the Resynth, Pitch Former, and Beat Loop Modules perform real-time resynthesis of sample files. When you load a sample in one of these Modules, REAKTOR displays a message dialog asking if you want to save the analysis data into the sample file. Read this message carefully before deciding what to do; it explains the potential danger of embedding analysis data into your sample file. Note that REAKTOR can only save analysis data into a sample file which is not write-protected.

! Once REAKTOR has saved analysis data into a sample file, it assumes that this file has been fully analyzed. This can cause problems, for example, if you modify an analyzed sample file and then load it into one of the resynthesis Modules. REAKTOR assumes that the file is still fully analyzed (though it isn't, because you modified it). To get around this, simply rename an analyzed sample file when you modify it. Then, when you load it into a resynthesis Module, REAKTOR will re-analyze it.

10.2 Differences between the Sampler Modules

All the Sampler Modules, beside the Lookup Module, can be grouped into four categories. This arrangement is shown below. These categories are based on the relationship between the sample playback pitch and playback speed. In this section you will learn about the general differences between the different Sample Modules.

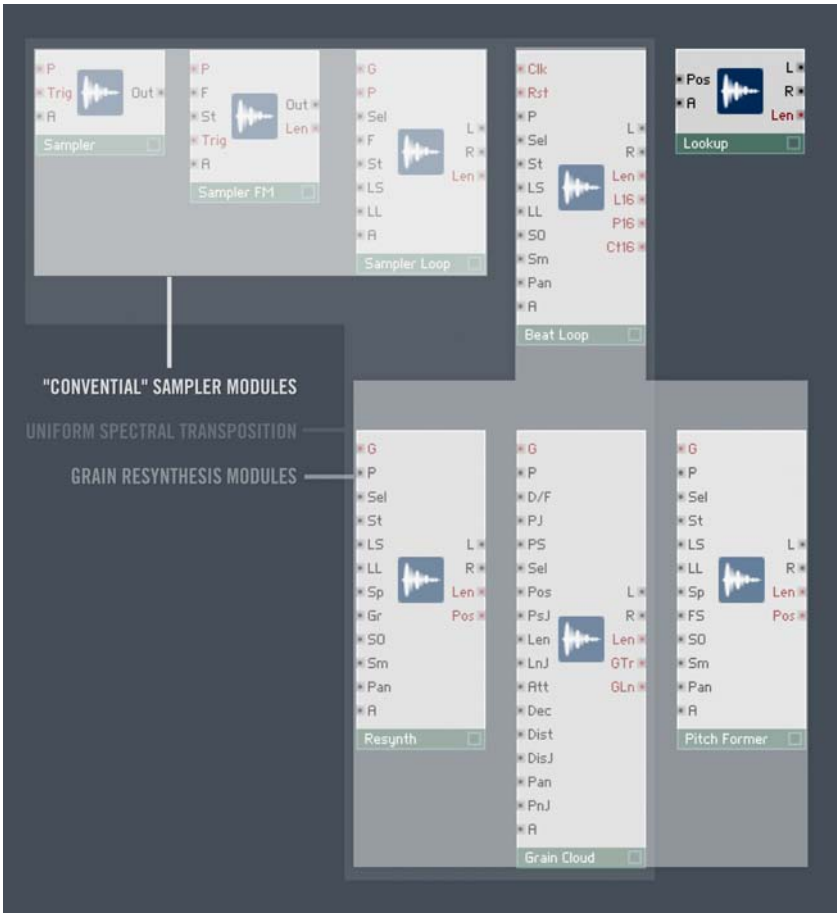


Fig. 10.1 All the Sampler Modules, beside the Lookup Module, can be grouped into four categories based on the relationship between the sample playback pitch and playback speed.

10.2.1 “Conventional” Sampler Modules

One can put the Sampler, Sampler FM, and Sampler Loop Modules into the “conventional” Sampler Modules category, as seen in the screenshot above. These Modules are used for real-time polyphonic and transposed playback of samples in Sample Maps and are similar to conventional hardware samplers. In REAKTOR, Sampler Modules maintain a pointer

for each Voice. This pointer points to the current position in the sample and the value of the Audio signal at the Module's output port is always the same as the amplitude value of the sample at the pointer position. When you send a trigger signal to one of these Sampler Modules, the pointer will move linearly through the sample (with constant speed) and hence generate a time-varying Audio signal at the output port.

The speed of the pointer movement determines the speed of the sample playback. At the same time it also determines the pitch that is heard: the more slowly the signal (which has been recorded in the sample) is sampled, the longer the periods of the oscillations occurring at the output are — the pitch therefore decreases. If the pointer stops moving, the output amplitude stops changing and no audible signal is produced. For the “conventional” Sampler Modules, this relationship between playback speed and pitch is fixed. If you send a higher pitch value to the “P” input port of the Module and trigger a playback, then the playback speed is increased. The spectral components of the sample are transposed uniformly, that is, all components are transposed by the same amount. This means that by transposing a sample too much, the perceived timbre of the sample is strongly altered. For example, an adult human voice that is transposed upwards an octave will sound like a chipmunk.

10.2.2 Grain Resynthesis Modules

The Grain Resynthesis Modules category includes the Beat Loop, Resynth, Pitch Former, and Grain Cloud Modules. In the same way as a “conventional” Sampler Module, a Grain Resynthesis Module uses a pointer at the current sample position. However, the signal at the output is not simply the sample amplitude occurring at the pointer position. What actually happens is that the output signal is generated by a synthesizer inside the Module. This synthesizer resynthesizes the signal at the pointer position. The pitch of the signal generated by this synthesizer is independent of the pointer's speed. Even if the pointer is not moving at all, the synthesizer continues to produce a sound. While the pitch of the output signal is determined, as is usual, over the “P” input, the “Sp” input determines the pointer speed for the Resynth and Pitch Former Modules. For the Grain Cloud Module one specifies the pointer position using the “Pos” input port.

As can be seen in the figure above, the Beat Loop, Resynth and Grain Cloud Modules, along with the “conventional” Sampler Modules feature uniform spectral transposition. In these Modules, the “chipmunk effect” (also mentioned at the end of the previous subsec-

tion) occurs when speech recordings are detuned. This effect can be traced back to the transposition of the formants whose position for a “real” human speaker would be largely independent of the fundamental pitch. However, Sampler Modules with uniform spectral transposition also shift these formants during transposition and hence cause changes in the sound like the “chipmunk” effect. On the other hand, the Pitch Former Module implements algorithms that compensate the unnatural change in timbre, when the sample is transposed up or down. So within certain limits, the Pitch Former Module decouples the pitch and the formant position from one another. The formant position can be shifted independent of pitch via the “FS” input port. Since the human ear uses all the spectral components to identify the fundamental pitch, you can manipulate this parameter to create interesting substitutions of fundamental pitch and timbre, especially at very low pitches. Similar sound effects are often created by synthesizer experts using the technique called “oscillator synchronization”.

10.2.3 The Sample Lookup Module and Wavetables

The Sample Lookup Module

The Sample Lookup Module (screenshot below) is the outsider in the context of the previously discussed categories. It can function as a sort of oscillator by making the loaded samples available as function value look-up tables. When a position within the sample is specified in milliseconds at the “Pos” input port, the value of the sample at that point is sent to the output ports.



Fig. 10.2 The Sample Lookup Module.

Wavetables

Note that you can use the Sampler Loop and Pitch Former Modules for wavetable synthesis. Wavetable synthesis uses a sample file, called wavetable, which consists of a succession of waveforms that have different spectra. The main application is to “morph” through these different waveforms to create sounds that have a varying spectrum over a period of

time. Most often the different waveforms should start and end with a zero crossing to avoid clicks when changing between waveforms. To use the Sampler Loop Module for wavetable synthesis you need to engage the Oscillator Mode checkbox in the Module's Function page (see the next section for more information on the Function page of Sampler Modules).

10.3 Setting the Properties of Sampler Modules

The Properties of the Sampler Modules have several settings which are specific to the Sampler Modules. This section will discuss these settings and how they affect the behavior of your Modules.

10.3.1 Working with the Function page

The Function page of a Sampler Module holds different elements depending on the Module.

Loading Samples or Sample Maps

All Sampler Modules except the Sample Lookup Module use the Sample Map to manage their samples. To open the Sample Map Editor for a particular Sampler Module, go to the Module's Function page and click on the waveform icon to the right of the Map display field, shown below. If no Sample Map has been loaded or defined yet, the Map display field will have the entry “empty”. Since the Sample Lookup Module uses only one sample, it has a folder icon instead of the waveform icon. Click this folder icon if you want to load a sample for the Sample Lookup Module to use.



Fig. 10.3 A typical Function page of a Sampler Module.

Storing the Sample Map with the Module

As discussed in subsection [↑10.1.4, Storing Samples with Modules](#), to store the Sample Map with the Module and hence the Ensemble, engage the Store Map with Module checkbox.

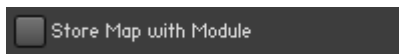


Fig. 10.4 To store the Sample Map with the Module and hence the Ensemble, engage the Store Map with Module checkbox.



A common mistake is to save a sampler Instrument without having engaged the Store Map with Module checkbox in the Function page of the Sampler Module(s), and the to share this Instrument with other REAKTOR users (for example, in the REAKTOR User Library). When another user opens the Instrument, the sample files will be missing. Bottom line: remember to engage the Store map with Module checkbox in all Sampler Modules before saving (and sharing) a sampler Instrument.

Changing the Quality of Sample Playback

For the Sampler Modules in REAKTOR, you have three choices for the quality of sample playback: Poor, Good, and Excellent. Obviously, lower quality playback takes up less CPU resources and higher quality takes up more. To change the sample playback quality, choose the desired quality from the Quality drop-down menu in the corresponding Sampler Module's Function page.



Fig. 10.5 To change the quality of the Module's sample playback, choose the appropriate quality setting from the Quality drop-down menu.

Using a Sampler Module as an Oscillator

When you use a “conventional” Sampler Module as an oscillator, you need to engage the *Oscillator Mode* checkbox. This causes the playback rate to be adjusted to suit the length of the current sample in order to produce the correct pitch when operating as a waveform oscillator. This is also relevant for the case that you want to use the Sampler Loop Module for wavetable synthesis. Note that for wavetable synthesis using the Pitch Former Module you don't need to engage the *Oscillator Mode* checkbox. In fact, the Function page of the Pitch Former Module doesn't even have one!



Fig. 10.6 When using a “conventional” Sampler Module as an oscillator, the Oscillator Mode checkbox needs to be engaged.

Changing a Sampler Module's Output to Mono

Some Sampler Modules like the Sampler Loop, Resynth, Grain Cloud, Beat Loop, and Pitch Former Modules give you the option to have a stereo or mono output signal. For a stereo output signal the “L” and “R” output ports are used for the left and right channels of the Audio signal, respectively. If you want to save CPU and just use a mono signal, engage the *No Stereo* checkbox in the Function page of that Sampler Module (see screenshot below) signal is forwarded only to the “L” output port and while the “R” output port receives no signal.

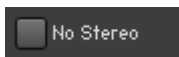


Fig. 10.7 To only receive an Audio signal at the “L” output port of a Sampler Module, engage the No Stereo checkbox.

Specifying Grain Settings for the Grain Cloud Module

The Grain Cloud Module has some unique settings in its Function page. First, you can specify the shape of the envelope of the grain using the *Attack* and *Decay* edit fields. Second, you can control the maximum number of grains that can be played simultaneously. This becomes relevant when the length of the grains is considerably greater than the distance between the starting points of the individual grains (these values can be specified using the corresponding input ports). Since more grains use up more CPU resources, you can save your CPU resources by reducing the number of overlapping grains with the *Overlap* edit field.

Another thing to consider when you have several grains playing simultaneously is the amplitude of the output signal. More overlapping grains cause a greater amplitude. The Grain Cloud Module has an amplitude compensation algorithm built into it. You can activate it by engaging the *Gain Correction* checkbox.

The Pitch Jitter effect causes the pitch to change in a random manner from one grain to the next. The change in pitch fluctuates in the range $[-PJ, PJ]$ where *PJ* is the Pitch Jitter value at the “*PJ*” input port. Sometimes you want this pitch fluctuation to be more orderly. By engaging the *PJ Static* checkbox you cause the change in pitch to alternate between the explicit values “*-PJ*” and “*PJ*”, instead of the whole range between them.

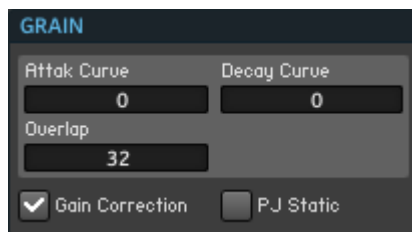


Fig. 10.8 The Grain Cloud Module has specific settings in its Function page. Using that area you can change the envelope and overlap of the grain window, activate gain correction, and make the Pitch Jitter effect non-random.

10.3.2 Working with the View Page

All the Sampler Modules have a Panel representation. Along with the Panel representation come settings in the View page of the Module's Properties. This subsection will discuss how to make the different parts of the Panel representation visible or invisible and change the size of the sample waveform display.



Fig. 10.9 The different parts of a Sampler Module's Panel representation.

As seen above, the Sampler Module's Panel representation consists of three main parts, the Label, the Value, and the Picture. Additionally, you can make the scrollbars appear or disappear and change the size of the sample waveform display. To see how to do all of these tasks, look at the following instructions:

1. First, make sure you have a Module from the *Built-In Module > Sampler* menu placed into the Structure.
2. Next, double-click on the Module to open the Module Properties.
3. Click on the *View* button.

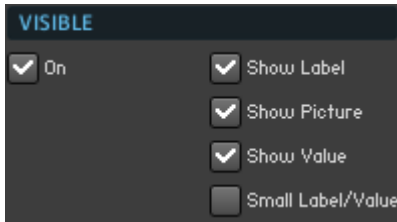


4. First, let's look at how to change the visibility of the different parts of the Panel representation. The whole Panel representation can be made invisible by engaging the *On* checkbox.

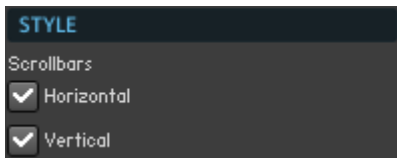


5. Disengage the checkbox and see the sampler disappear from the Panel. Engage the checkbox again to make it appear.

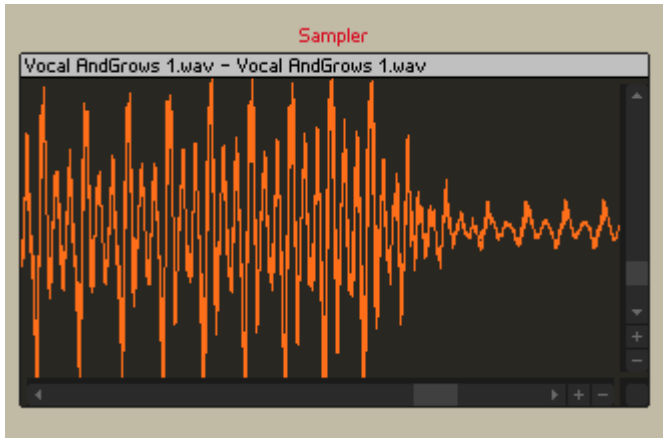
- Now let's turn our attention to hiding and showing the parts of the control separately. This can be done with the help of the checkboxes shown below. Disengage the **Show Label** checkbox to see the control Label disappear.



- Now, disengage the **Show Picture** checkbox to see the sample waveform display disappear.
- Finally, you can even make the **Value** field that shows the names of the sample and Sample Map disappear by disengaging the **Show Value** checkbox. Note that now clicking on the **On** checkbox to the right of the three checkboxes that you used last has no visible effect.
- Engage the **Show Label**, **Show Picture**, and **Show Value** checkboxes to make them appear.
- If the **Value** field at the top of the sample waveform display seem too large to you, you can reduce its size by engaging the **Small Label/Value** checkbox.
- Sometimes you want to be able to zoom in on the sample waveform in the display and focus on different areas of the sample. You can do this for both the horizontal and vertical axes of the sample waveform. To enable the buttons for zooming and the scrollbars for scrolling the sample waveform display, engage the **Horizontal Scrollbar** and **Vertical Scrollbar** checkboxes:



- Here the waveform has been zoomed in on:



13. The last thing that you can change is the size of the sample waveform display. This is done by typing in the horizontal and vertical size in pixels into the **Width** and **Height** edit fields, respectively. These edit fields are shown below.

SIZE

Width
 Pixels

Height
 Pixels

11 Working with Table Modules

In this chapter you will learn the basic information for operating the Audio and Event Table Modules. First, you will get a brief introduction to these Modules, followed by a tutorial that will teach you how to set up an Audio Table to record an Audio signal. In this tutorial you will get to know the operating principles of the Table Modules. After that, 2D graphs and the use of Event Tables for automation are discussed. Finally, the rest of the options on the View page of the Tables Modules will be addressed.

11.1 What Is a Table Module?

The Event and Audio Table Modules allow for very flexible handling of Events and Audio data. They can be used to design oscillators, LFOs, or waveshapers by drawing your own waveforms with the mouse. Additionally, you can crossfade between wavetables and create envelopes with curve shapes drawn by hand or with an arbitrary number of breakpoints. In many cases, the Event Table comes handy as a sequencer for outputting gate and pitch values or for controlling any parameter in a REAKTOR Ensemble. Another useful application for the Event Table is its use as buffer. From a more technical perspective, Table Modules let you associate them with 2-dimensional arrays and give you access to read and write operation on the values of these arrays. The following figures show the Modules in the Structure and a Panel representation, respectively. You will find the Audio Table Module in the *Built-In Module > Audio Modifiers* submenu and the Event Table Module in the *Built-In Module > Event Processing* submenu.

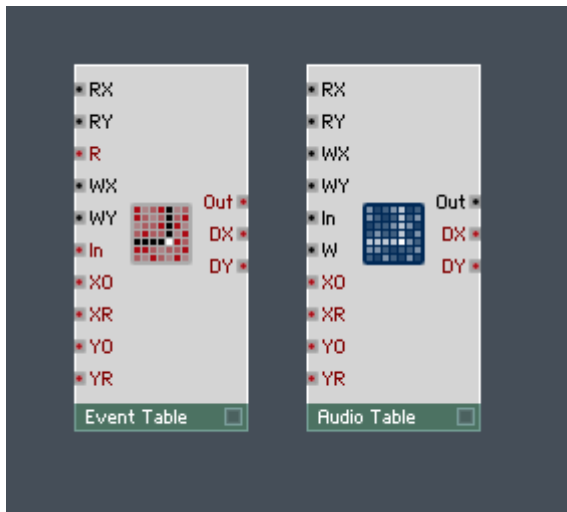


Fig. 11.1 The Event Table and Audio Table Modules differ in the signal type that they process. Also, the Event Table Module sends an Event from its “Out” output port only in response to an Event at its “R” input port.

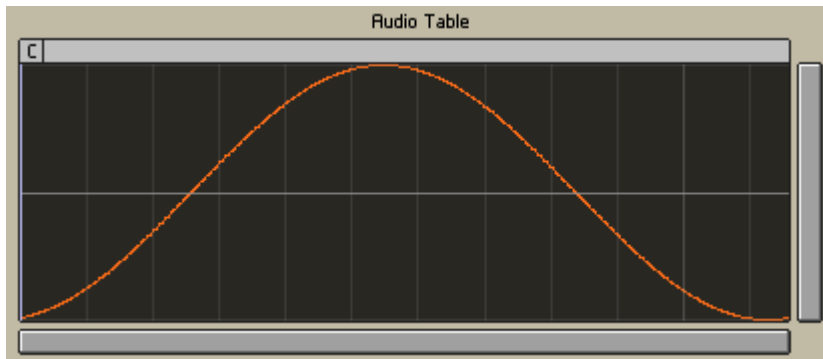


Fig. 11.2 A Panel representation of an Audio Table Module with a sine wave as the table data.

Differences between the Event and Audio Table Modules

One difference between the Event and Audio Table Modules is that the Event Table stores and sends only Events whereas the Audio Table stores either signal type but sends only Audio signals from its “Out” output port. Note, however, that the read and write locations for the Event Table as well as Audio Table can be controlled by Audio signals at the “RX”

and “RY”, and “WX” and “WY” input ports, respectively. Another difference lies in the way read operations are triggered for the two Modules. Whereas the read operation for the Event Table needs to be explicitly triggered by an event at the “R” input port, the Audio Table automatically sends the value at the table position specified by the values at the “RX” and “RY” input ports to its output port. As for the write operation, the Event Table writes the value of the Event at the “In” input port when that Event arrives at its “In” input port. In the case of the Audio Table, the value at the “In” input port is written into the specified table cell when the value at the “W” input port is greater than zero. In both cases, the table cell to which the value is written is specified by the “WX” and “WY” input ports.

11.2 The Function page of the Audio Table: Recording and Playing a Signal

In this tutorial you will learn how to build a simple structure that records an Audio signal from a Sine oscillator into an Audio Table. After that you will see how to play back the recorded data as an audible signal. Along the way you will get familiar with the most important input and output ports and the whole Function page of the Audio Table Module. The tutorial is distributed among several subsections so that when you to return to this section later wanting to refresh your memory on a specific task, it is easier for you to find the content.

11.2.1 What Is the Goal?

The goal of the tutorial is to learn the basic operating principles of the Table Modules. A schematic of the Structure that you will build is shown below. The signal that is to be recorded stems from the output of the Sine Module and is routed to the “In” input port of the Audio Table Module. The pointer of the table cell to which the current signal value at the “In” input port will be written is specified by the values at the Table Module’s “WX” and “WY” input ports. We will only use the “X” dimension of the table to record our values so we only need to use the “WX” input port. The pointer should be a linear ramp that goes from the beginning of the table to the end of the table. We generate this linear ramp with the Ramp Module and the amplitude of the ramp has to match the length of the table, which we fetch from the “DX” output port of the Audio Table Module. In order for the in-

coming signal to be written to the table, a value larger than “0” is needed at the “W” input port. We want to continuously write to the Table Module; therefore a Constant Module with the value “1” is connected to the “W” input port. There are a couple more specifics to which we will get to in more detail later, but all in all this explanation should give you a pretty good idea where we are headed in this tutorial.

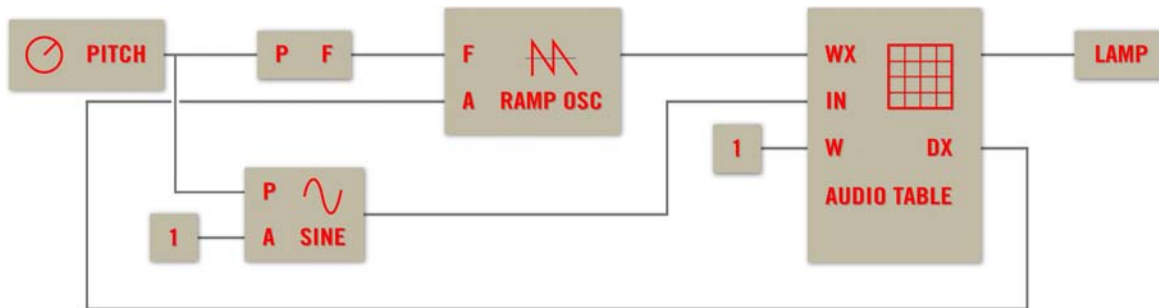


Fig. 11.3 A schematic of the Structure that uses an Audio Table Module to record a signal from a Sine Module.

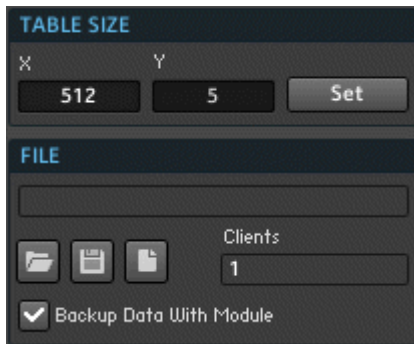
11.2.2 Creating a Table, Table Files, and Linking Table Modules

To start off the tutorial, follow these instructions:

1. Open a new Ensemble.
2. Go to Structure View and place an Audio Table Module into the Structure. You will find in the *Built-In Module > Audio Modifiers* submenu.
3. Right now your Audio Table Module is just an empty shell. Let's tell the Table Module the dimensions of the table that it should hold. Double-click the Module to see the Module's Properties.
4. Click on the *Function* button.

FUNCTION

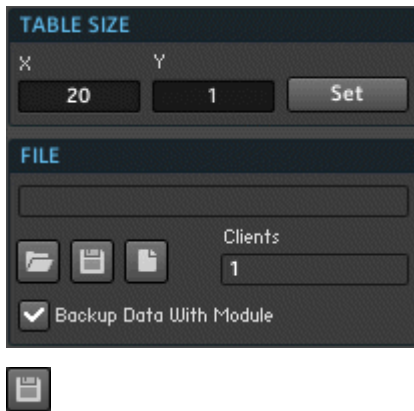
5. Now look at the Table Size area shown below. The Table File display says “<empty>” to show that the Table Module is not associated with any table. Additionally you see that the X and Y displays in the Table Size area show that no dimensions for the table have been specified. Click on the *Set* button to the right of these displays.



6. The Set Table Size dialog shown below should have popped up. As discussed in subsection [↑11.2.1, What Is the Goal?](#), we will create a table that only has an “X” dimension. Therefore, leave the Y edit field set to “1”. For the X edit field, let's choose “20”, a good size to demonstrate some of the aspects of the Table Modules. Click on the X edit field and enter the value “20” and click on the Apply button. Note: If you reduce the number of cells, that is, the dimension of the table, any data contained in the removed cells will be deleted!



7. Looking at the X Size and Y Size displays you see that the values you just entered have been taken as the size of the table. Furthermore, the Clients display shows that there is one Table Module using the table that you just created. If you save a table as a file and load that file into another Table Module, the Clients display should change to “2” on both Modules. This means that the table is shared by the two Modules. Any changes that are made to the table by one Module will be recognized by the other Module. Now let's see how to save a table file. Click on the [Save Table](#) button (disk icon).



Backing Up Table Data with the Table Module

Besides saving your table data in a table file there is a more convenient method to holding on to this information. By engaging the **Backup Data With Module** checkbox, you can cause the table data to be saved with the Table Module when you save your Ensemble, Instrument, or Macro that holds that Table Module. In that case, you don't have to have the table data saved in a file in order to be able to access it when you reload your Ensemble. If you have specified a file for the table data, then upon reloading the Ensemble the Table Module will remember this association but will use the data it has saved rather than try to load it from the table file.

However, when you have the **Backup Data With Module** checkbox disengaged and no table file saved, you will lose your table data when reloading the Ensemble. What will remain, though, is the configuration information for the table like table size, value ranges, and so on. If you have your table data saved in a file and the **Backup Data With Module** checkbox is disengaged, then REAKTOR will attempt to find the table file upon loading to retrieve the table data. If the table file has been modified in between the instances of this Ensemble, then your Ensemble will use this new information as the table data. In the case that the table file has been moved or deleted between sessions, you lose the table data but not the configuration information of the Table Module.

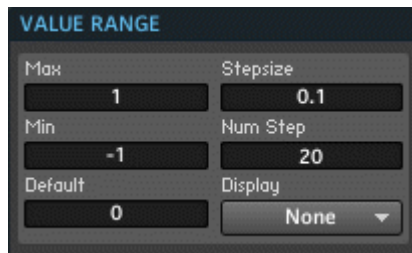


Regardless of the setting of the **Backup Data With Module** checkbox, Table Modules that use the same table file will stay linked. This will be the case even if the actual file has been renamed, moved, changed, or deleted.

11.2.3 Setting the Range, Resolution, and Display Units of Table Values

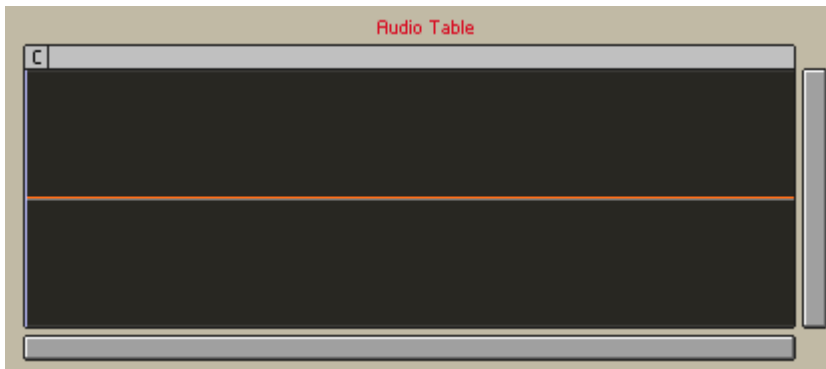
The Table Module holds a table which is a 2-dimensional array of values. You can specify the range and resolution for these values just like in the case for controls like knobs and faders. The resolution is only relevant for the Panel representation of the Module which you use to draw values into the Table. However, the range affects how the incoming values at the “In” input port are treated. Namely, values outside of the range are clipped. To specify the range and resolution of the values in the Panel representation in the context of the tutorial from the previous subsection, follow these instructions:

1. Make sure you have followed the instructions from the previous subsection and now have the same Structure open to continue with this tutorial. Open the Function page of the Audio Table Properties.
2. First, let's change the minimum and maximum values of the table. Later in the tutorial we will use a sine wave with unit amplitude as the input signal. We want to match our range to that, so we should set it to $[-1, 1]$. We can do this using the **Min** and **Max** edit fields as shown below. Change the minimum value of the table to “-1” and the maximum value to “1”. You just need to double-click the corresponding edit field and type in the new value.

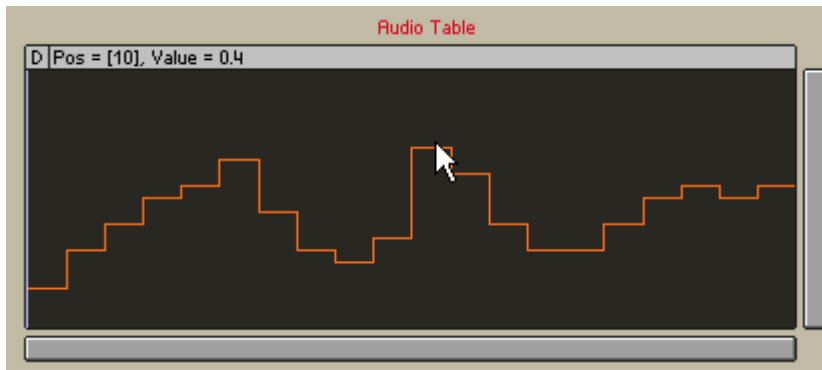


3. Now that your table has the desired range we will turn our attention to its resolution. Although the precision of the display of the incoming values is independent of the Stepsize setting, you can set the resolution at which you can draw values into the table. Let's try this. Change the **Stepsize** edit field as shown above to the value “0.2”. Notice that the **Num Steps** edit field automatically changed its value—the range and the Stepsize and Num Steps edit fields are interdependent.

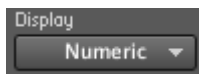
4. We said that the resolution refers to the Panel representation. So we better try to test our changes by drawing on the table in the Instrument Panel. Look at the Panel representation of your Audio Table Module, shown below. Notice how all values are half of the range of the table, “0”. This correctly reflects the value “0” in the *Default Value* edit field. Before you can draw anything in the table, you need to change the Table Mode. Right now you are in Table Control Mode, as indicated by the “C” in the top left corner of the table. Click that icon twice, skipping Table Select Mode, to reach the Table Draw Mode. The icon in the top left of the table should now be a “D”. Please refer to subsection [↑11.3.5, Table Modes and Basic Table Editing Operations](#) for more information on the different Table Modes. Now use your mouse to draw new values into the table!



5. After drawing new values into the table, the Panel representation of your Audio Table Module should look similar to what is shown the screenshot below. With the Num Steps value set to “10”, the signal is quantized quite heavily and appears stepped. Notice that when you have the mouse button pressed information about the “X” coordinate of the table cell you are writing to and the value that you are writing appear at the top of the table. To change the units of the value, go to the Function page of the Audio Table Module.



6. You can change the units in which the value is displayed in the Value Bar of the table using the **Display Units** drop-down menu, shown in the following screenshot. Numeric units just display the units that follow directly from the settings of the range and resolution you set above. Choosing the *MIDI Note* menu entry causes the value that would be displayed in Numeric units to be converted to the closest note in the MIDI scale (see section [↑9.4, Different Value Ranges and Scales in REAKTOR](#) for more information on scales). If you choose the *0 - 100%* menu entry, the display shows the percentage of the value from its total range. In our case, the range is [-1, 1] and the value “0” lies right in the middle, hence it would be displayed as 50%.



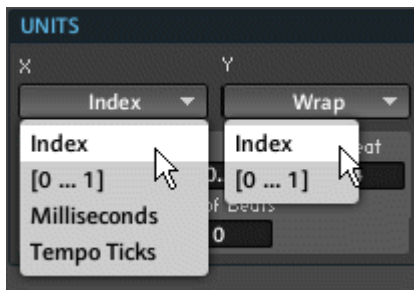
7. Similarly as for the value of the table data, you can also choose the units for the table cell position. However, whereas for the table values these units only affected the display, for the table cell positions these units also determine how the values at the “WX”, “WY”, “RX”, and “RY” input ports are interpreted. See the tip below for more information on writing to tables. Changing these units is done in the Units area, using the X and Y drop-down menus for the “X” and “Y” dimensions of the table, respectively. The different units possible are shown in Units dialog. The *Index* menu entry is the default setting. The index refers to the number of the table cell. This can be seen in Audio Table (see above) where the Value Bar shows the index “10” as the tenth table cell is being edited. Click on the second entry, *[0 ... 1]*. This causes the value

of the index divided by the size of the table's corresponding dimension to be displayed. In our case the size of the “X” dimension is “20” so an “X” index of “10” would be displayed as $20 \div 10 = 0.5$.

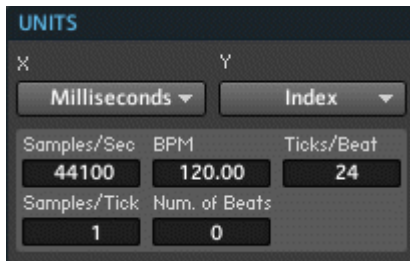


When writing to the table, the value at the “In” input port is written to the table cell specified by the values at the “WX” and “WY” input ports. The units with which the table cell is specified are determined by the X and Y drop-down menus in the Units area of the Function page. If the specified position lies between two table cells, the position is rounded down. For example, if you are using the “Index” units and have a value “0.6” at the “WX” input port, then the write operation happens to the table cell at “X” position “0”.

8. Now click on the *Milliseconds* entry. This causes the index to be taken as the sample number from the start of a sample file. This number is then converted to milliseconds using the Sample Rate as specified in the Samples/Sec edit field. For example, 10 samples at a Sample Rate of 44'100 Hertz results in $10 \div 44'100 = 0.0002$ seconds = 0.2 ms.



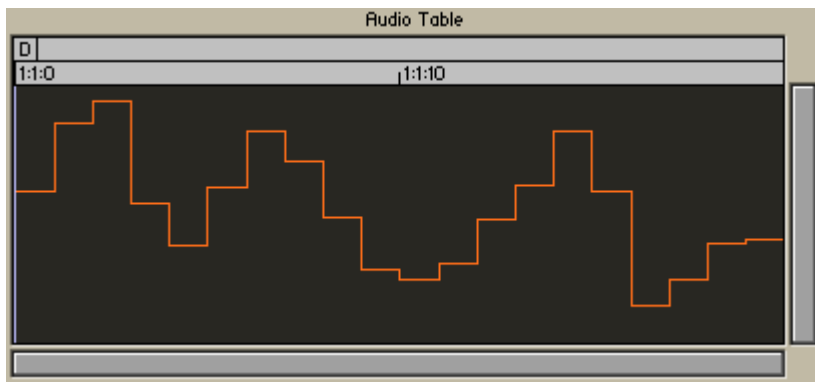
9. To see the last “X” unit, click on the *Tempo Ticks* menu entry. This option is especially useful in the Audio Table Module when a rhythmic audio sample, for which you know the BPM (beats per minute) tempo, has been loaded as the table data. The position is taken as number of samples, from which the position in ticks is calculated using the *Samples/Tick* edit field (see below). Using the value specified in the *Ticks/Beat* edit field (see below), the position in ticks is then converted to the position in beats. You can either specify the BPM of the sample file or alternatively the total number of beats that are contained in it. When a sample file is loaded, the Sample Rate is automatically read from the file and set as the value in the *Samples/Sec* edit field.



⚠ When saving a sample as an audio file, the value in the *Samples/Sec* edit field will be written to the file as the Sample Rate. If you want to open the file in an audio editor, make sure that you have numbers that make sense. A Sample Rate of “1” could very well cause problems with your audio editor program.

⚠ When saving a sample as an audio file, the value in the *Samples/Sec* edit field will be written to the file as the Sample Rate. If you want to open the file in an audio editor, make sure that you have numbers that make sense. A Sample Rate of “1” could very well cause problems with your audio editor program.

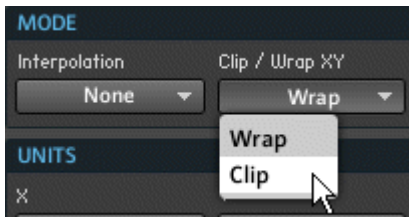
You can also display a ruler that shows the different values (in the chosen units) corresponding to the different positions in the table. To see this ruler, right-click the Panel representation of the Table Module and select the *View > Show Horizontal Position Line*. Below you see a table having the ruler activated while using the “Tempo Ticks” unit.



11.2.4 Clip, Wrap, and Interpolation Modes

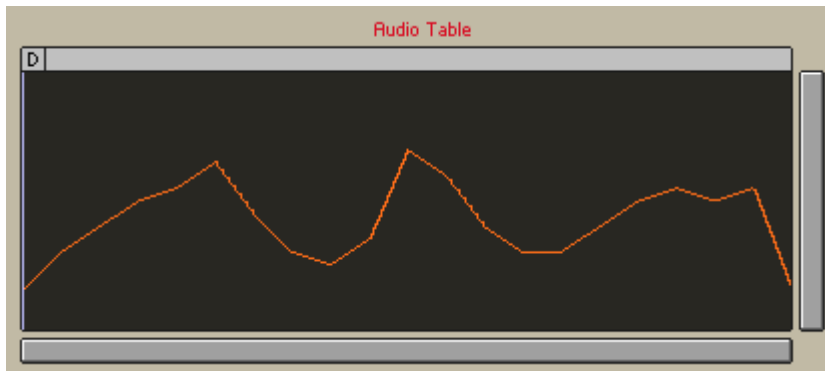
As mentioned in the previous subsection, when writing to the table, the value at the “In” input port is written to the table cell specified by the values at the “WX” and “WY” input ports. Analogously, reading happens from the table cell specified by the values at the “RX” and “RY” input ports. What happens if your position values go beyond the size of the corresponding dimension? In such a case, the behavior is determined by the *Clip / Wrap XY* drop-down menu. When writing to the table, the write position values at the corresponding input ports are rounded down to the values corresponding to the nearest table cell. For read operations, though, you have the option for interpolation between the values held in table cells. This applies when your “RX” and “RY” values lie between the discrete table cell positions. To see how these operation modes of the Table Module affect its behavior, follow the instructions below:

1. Make sure you have followed the instructions from the previous two subsections and now have the same Structure open to continue with this tutorial.
2. First, let's look at how clipping and wrapping works. Make sure that you have the Function page of the Audio Table open and that you can see the Panel representation of the Module.
3. Select the *Clip* menu entry from the *Clip / Wrap XY* drop-down menu (see below).



4. To make sure that you are using the correct graph display for this part of the tutorial, right-click on the Panel representation of the Audio Table Module and verify that the *Graph > Line* menu entry has been selected.

5. Now draw data into the table, going beyond the left or right border of the table. You see that once your mouse has gone beyond the border, the leftmost (or rightmost, depending on the border) table cell is the one that is being changed. This behavior is called clipping: all positions beyond the actual size of the table are being replaced by the position that is at the bound that has been crossed.
6. Next, let's see how wrapping works. Select the *Wrap* menu entry from the *Clip / Wrap XY* drop-down menu (see “Mode” dialog above).
7. Again, draw data into the table, going beyond the left or right borders. This time, however, you should see that if you cross a border, the draw position “wraps around” and starts again at the opposite end of the table.
8. You now have seen how the clipping and wrapping works in the “X” dimension. For the “Y” dimension of the table it works analogously. Next up is the interpolation setting. Select the *X* menu entry from the *Interpolation* drop-down menu (see “Mode” dialog above).
9. The appearance of your stepped graph should have changed to is linearly interpolated between the discrete values (see following figure). This interpolation does not only affect the display of the table values but also applies to the output signal of the Table Module. For “RX” and “RY” values that lie between two table cell positions, the value of the linear interpolation between the corresponding values is sent to the output. In essence, the graph on the Instrument Panel displays the linear interpolation and hence the values that will be sent to the output. The *Y* and *XY* entries of the *Interpolation* menu effect linear interpolation in the “Y” dimension and “XY” plane analogous to what you saw in this tutorial. Notice that since you have selected the *Wrap* menu entry, the table also interpolates between the last and first table values. Select the *Clip* menu entry in the *Clip / Wrap XY* drop-down menu.
10. Notice how now there is no interpolation between the last and first table values. With this we finish this part of the tutorial. You can find the next instruction in the following subsection or save the Ensemble and continue there at some later point.



! Note that for the Multi Line and 2D Color graph styles, as described in subsection [↑11.3.1, Graph Style](#), the visual feedback regarding the interpolation is not supported. However, when reading values from the table, the result is interpolated as in the case for the other graph displays.

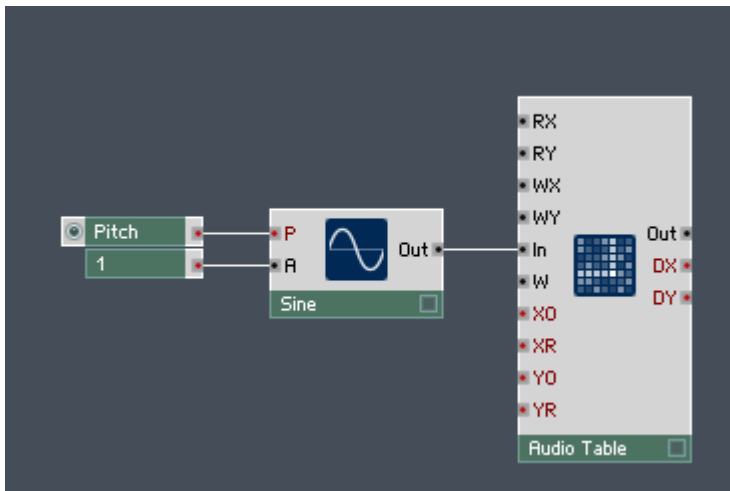
! Note that for the Multi Line and 2D Color graph styles, as described in subsection [↑11.3.1, Graph Style](#), the visual feedback regarding the interpolation is not supported. However, when reading values from the table, the result is interpolated as in the case for the other graph displays.

11.2.5 Setting Up the Generator Signal and Activating the Audio Table

Before we can continue exploring the experimentation with our Audio Table Module, we need to set up an Audio signal that we can then use to demonstrate how the Table Module reacts to signals at its “In” input port. Also, we need to connect the Audio Table to an active Module so that the Table Module will be activated as well. To start, follow these instructions:

1. Make sure you have followed the instructions from the previous subsections and now have the same Structure open to continue with this tutorial.
2. First, go to the Set Table Size dialog and set the X edit field to “44100”. This corresponds to 1 second of Audio signal at a Sample Rate of 44'100 Hz.
3. We will use a sine signal for this tutorial, so insert a Sine Module from *the Built-In Module > Oscillators* submenu into the Structure.

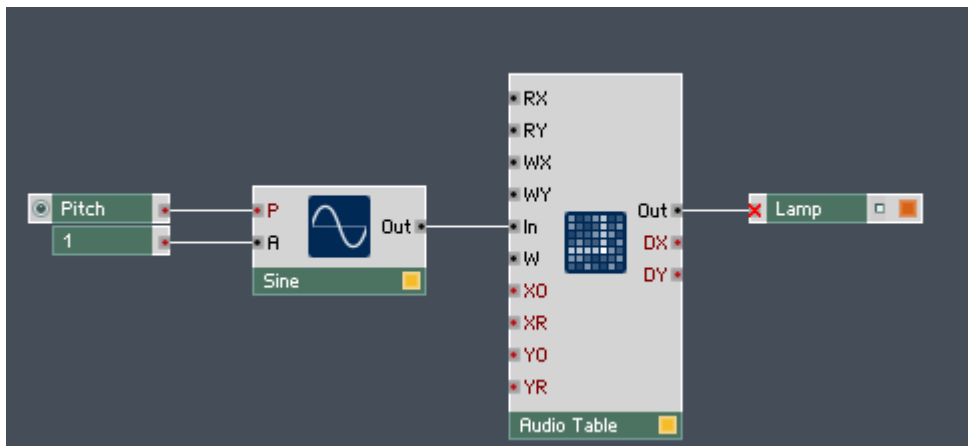
4. Right-click on the “A” input port of the Sine Module and select the *Create Constant* menu entry.
5. Make sure that the Constant has a value “1”. This causes the amplitude of the sine waveform to be “1”, that is, the value of the Audio signal oscillates between “-1” and “1”.
6. Next, right-click on the “P” input port of the Sine Module and choose the *Create Control* menu entry.
7. Now you have a knob with which you can control the pitch of the Sine Module. Connect the output of the Sine Module to the “In” input port of the Audio Table Module. Your Structure should look like the one below.



8. As you can see above, both Modules and the wires between them are inactive. In order to activate this signal flow, we need to connect these Modules to an active Module. One way to do this is to insert a Module with an *Always Active* checkbox in its Function page. An example of such a Module is the Lamp Module. Insert one into the Structure, you will find it in the *Built-In Module > Panel* submenu.
9. Now go to the Function page of the Lamp Module and engage the *Always Active* checkbox (see below). You will see the lamp on the bottom right corner of the Module light up.



10. Now connect the “Out” output port of the Audio Table Module to the input port of the Lamp Module. The lamps of all the Modules should have lit up now, just as shown below. The use of a Panel Element is a common way to activate signal flows that would be otherwise inactive.



11. Last save the Ensemble and continue with the tutorial in the next subsection now or at some later point.

11.2.6 Setting Up the Recording Pointer Using a Ramp Module and the “DX” Output Port

Now that we have the signal that we wish to record connected to the “In” input port of the Audio Table, we will focus on setting up the pointer that tells the Audio Table Module to which table cell the incoming signal value should be written to. To continue with the tutorial from the previous section, follow these instructions:

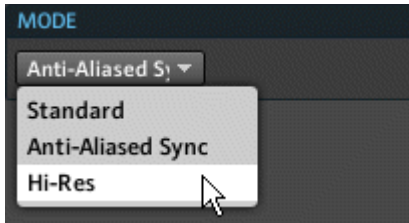
1. Make sure you have the Structure from the previous chapter with the configured Audio Table open.
2. We are recording one Audio signal into a one-dimensional table with 44'100 cells. This means that we need a pointer that linearly scans the pointer from cell 1 to cell 44'100. Since the pointer is meant for the recording signal, it should be connected to

the “WX” input port of the Audio Table Module. The Ramp Module is meant exactly for the purpose of creating an exact linear ramp. Insert one into the Structure, you will find it in the *Built-In Module > Oscillator* submenu.

3. Now double-click the Ramp Module to open its properties.
4. Click on the *Function* button.

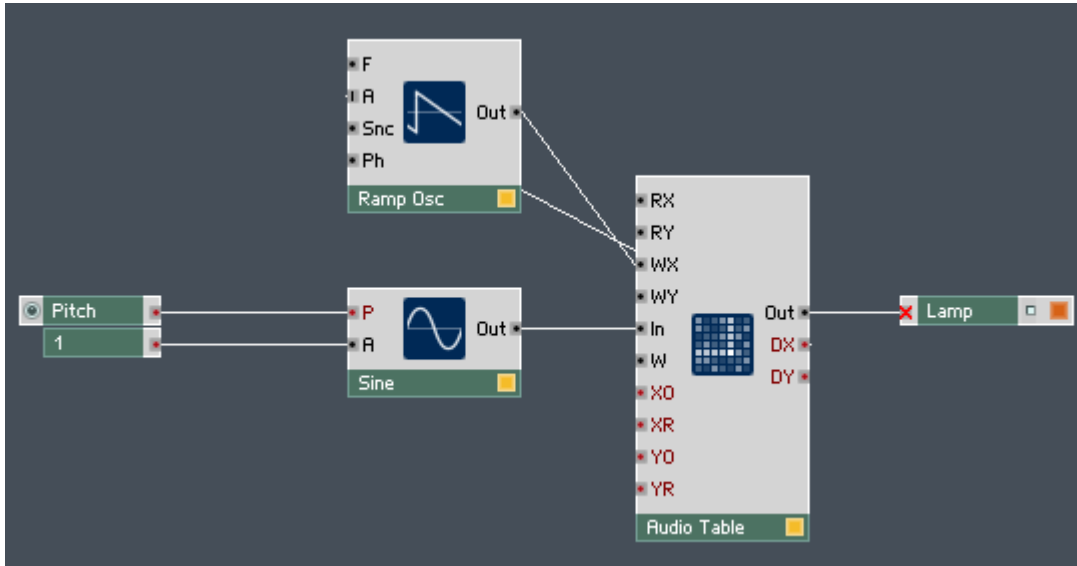
FUNCTION

5. You should now see the *Mode* drop-down menu in the Function page (see below). The main difference between the different types is the way anti-aliasing is treated. In our case, we need a ramp that is as precise as possible, without any anti-aliasing which distorts the waveform. For that reason, choose the *Hi-Res* menu entry. It is especially important for long, slow, ramps that this Mode is chosen. For fast ramps you can choose the *Standard* menu entry and save some CPU. If you wanted to use the Ramp Module as the source of an anti-aliased Audio signal, it is recommended that you choose the *Anti-Aliased Sync* menu entry.



6. We said that we wanted the pointer to run through the whole table, from beginning to end. This means that the amplitude of the ramp, specified at the “A” input port, should be equal to the size of the “X” dimension of the table. We could specify this value, which we know to be 44'100, manually with the use of a constant. However, this is a good moment to demonstrate the usefulness of the “DX” (and “DY”) output ports of the Audio Table Module. The “DX” and “DY” output ports of the Table Modules give us the size of the “X” and “Y” dimensions of the table, respectively. Hence, drag a wire from the “DX” output port to the “A” (amplitude) input port of the Ramp Module.

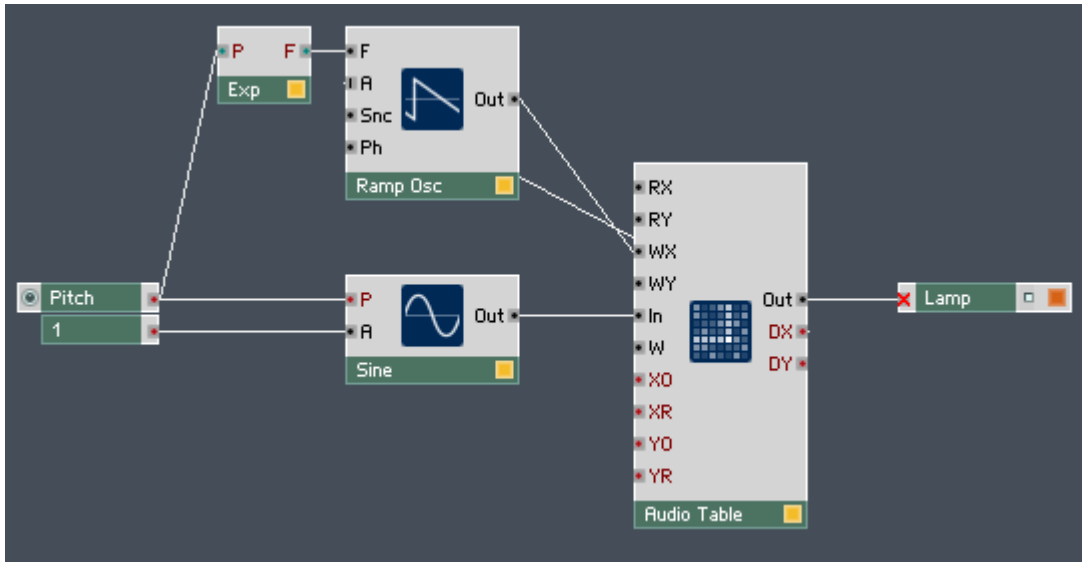
7. We already hinted in step 2 at the fact that we want to use the output signal from the Ramp Module as a pointer to determine the write position of the Audio Table. Therefore, connect the output of the Ramp Module to the “WX” input port. The result should look something like the Structure below.



8. The last thing that we need to do to make the pointer work is to specify the frequency at which the pointer goes from the beginning of the table to the end of the table. If we want to use the resulting table as an oscillator, we would like the table to hold exactly one cycle of the waveform. This means that the frequency of the ramp should match that of the sine wave from the Sine Module. Therefore we should use the information from the Knob Module labeled “Pitch” to specify the frequency of the Ramp Module. The pitch value, however, is specified in the logarithmic pitch scale whereas the “F” (frequency) input port of the Ramp Module requires that we specify the frequency in linear scale. If you are confused about the different scales in REAKTOR and would like to learn more about them, please read section [↑9.4, Different Value Ranges and Scales in REAKTOR](#). Otherwise you can content yourself with the fact that REAKTOR already has Modules to switch between the two scales. Insert an Exp

(P-to-F) Module into the Structure to convert from the logarithmic pitch scale to the frequency scale (If you liked to convert from the linear frequency scale to the logarithmic pitch scale you should use the Log (F-to-P) Module).

9. Now connect the output of the Knob Module labeled “Pitch” to the input port of the Exp (P-to-F) Module and the output of the Exp(P-to-F) Module to the “F” input port of the Ramp Module. Your resulting Structure should look like the one shown below.



10. Last save the Ensemble and continue with the tutorial in the next subsection now or at some later point.



When writing to the table, the value at the “In” input port is written to the table cell specified by the values at the “WX” and “WY” input ports. The units with which the table cell is specified are determined by the X and Y drop-down menus in the Units area of the Function page. If the specified position lies between two table cells, the position is rounded down. For example, if you are using the “Index” units and have a value “0.6” at the “WX” input port, then the write operation happens to the table cell at “X” position “0”.

11.2.7 Writing to the Table and Reading the Result

Now we are ready to write the signal to the Table Module and after that also listen to the result! Just follow these instructions:

1. Make sure you have the Structure shown in the previous chapter with the configured Audio Table open.
2. First, go to the Instrument Panel and tidy up the different controls.
3. Next, make sure that the “Pitch” knob has a value between “0” and “10”.
4. Now you are ready to active the writing of the incoming signal to the table. Go back to Structure View and right-click the “W” input port of the Audio Table Module. Select the *Create Constant* menu entry.
5. Now make sure that the value of the Constant Module is “1”.
6. Look at the Instrument Panel to see that a sine waveform is being written into the table. Now that the sine wave has been recorded to the table, disconnect the Constant Module from the “W” input port.
7. Although the pointer keeps running through the table, no write operation is taking place. Instead, we can use the pointer for a read operation. Connect the output port of the Ramp Module to the “RX” input port. This causes the pointer to specify the table cell whose value is sent to the “Out” output port of the Audio Table Module.
8. Finally, connect the “Out” output port of the Audio Table Module to the Audio Out Modules in your Ensemble. Don't forget to use Audio Voice Combiner Modules to make the output signal Monophonic, if need be. Your final Structure should look like the one shown below.

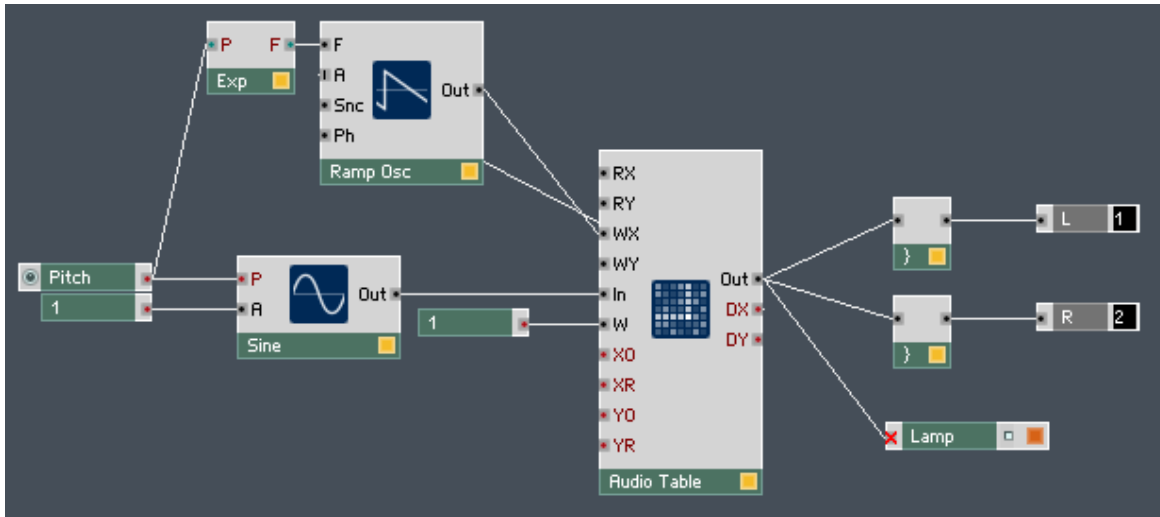


Fig. 11.4 Connecting the output port of the Ramp Module to the “RX” input port of the Audio Table Module turns the Audio Table into an oscillator. Your final Structure should look something like this.

Congratulations, you have reached the end of the Audio Table Tutorial! Now you should know how to configure a table for the Table Module, change the units of the value and cell position, create a pointer for read and write operations, and activate the Module using the Lamp Module. To learn about the various options regarding graph display and appearance of the Table Modules, please refer to section [11.3, Graph Display and Appearance Properties](#).

11.3 Graph Display and Appearance Properties

Part of what makes the Table Modules such powerful tools is their flexibility in displaying data. In this section you will learn all the possibilities related to visualizing table data through the Panel representation of the Table Module. Along the way, all the elements of the Table Module's View page will be covered.

11.3.1 Graph Style

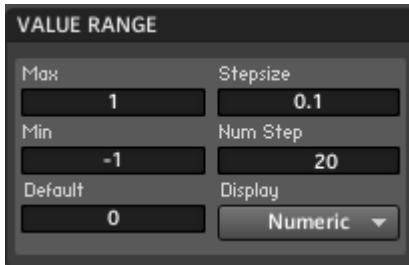
The first place to start is to get acquainted with the different graph styles of the Table Module available to you. Follow these instructions:

Viewing 1-Dimensional Data

1. Open up a new Ensemble and place an Event Table Module into the Structure.
2. Double-click the Event Table Module to enter the Module Properties.
3. Click on the Function button.

FUNCTION

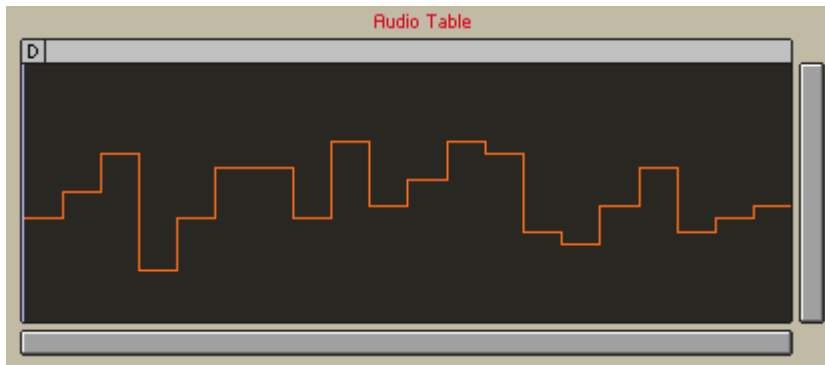
4. Go to the Set Table Size dialog (discussed in subsection [↑11.2.2, Creating a Table, Table Files, and Linking Table Modules](#)) and set the X edit field to “20” and the Y edit field to “5”.
5. Let's change the minimum and maximum values of the table to [-1, 1]. Do this using the Min and Max edit fields as shown in the next figure. Change the minimum value of the table to “-1” and the maximum value to “1”. You just need to double-click the corresponding edit field and type in the new value. Also, set the Num Steps edit field to “20”.



VALUE RANGE

Max	Stepsize
1	0.1
Min	Num Step
-1	20
Default	Display
0	Numeric ▼

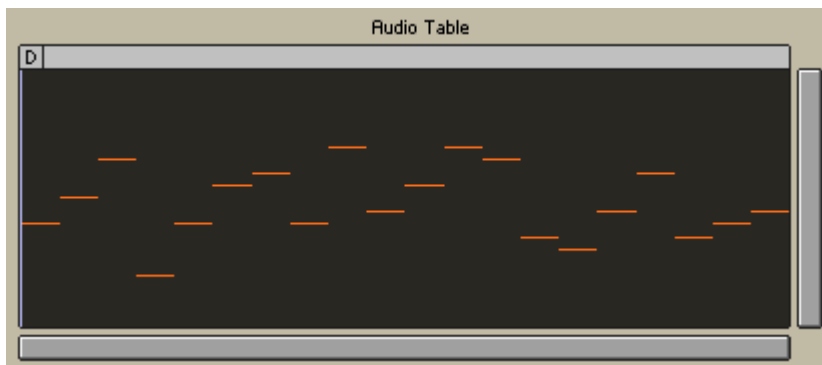
6. Now look at the Panel representation of your Table Module. Right now you are in Table Control Mode, as indicated by the “C” in the top left corner of the table. Click that icon twice, skipping Table Select Mode, to reach the Table Draw Mode. The icon in the top left of the table should now be a “D”.
7. Draw table data into the table. The resulting graph should look similar to the figure below. This graph style is called “Line”.



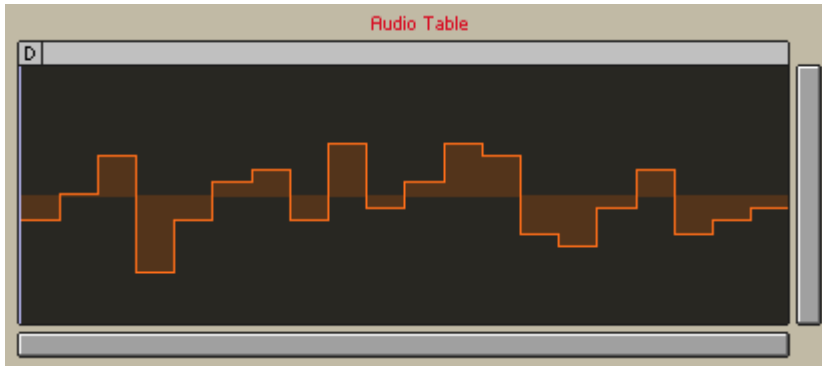
8. Right-click on the Panel representation of the Event Table module and select the *Graph > Pixel* menu entry to change the graph style. Note that you can also change the graph style from the View page of the Table Module, using the Graph drop-down menu shown below.



9. After changing the graph style to “Pixel”, the resulting graph should look similar to the figure below.



10. Now choose the *Graph > Line + Fill* menu entry. The resulting graph should look similar to the one below.



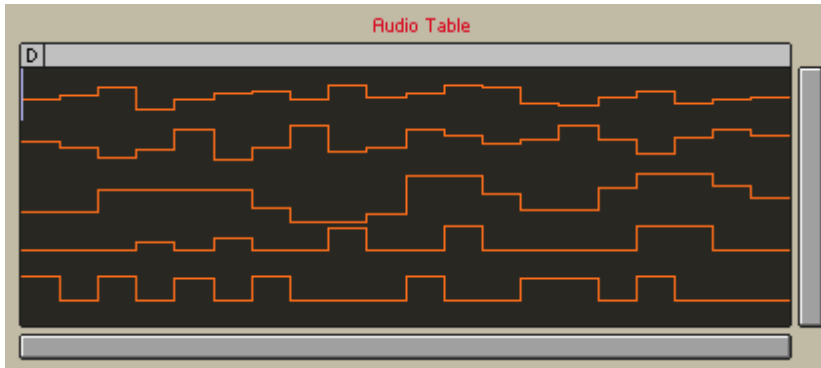
11. Next, choose the *Graph > Fill* menu entry. The resulting graph should look similar to the one below.



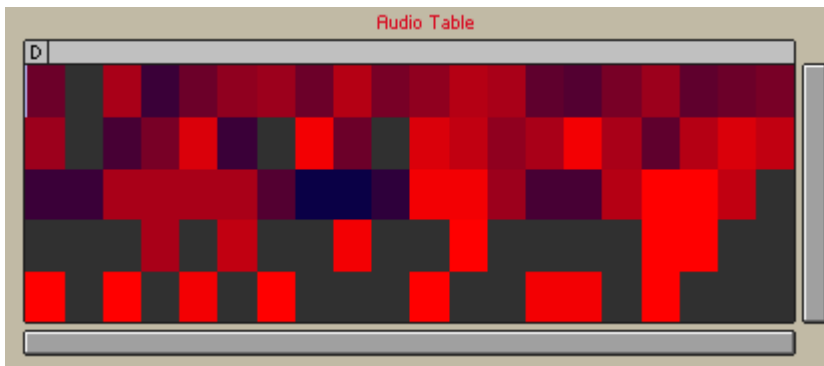
Fig. 11.5 The Fill graph style causes the area between the values and the x-axis to be filled with color. It is the Line + Fill graph style without the lines.

Viewing 2-Dimensional Data

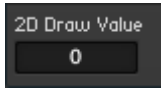
1. The graphs from the previous heading were meant only for viewing the “X” dimension of your table. Choose the *Graph > Multi Line* menu entry to see a graph depicting the table data in 2 dimensions.
2. Draw data into table cells with other “Y” position values as well. The resulting graph should look similar to the figure below.



3. Now choose the *Graph > 2D Color* menu entry. The resulting graph should look similar to the next screenshot.
4. With this graph style, you have to specify which value you want to insert into the corresponding table cell. Comparing the screenshots above and below, you see that dark cells correspond to low values and bright (red) cells correspond to high values. Draw a bit into the table and see that only one value gets written to all the cells that you pass while drawing.



5. To change the value which gets drawn to the cells, right-click on the table and choose the *Process > Select 2D Draw Value ...* menu entry.



6. Try drawing again to the table and see that the cells that you draw get the color that corresponds to the value that you previously chose.
7. Maybe you noticed the *Quantize Draw Values To Step Size* menu entry in the *Process* submenu. When this menu entry has a check mark next to it, it means that the quantization grid that you specified using the *Num Steps* edit field in instruction step 6 of this list is applied to the 2D draw value. Now we will look at how to change between the different “Y” positions while only displaying a 1-dimensional graph. For this, make sure that you have different graphs for the different “Y” positions. Redraw the graphs in the Multi Line graph style, if need be.
8. Now choose the *Graph > Line + Fill* menu entry to again view a 1-dimensional graph.
9. Make sure the Panel representation of the Event Table Module is selected. You can switch between the different “Y” positions of the table by using the Pg Up and Pg Down keys. Another possibility is right-clicking the table and choosing the *Show > Next Y* or *Show > Prev Y* menu entries. Switch between the different pages to see how this feature works.

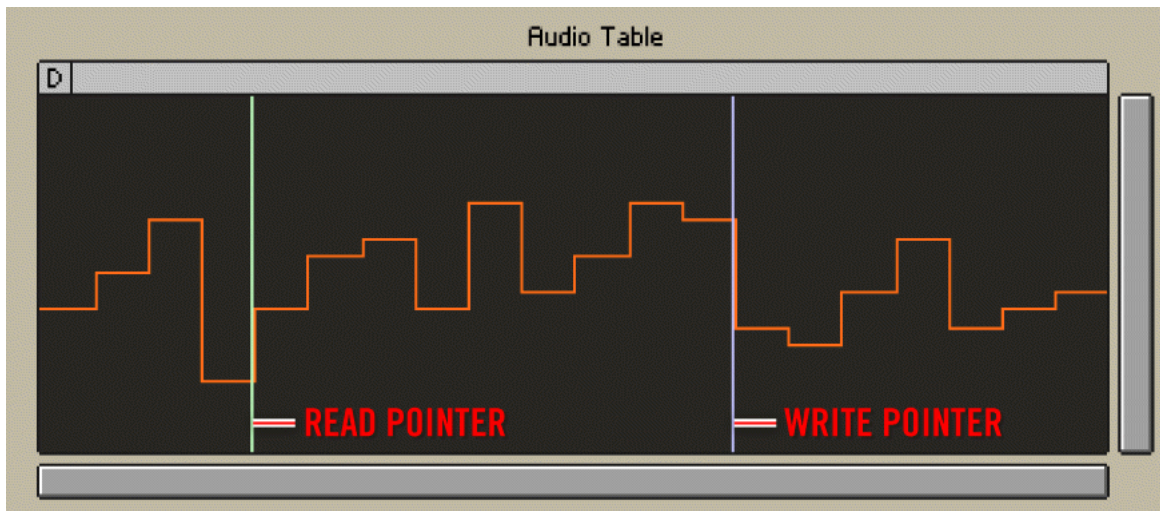


You can also modify the values of single or multiple table cells when working in Table Select Mode. First, select the table cell(s) you wish to change. Then, while holding down the Ctrl key if you are on Windows (Cmd key if you are working in OS X), click and drag the table cell up and down. See the header “Shifting Table Entries and Scaling Values” in the subsection [↑11.3.5, Table Modes and Basic Table Editing Operations](#) for more information on editing table values.

Displaying Read and Write Positions

1. Continuing from step 9 in under the previous header, choose the first “Y” position again. You can see this in the Value Bar, where “Pos = 0” should appear.
2. Next, go to the Structure and right-click on the “RX” input port and choose the *Create Control* menu entry.
3. Right-click on the “WX” input port and again choose the *Create Control* menu entry.
4. Go to the Instrument Panel and move the knobs to an area where they can easily be accessed.

5. Now make sure that in the Clip / Wrap XY drop-down menu the *Wrap* menu entry has been selected.
6. Turn the knobs. The “RX” knob changes the position of the read pointer and the “WX” knob changes the position of the write pointer. Notice how you can see the change in the pointer position on the Panel representation of the Table Module. The ranges of the knobs are greater than the size of the table and the “Wrapping” feature has been activated. Hence, when the output value of the knob surpasses a multiple of the table size, it “wraps around” to the beginning of the table. See the screenshot below for identification of the two pointers. One thing you should have noticed in the previous tutorial was that when turning the “Pitch” knob to a low value you could actually see the pointer moving in the Audio Table Module's Panel representation.



7. To show or hide the read or write pointers, right-click the Panel representation of the Table Module. Then choose the *View > Show Read Position* or *View > Show Write Position* menu entries for the corresponding pointer. This wraps up the information in this subsection!

11.3.2 Scrollbars and Zooming

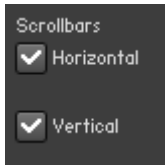
In addition to being able to change the graph style, you can also scroll and zoom the display.

Using the Table Module's Scrollbars

To see how to manipulate the different functions associated with scrolling and zooming, follow these instructions:

1. Set up the environment to demonstrate the scroll function. To do this, create an Event Table Module with X Size “20” and Y Size “5”. The minimum and maximum values should be “-1” and “1”, correspondingly. Also, set the Stepsize parameter to “0.1”. To see a step by step instruction for these tasks, please refer to the first six instruction steps in subsection [↑11.3.1, Graph Style](#).
2. Next make sure that in the Clip / Wrap XY drop-down menu (Function page) the *Wrap* menu entry has been selected.
3. Now look at the Panel representation of your Table Module. Right now you are in Table Control Mode, as indicated by the “C” in the top left corner of the table. Click that icon twice, skipping Table Select Mode, to reach the Table Draw Mode. The icon in the top left of the table should now be a “D”.
4. Use your mouse to draw data into the table.
5. Next, click and drag the horizontal scrollbar in the bottom part of the Table Module's Panel representation left and right. Notice how the display scrolls left and right, but the pattern that you initially drew into the table is repeated. This is the effect of the “Wrapping” feature.
6. To zoom in, click on the edge of the scrollbar and drag in the direction of the bar.
7. To restore the initial view in respect to scrolling and zooming, double-click that scrollbar. This causes the number of displayed cells to be the same as the number indicated by the X Size display in the Function page of the Table Module.
8. Now select the *Clip* menu entry from the Clip / Wrap XY drop-down menu in the Function page.
9. Click and drag the horizontal scrollbar again to see that now the edges of the table are clipped, the table does not repeat itself in the “X” dimension.

10. Also, try clicking and dragging the vertical scrollbar. Feel free to experiment with zooming too.
11. Leave the display setting in a zoom and scroll state that differs from the initial state.
12. Now go to the View page of the Table Module's Properties.
13. Disengage the *Horizontal Scrollbar* and *Vertical Scrollbar* checkboxes to hide the scrollbars from the Table Module's Panel representation.



14. Now that you don't have scrollbars, it is still possible for you to return to the initial, fitted, display settings. To fit the "X" dimension of the display, engage the *X Auto Fit* checkbox (see below). This causes the number of displayed cells to be the same as the number indicated by the *X Size* display in the Function page of the Table Module.



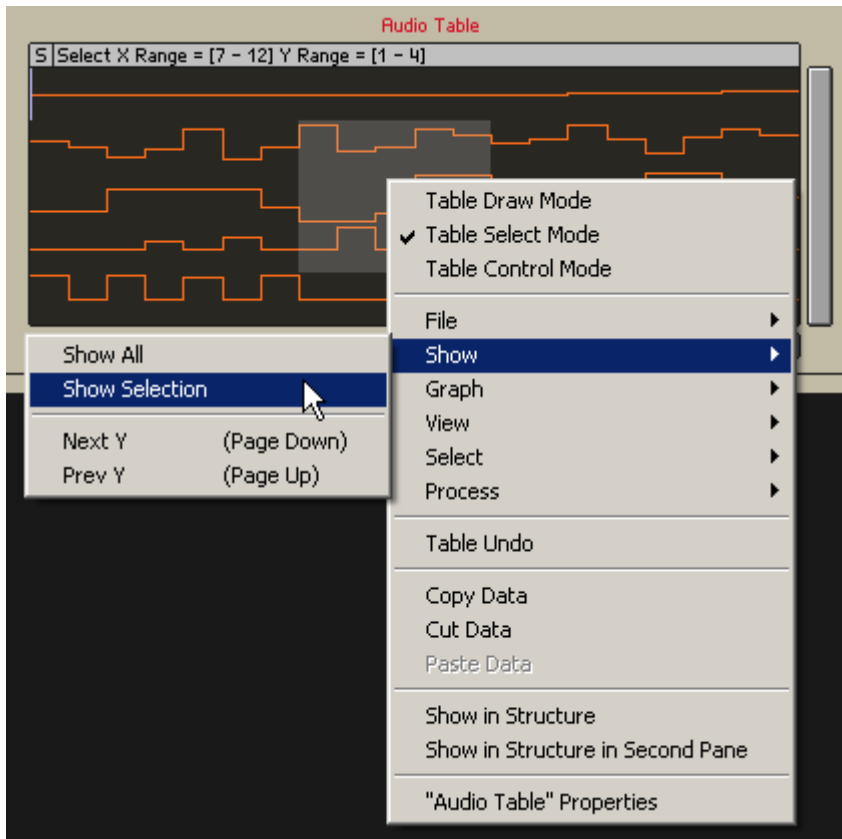
15. Similarly, when viewing a graph style that only shows the "X" dimension, you can use the *Value Auto Fit* checkbox (see above) to fit the vertical axis of the display. Engage the checkbox.
16. Now you should be back to the fitted table display that you started out with. To see how scrolling affects your display when viewing a 2-dimensional graph style, go to the Function page of the Table Module.
17. Select the *Multi Line* menu entry from the Graph menu.



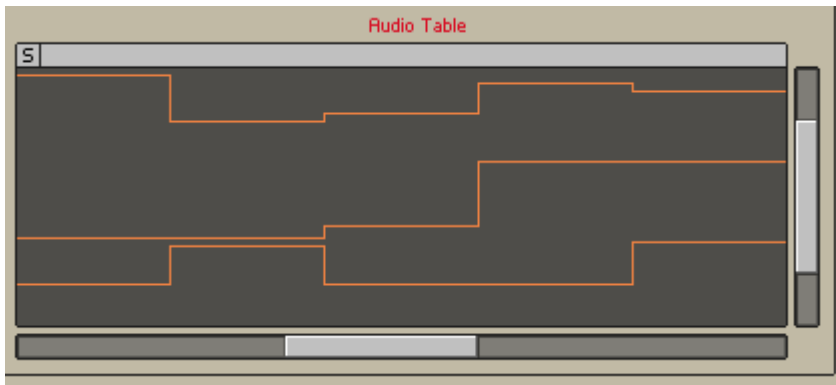
18. Engage the **Horizontal Scrollbar** and **Vertical Scrollbar** checkboxes to show the scrollbars from the Table Module's Panel representation again.
19. The horizontal scrollbar still functions the same way it did for the 1-dimensional graph style you just used. However, whereas for the one-dimensional case the vertical scrollbar affected the scaling of the values, in a 2-dimensional graph style the vertical scrollbar is used for the “Y” dimension. Scroll and zoom in the “Y” dimension as you did in the previous steps for the “X” dimension, leaving the final state as something different than you had in the beginning.
20. To fit the display in the “Y” dimension again, you can either double-click the scrollbar or engage the **Y Auto Fit** checkbox in the View page of the Table Module. Try both ways.

Zooming Table Data

1. Besides using scrollbars, there is another way to view particular areas of your table: zooming in on a selection in the table. Continuing from step 20 under the previous header, make sure that you have a Table Module with data in it. Go to the Table Select Mode by clicking the small boxed letter in the top left of the Table Module's Panel representation. You will know you have reached the Table Select Mode when that icon turns to an “S”.
2. Next, select an area of your table.
3. Now right-click anywhere on the table and select the *Show > Show Selection* menu entry.



4. The result should be something like what you see in the screenshot below. Notice that the area is still selected. To get back to where you were before, just double-click each scroll-bar or engage the X Auto Fit and Y Auto Fit checkboxes in the View page of the Table Module's Properties.



5. You have now become acquainted with the various ways of zooming the display area of your Table Module. Continue to the next subsection to learn about displaying a grid on your table.

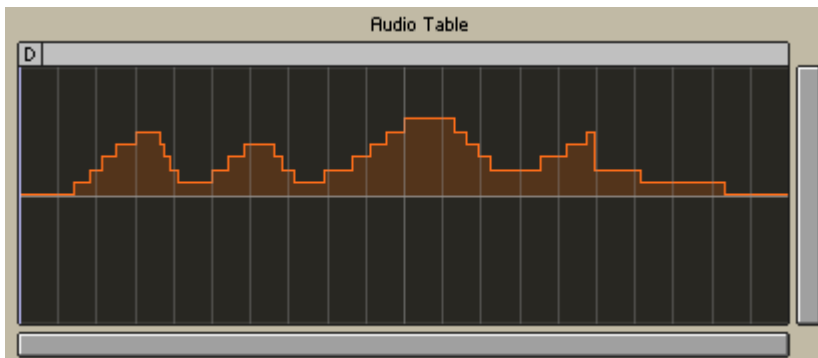
11.3.3 Displaying a Grid in the Table

To further help you with drawing and reading tables, REAKTOR allows you to display a grid in the Panel representation of Table Modules. To see how to set up a grid, follow these instructions:

1. Set up the environment to see how the table grid works in REAKTOR. To do this, create an Event Table Module with X Size “20” and Y Size “5”. The minimum and maximum values should be “-1” and “1”, correspondingly. Also, set the Stepsize parameter to “0.1”. To see a step by step instruction for these tasks, please refer to the first six instruction steps in subsection [↑11.3.1, Graph Style](#). Double-click the Event Table Module to enter the Module Properties.
2. Now look at the Panel representation of your Table Module. Right now you are in Table Control Mode, as indicated by the “C” in the top left corner of the table. Click that icon twice, skipping Table Select Mode, to reach the Table Draw Mode. The icon in the top left of the table should now be a “D”.
3. Use your mouse to draw data into the table.
4. Now enable the grid for the “X” dimension by engaging the X Grid checkbox as shown below.



5. Now that you have enabled the grid in the “X” dimension, you should see something like what is shown in the next screenshot. However, the grid in the figure is configured in a certain way. Look at the next step’s figure, to see the Grid Configuration table that holds the edit fields necessary to make your grid look like the one shown below. The leftmost column of edit fields is called the **Grid Step** column. The value entered here relates to the units set for “X” and “Y” in the Function page. It sets the spacing of the grid. Right now you see that the X Grid Step edit field holds the value “1”. This means that the basic resolution of the grid is one unit. The edit fields to the right of the Grid Step edit field refer to the basic grid resolution defined in the **Grid Step** edit field. If you liked to have every second unit as the basic resolution for the grid, enter “2”. If you liked to have two grid steps per unit, you should enter “0.5”. For now, make sure that the X Grid Step is set to “1”.



6. Now let's examine the other edit fields in the “X” row. There are four different line widths, or sizes, available for the grid lines: 1 is the finest and 4 is the thickest “line size”. The columns labeled “Sz1” to “Sz4” refer to these “line sizes”. The number you enter into the corresponding edit fields defines how often a line with that thickness is drawn (number of grid steps per line). Enter “1” for a line at every grid step, or “2” for a line at every other grid step, and so on. Enter “0” if you do not want to use a size at all. As you can see in the settings for the table above (the settings are

shown below), the grip step is set to “1”. The finest line coincides with every grid step, as indicated by the value “1” in the Sz 1 column. Similarly, the line with thickness 2 coincides with every second grid step, the line with thickness 3 with every fifth grid step, and the line with thickness 4 with every tenth grid step. Enter the numbers you see below into your edit fields. Feel free to experiment and find the grid that you enjoy the most, if you like.

Grid					
Grid	Step	Sz1	Sz2	Sz3	Sz4
X	1	1	2	5	10
Y	-	-	-	-	-
Value	-	-	-	-	-

- Now experiment with the value grid which you see in all graph styles besides the Multi Line and 2D Color graph styles. To see what the grid looks like in the “Y” dimension, choose the *Multi Line* menu entry from the Graph menu in the View page.
- The screenshot below shows a grid in the “Y” dimension. With this, you have reached the end of the instruction list regarding grid settings for Audio and Event Table Modules!

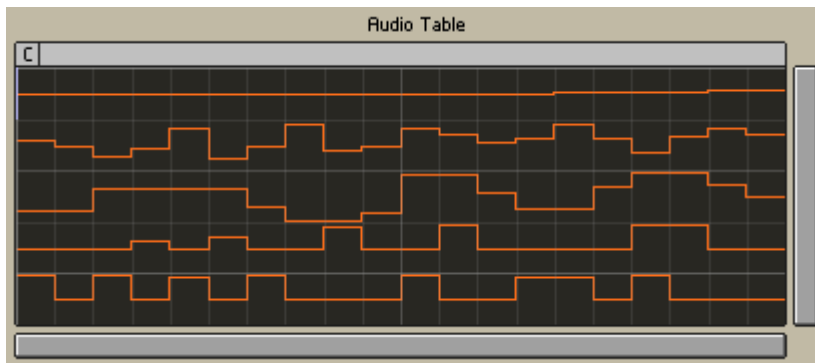


Fig. 11.6 A grid in the “Y” dimension as seen in the Multi Line graph style. Note that this grid has the edit fields for all sizes relating to the “Y” dimension set to “1”.

11.3.4 Setting Basic Visibility Settings for the Table Modules

This subsection will discuss two things. First, you will see how to make the Label, Picture, and Value parts of the Panel representation of the Table Modules visible or invisible. Second, you will learn how to change the size of the Panel representation of the Table Module.

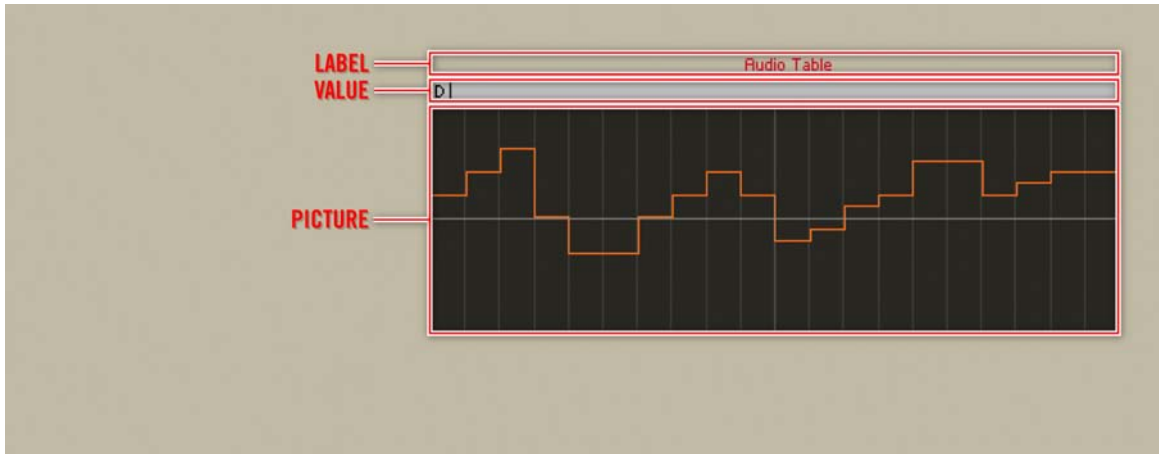


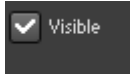
Fig. 11.7 The different parts of a Table Module's Panel representation.

As seen above, a Table Module's Panel representation consists of three main parts, the Label, the Value, and the Picture. Additionally, you can make the scrollbars appear or disappear (see subsection [↑11.3.2, Scrollbars and Zooming](#)) and change the size of the Panel representation display. To see how to do all of these tasks, look at the following instructions:

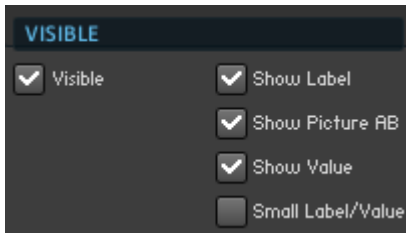
1. First, make sure you have an Audio Table Module from the *Built-In Module > Audio Modifier* menu placed into the Structure.
2. Next, double-click on the Module to open the Module Properties.
3. Click on the View button.



4. First, let's look at how to change the visibility of the different parts of the Panel representation. The whole Panel representation can be made invisible by clicking the Visible checkbox.



5. Disengage the checkbox and see the table disappear from the Panel. Engage the checkbox again to make it appear.
6. Now let's turn our attention to hiding and showing the parts of the table separately. This can be done with the help of the checkboxes shown below. Disengage the Show Label checkbox to see the control Label disappear.



7. Now, disengage the Show Picture checkbox to see the table display disappear.
8. Finally, you can even make the Value Bar that shows the table cell position and value disappear by disengaging the Show Value checkbox. Note that now clicking on the Visible checkbox to the right of the three checkboxes that you used last has no visible effect.
9. Engage the Show Label, Show Picture, and Show Value checkboxes to make them appear.
10. If the Value Bar at the top of the sample waveform display seem to large to you, you can reduce its size by engaging the Small Label/Show Value checkbox.
11. The last thing that you can change is the size of the table display. This is done by typing in the horizontal and vertical size in pixels into the Width and Height edit fields, respectively. These edit fields are shown below. This is it for the basic visibility options for the Table Modules!



Fig. 11.8 You can change the size of the table display on the Panel using the Width and Height edit fields.

11.3.5 Table Modes and Basic Table Editing Operations

In the previous subsections you have learned the necessary things to use the table in a Structure. However, there are a few more features that concern access to the table from the Instrument Panel. First, we will briefly outline the three Table Modes. Then you will learn how to manipulate table data from the Instrument Panel with the use of a few simple operations.

Table Modes and How to Switch Between Them

The Panel representations of the Table Modules can be set to operate in three different modes, the so-called Table Modes. They are the Table Control Mode, the Table Select Mode, and the Table Draw Mode. If the Value Bar is set to be visible on the Instrument Panel, you see a small box with a letter in it on the left edge of the Value Bar, as shown below. Just click on that box to switch to the next Table Mode. Alternatively, you can right-click on the table and select the corresponding menu entry from the context menu. To learn how to change the visibility of the Value Bar, please refer to subsection [↑11.3.4, Setting Basic Visibility Settings for the Table Modules](#).

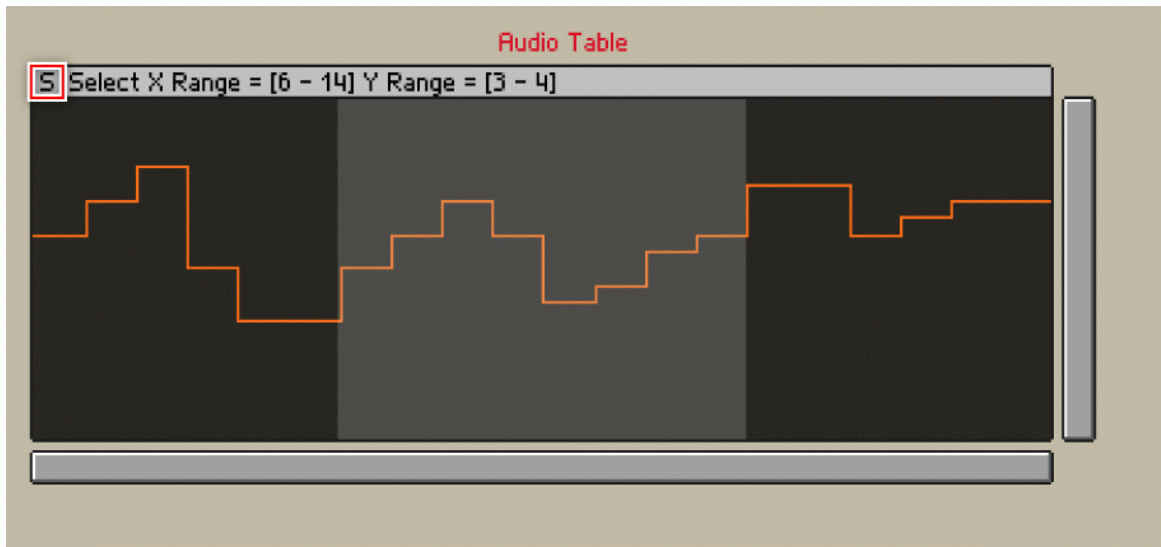


Fig. 11.9 The Panel representation of an Audio Table Module in Table Select Mode. You can switch between Table Modes by clicking on the letter enclosed by a box on the left edge of the Value Bar.

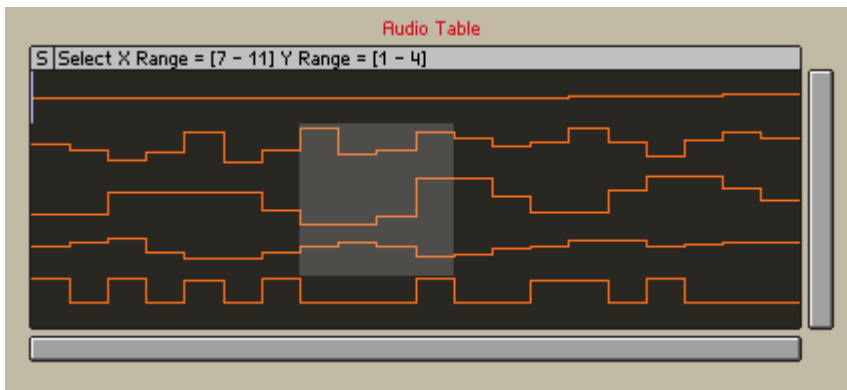
A brief description of the three Table Modes follows:

- The Table Control Mode, denoted by a “C” in the Value Bar, does not allow you to freely edit any information contained in the table. However, you can load and save table files as well as scroll and zoom.
- The Table Select Mode, denoted by an “S” in the Value bar, allows you to select different areas of the table. You can then copy, cut, and paste the data from and to the selected area. Additionally, you can zoom in on the selected area such that the selected data fills the whole display. This is done using the *Show > Show Selection* context menu entry.
- The Table Draw Mode, denoted by an “A” in the Value Bar, allows you to draw values into the table. To have the drawn values quantized to the resolution set in the Function page of the Module, right-click on the table and select the *Process > Quantize Draw Values To Step Size* menu entry.

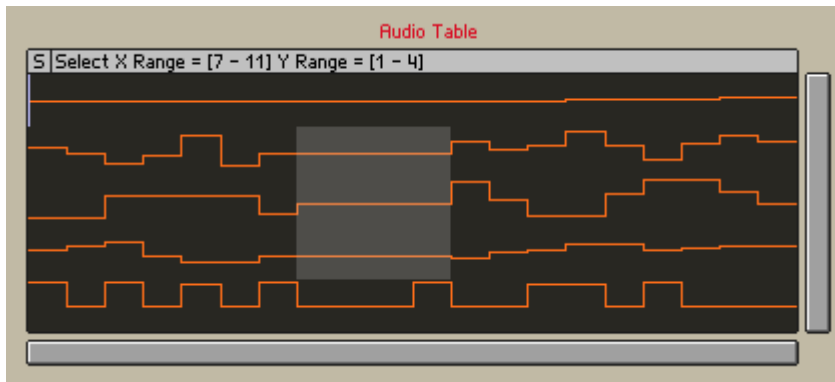
Table Manipulation in Table Select Mode

To see how to process and zoom in on selected table data from the Instrument Panel, follow these instructions:

1. Set up the environment to see how to manipulate the table data in Table Select Mode. To do this, create an Event Table Module with X Size “20” and Y Size “5”. The minimum and maximum values should be “-1” and “1”, correspondingly. Also, set the Stepsize parameter to “0.1”. To see a step by step instruction for these tasks, please refer to the first six instruction steps in subsection [↑11.3.1, Graph Style](#). Next, in the View page, choose the *Multi Line* menu entry from the *Graph* drop-down menu.
2. Now look at the Panel representation of your Table Module. Right now you are in Table Control Mode, as indicated by the “C” in the top left corner of the table. Click that icon twice, skipping Table Select Mode, to reach the Table Draw Mode. The icon in the top left of the table should now be a “D”.
3. Use your mouse to draw data into all the lines of the table.
4. Now that we have set up the table that we will work with, go to the Table Select Mode. The icon in the top left of the table should now be an “S”.
5. Select an area of the table.



6. To see how the copy, cut, and paste operations work, let's try cutting and pasting. Right-click the table and choose the *Cut Data* menu entry. The values within the cut table cells are replaced with the Default Value, as shown below.



7. Now select an identical area at some other location of the table.
8. Right-click the table and choose the *Paste Data* menu entry. The values of the cut cells should reappear in the new area, as shown below. If the selected area to which you want to paste the data is smaller than the data to be pasted, REAKTOR will match the top-left cells of both areas and paste the data to the cells which are available. Everything else is clipped.

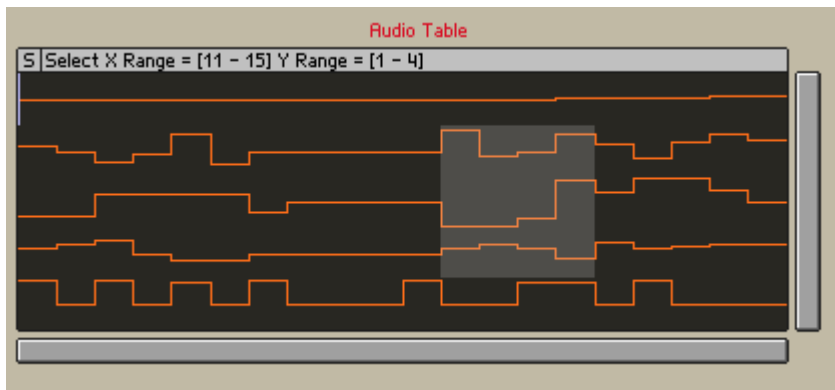


Fig. 11.10 With the Paste Data menu entry you can paste the values from copied or cut cells to the cells in the selected area.

Shifting Table Entries and Scaling Values

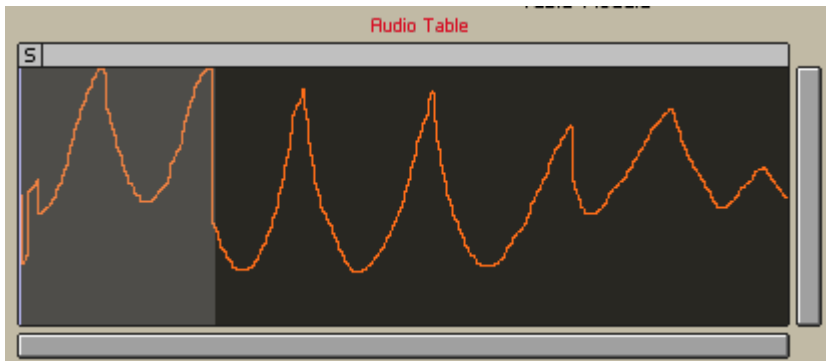
1. From the Instrument Panel you can conveniently shift data in the tables. Make sure you are in Table Select Mode and have an area of your table selected, as shown above.
2. While holding down the Shift key, click and drag left and right on the table. Notice how the table entries shift left and right, respectively and wrap around the borders of the selected area.
3. Also, you can scale table values of selected table cells. Again, make sure you have an area of your table selected, as shown in previous section's Audio Table.
4. Now, while holding down the Ctrl key if you are on Windows (Cmd key if you are working in OS X), click and drag up and down on the table. Notice how the values are scaled between the minimum and maximum values.



Scaling values using the Ctrl modifier key if you are on Windows (Cmd key if you are working in OS X) is particularly useful when you are using the 2D Color Graph display. This way you can quickly modify the values of single or multiple table cells.

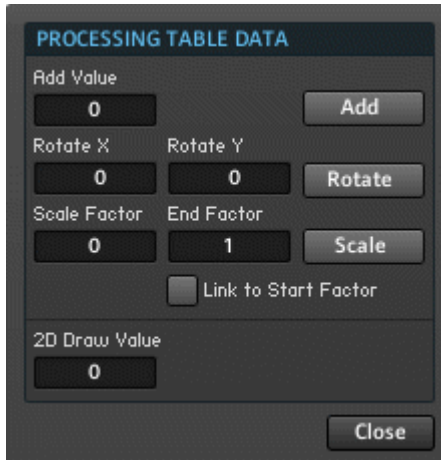
Fade-In and Fade-Out and Other Processing Options

1. You can use dynamic scaling to apply the “fade-in” and “fade-out” effect to your table values. To do this, be sure you are in Table Select Mode and have an area of your table selected, as shown below.



2. Now right-click on the Table background and choose the *Process > Set 2D Draw Value...* menu entry.

- To do a complete “fade-in”, you need to set the scaling start value to “0” and the end scaling end value to “1”. These two values are set using the **Start Factor** and **End Factor** edit fields, respectively (shown in the screenshot below). To be able to edit the fields individually, make sure the **Link to Start Factor** checkbox is disengaged.



- Now press the **Scale** button.
- Press the **Close** button to close the Processing dialog window. Your Table should now exhibit a “fade-in” effect, as shown in the screenshot below.

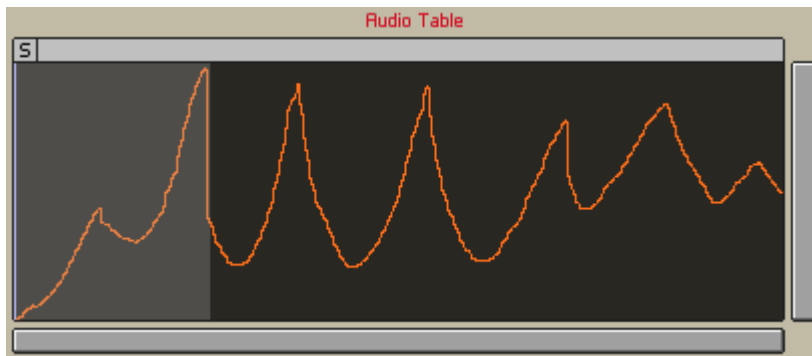


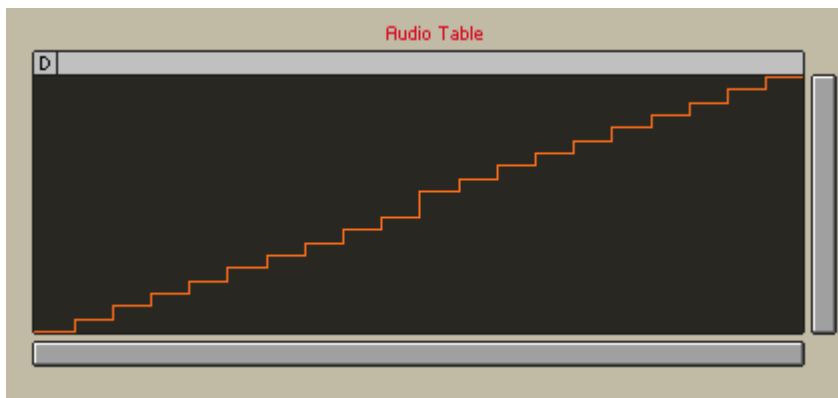
Fig. 11.1 Table data with a “fade-in” scaling effect.

Other table data manipulation possibilities in the Processing dialog include:

- Adding a constant value to all of the selected table cells. This is done using the **Add Value** edit field.
- Rotating the table values through table cells in the X and Y dimensions. The extent of the rotation in the X and Y dimensions is set with the **Rotate X** and **Rotate Y** edit fields, respectively. The rotation command is executed by pressing the **Rotate** button. For example, rotating the data shown in the screenshot above by a positive value shifts all values to the right, but within the selected bounds. Cells shifted beyond the right edge are “wrapped around” to the left edge.

Drawing Straight Lines

1. The Panel representation of the Table Module allows you to draw straight lines between two points in the visible part of the table. To see how this works, make sure you are looking at a table that is in Table Draw Mode. The graph style should be Line, as shown below.
2. While holding down the Shift key, click and drag from one part of the table to another part. Notice how a straight line is drawn between the two points. This is shown in the screenshot below, where a straight line has been drawn from the bottom left corner to the top right corner.



- Also, you can draw a perfectly horizontal line over table cells. While holding down the Ctrl key if you are on Windows (Cmd key if you are working in OS X), click a table cell and drag left or right on the table. Notice how the values of all cells that you cross are set to the value corresponding to the point where you first clicked.

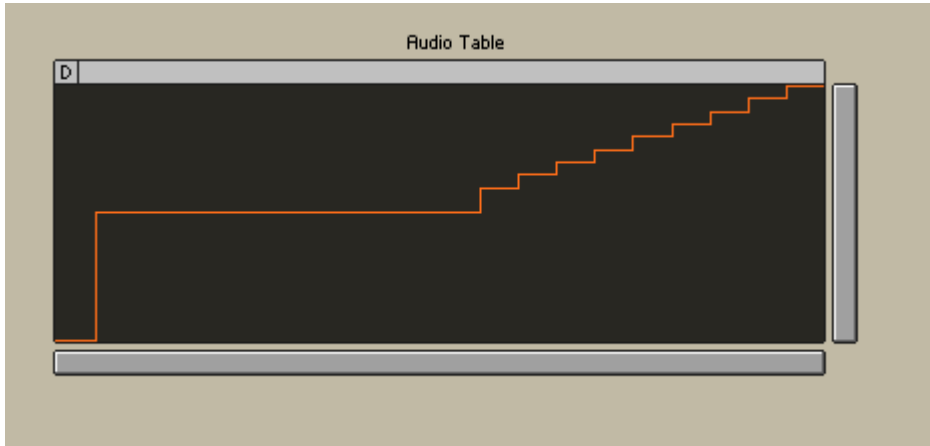


Fig. 11.11 A perfectly horizontal line has been drawn using the Ctrl modifier key on Windows (Cmd key if you are working in OS X).



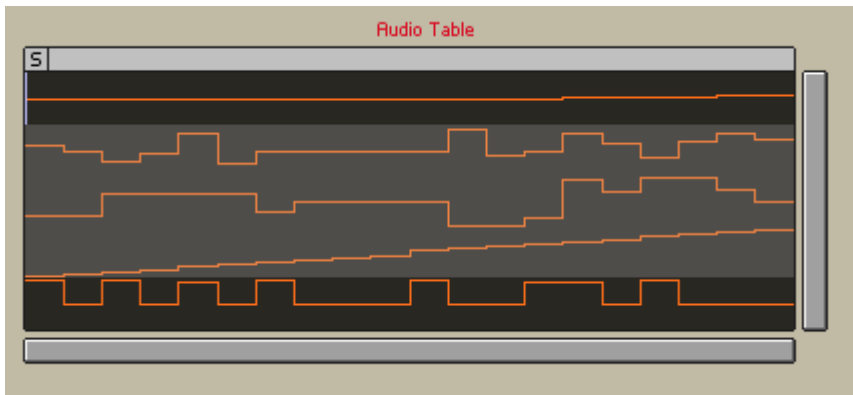
The features discussed under this header only apply to 1-dimensional graph styles.

Table Undo and Redo

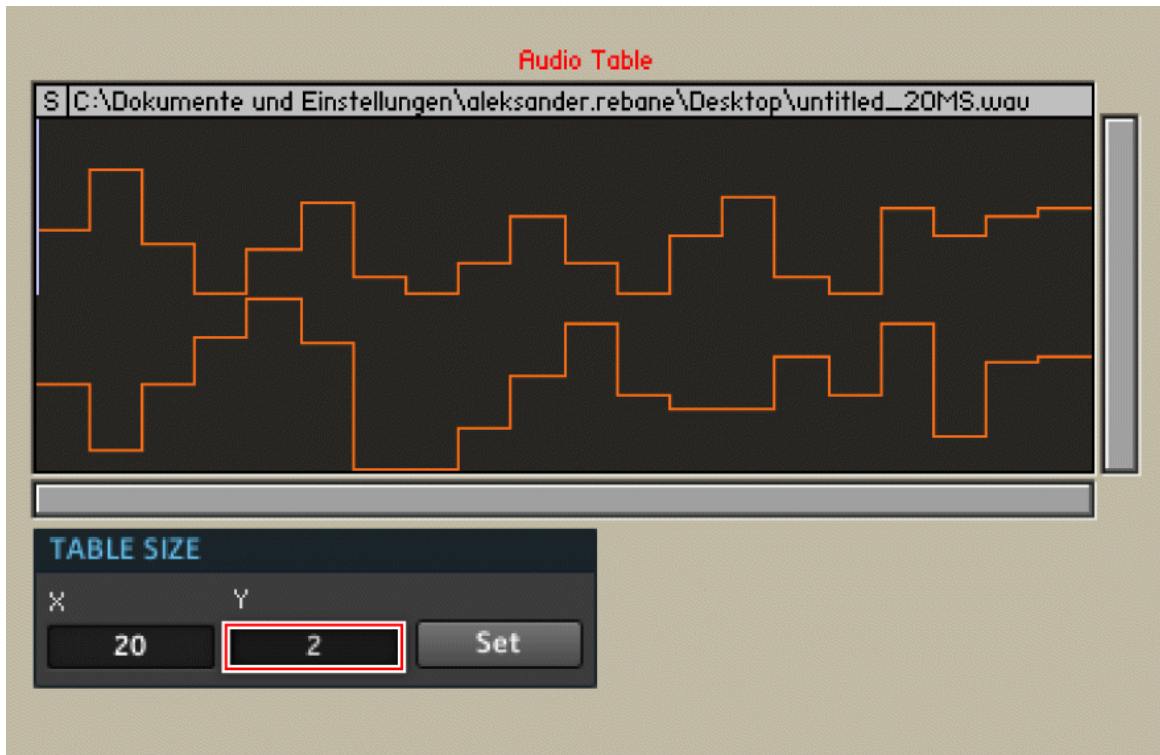
- To undo the table operation you just did, right-click the table and select the *Table Undo* menu entry. Try this.
- Redo by clicking on the same menu entry again.

Working with Table Rows and Columns

- Now we will see how to select whole rows and columns of the table. Select some table cells.
- Right-click on the table and choose the *Select > Select X All* menu entry.
- The result should look something like what is shown below. The whole rows to which the selected table cells belonged to become selected with the *Select X All* menu entry. Similarly you can select columns, using the *Select Y All* menu entry. Try this.



4. Also, you can select all visible table cells using the *Select All* menu entry. But now let's see how to delete rows from the table. Select some table cells that belong to the rows you would wish to delete.
5. Now right-click on the table and choose the *Process > Delete Rows* menu entry. The below table results from deleting the three rows highlighted in the Audio Table above.



6. Finally, to insert two rows into the table, select two table cells from different rows.
7. Now right-click the table and choose the *Process > Insert Rows* menu entry. You have now learned how to edit the table from the Instrument Panel using the Table Select Mode!

Working With Table Files from the Instrument Panel

Not only can you draw and manipulate the table data from the Instrument Panel using the Table Draw and Table Select Modes, but you can also load, reload, and save table data to and from files. This is possible in all Table Modes. To save the data contained in the table as a table file, right-click the table and choose the *File > Save Table Data* or *File > Save Table Data As...* menu entries depending if you want to save the data in a new file or not. To load new data from a table file into the table, right-click the table and choose the *File*

> *Load Table Data into Table...* menu entry. If you have edited a table that is associated with a table file and would like to restore the content of the table to match the table file, right-click the table and choose the *File > Reload Table Data* menu entry.

12 Working with the Sine and Modal Bank Modules

REAKTOR supports two advanced synthesis methods: additive synthesis and modal synthesis. For these synthesis methods the Sine and the Modal Bank Modules, respectively, have been implemented in the Primary Level. These two Modules create opportunities for Instrument builders to generate sound in a completely new way. In this chapter you will learn about some of the applications for these Modules and how to use them. However, it should be noted that these are rather advanced Modules. Not only does one need builder know-how to implement working Structures with them, but it is also essential that the Modules are supplied with the right values before they begin to produce musical results. This chapter will not attempt to give a comprehensive discourse on building Instruments with the Sine Bank and Modal Bank Modules but will teach you the principles which govern them.

12.1 Working Principles of the Sine and Modal Bank Modules

Although the two Modules fundamentally differ from each other in the way they generate sound, there are many similarities between the ways that they are used in a Structure. The below screenshot shows the Sine Bank and Modal Bank Modules. Comparing the input and output ports of the two Modules, you see that both have the same output ports and “Num”, “Idx”, “P”, “Rtio”, “AL”, “AR”, and “App” among the input ports. In this section we will first concentrate on the similarities of the two Modules so that you will learn how these ports are tied to the principles behind the Modules. Then you will see how to build the simplest Structure for each Module to start getting your own custom sound out of it.

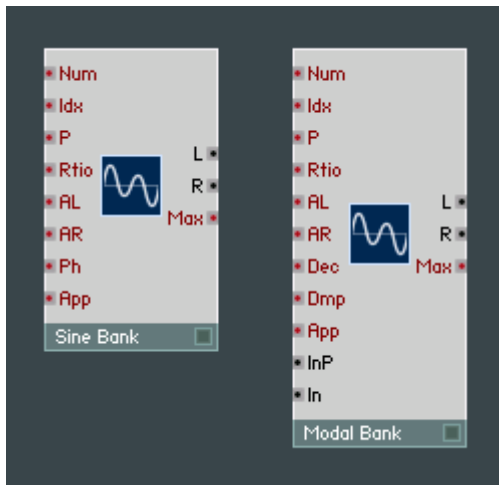


Fig. 12.1 The Sine Bank and Modal Bank Modules are accessed and configured in a similar way from the Primary Level.

Setting the Parameters of Partial

Let's look at the Sine Bank Module to learn how the Sine and Modal Bank Modules are used in a Structure. As the name implies, the Sine Bank Module embodies a bank of an (almost) arbitrary number of sine oscillators, called partials. The amplitudes, frequency ratios, and phases of these partials are specified by the “AL” and “AR”, “Rtio”, and “Ph” inputs, respectively. The resulting sine signals are then mixed together and sent to the “L” and “R” outputs for the left and right stereo channel signals of the Sine Bank Module. The typical steps that need to be taken before a Sine Bank Module can send out a desired signal are as follows:

1. Activate the Module by connecting the “L” and “R” output ports to an active signal flow that, preferably, is audible at the output of your soundcard.
2. Set the number of partials of the Module using the “Num” input port or the corresponding edit field in the Function page.
3. Specify the amplitude, frequency, and phase, for all partials.
4. Tell the Module from which fundamental pitch the frequency ratios of the partials are calculated. This is done by sending the corresponding pitch value to the “P” input port.

5. Send an “apply” Event to the “App” input port to commit the changes you made to the partial parameters.

The trickiest part is sending the amplitude, frequency, and phase information for each partial without mixing up the correct order of Events. Each partial has an integer index associated with it, with “1” for the first partial, “2” for the second partial, and so on. Before sending any amplitude, frequency, or phase values to the Sine Bank Module, it needs to know for which partial these values are meant. Because of this, you need to send the partial number to the “Idx” input port *before* sending the parameters for that partial to the corresponding input ports. Let's look at a simple example with two partials. Here we set the first partial to the fundamental frequency defined by the “P” input port and give it unit amplitude in both stereo channels. For the second partial we want to have double the frequency of the first partial and an amplitude factor of “0.5”. For this you would send the Events to the Module's inputs in the following order (you can build a Structure that realizes this using Constant and Knob Modules; the Events are sent from the Knobs when you turn them on the Instrument Panel):

1. First you need to specify the number of partials. For this purpose, make sure that you send an Event with the value “2” to the “Num” input port of the Module and that the Max. Partial edit field in the Module's Function Page is greater or equal to “2”.
2. After having specified the number of partials, send an Event with the value “1” to the “Idx” input port. This selects the first partial.
3. Next, send an Event with the value “1” to the “Rtio” input port. This causes the frequency of the first partial to have a ratio of “1” to the fundamental frequency which is determined by the value at the logarithmic pitch input port, “P”.
4. Now send an Event with value “1” to the “AL” and “AR” input ports. This causes the amplitude of that partial to be “1” in both the left and the right stereo channels.
5. To choose the second partial, send an Event with the value “2” to the “Idx” input port.
6. Next, send an Event with the value “2” to the “Rtio” input port to set the frequency of the second partial to have a ratio of “2” to the fundamental frequency set at the “P” input port. In this case the second partial would have a ratio of “2” also in respect to the first partial, that is, have twice the frequency of the first partial.
7. Then send an Event with the value “0.5” to the “AL” and “AR” input ports to have amplitude “0.5” on both stereo channels.

8. Finally, you also need to specify the fundamental pitch with an Event at the “P” input port. You could use a Note Pitch Module for this.
9. To commit the changes, send an Event with the value “0” to the “App” input port (Sending different values to the “App” input port commits the changes made at different input ports).
10. As you probably noticed, the “Idx” value has to be set once to choose a partial and after that you can set the parameters of that partial as many times as you like without having to send another value to the “Idx” port. The same procedures for setting the amplitudes and frequencies of the partials as shown above also apply to the Modal Bank Module. Whereas the third partial-specific parameter for the Sine Bank Module is the phase (specified at the “Ph” input port), the Modal Bank Module has the decay parameter (specified at the “Dec” input port). These parameters are set the same way as the frequency and amplitude values, although the scale and range of values differs.



The maximum number of partials for both Modules is 10'000. Increasing the number of active partials also increases CPU load.

12.2 Building a Simple Structure with the Sine Bank Module

In this tutorial you will learn how to build a very simple additive synthesizer which has 5 partials. For each partial you have two faders controlling the amplitude and the frequency, respectively. Since the Sine Bank Module functions much like a conventional oscillator, we will apply an amplitude envelope to its output signal. The tutorial will be split up under a number of subsections so that you can later more easily find the task that you are looking for. Start the tutorial by performing the following steps.



Many building principles also apply to the Modal Bank Module.

12.2.1 Setting the Maximum Number of Partial

1. Create a new Ensemble and go to the Structure View.
2. Right-click on the Structure background and choose the *Built-In Module > Oscillator > Sine Bank* menu entry.

3. Double-click on the Sine Bank Module to open its Properties and go to Function page.

FUNCTION

4. Since we specified that we would like to use 5 partials, double-click the Max. Partial edit field shown in the screenshot below and enter the value “5” into it. Now any value at the “Num” input port (which specifies the number of active partials in the oscillator) that goes beyond “5” will be clipped. Also, if nothing is connected to the “Num” input port, the number of partials will automatically be set to “5”.

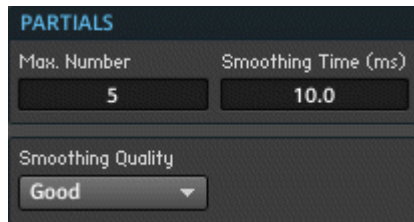
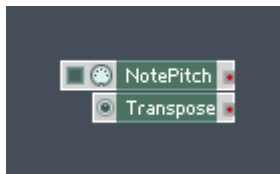


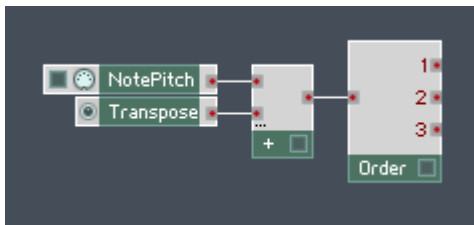
Fig. 12.2 The maximum number of partials is set with the Max. Number edit field in the Function page of the Sine Bank Module. This is also the number of partials that automatically is chosen if nothing is connected to the “Num” input port.

12.2.2 Sending Pitch Information to the Sine Bank Module

1. Next you will want to control the pitch of your Sine Bank oscillator. For that purpose, insert the Note Pitch Module from the *Built-In Module > MIDI In* submenu.
2. If you are using a computer keyboard to play your up and coming synth, the range is rather limited. To increase the range you would use a Panel control that transposes the notes played. Insert a Knob Module from the *Built-In Module > Panel* submenu into the Structure.
3. Double-click the Knob Module to view its Properties.
4. Change the Label of the Module to “Transpose”. The Note Pitch and Knob Modules should now look like the ones shown below.



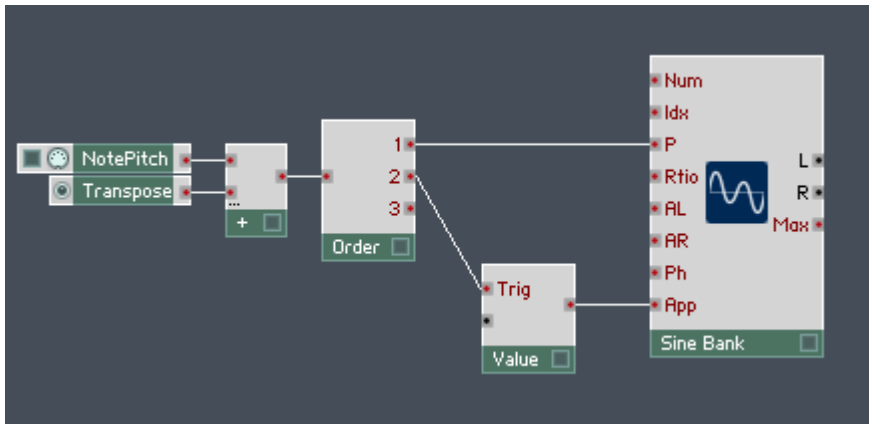
5. Since we are simply transposing the pitch coming from the Note Pitch Module, we add the transposition value and the note pitch value together. For that, insert an Add Module from the *Built-In Module > Math* submenu into your Structure.
6. Now connect the output ports of the Note Pitch Module and the Knob Module to the two input ports of the Add Module.
7. As discussed in the previous subsection, after new values are sent to the input ports of the Sine Bank Module, they will not affect the oscillator until the changes are applied. This also counts for the pitch value which we will send to the “P” input port. Note that the order of the Event carrying the pitch value and the “apply” Event is relevant: the “apply” Event should arrive at the “App” input port of the Sine Bank *after* we told it what the pitch should be. This sounds like a case for the Order Module! Insert one from the *Built-In Module > Event Processing* submenu into your Structure.
8. Now connect the output port of the Add Module to the input port of the Order Module. The result should look like the screenshot below. We will have the “1” output port of the Order Module send the pitch value to the “P” input port of the Sine Bank. Subsequently the “2” output port of the Order Module will send an Event that should trigger a suitable value to be sent to the “App” input port of the Sine Bank Module.



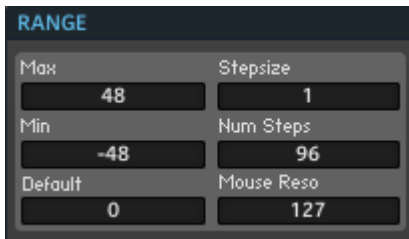
9. Connect the “1” output port of the Order Module to the “P” input port of the Sine Bank.
10. Next, we want the “2” output port to trigger an Event to be sent to the “App” input port. For that, insert a Value Module from the *Built-In Module > Event Processing* submenu into the Structure.
11. We are fine with the value “0” to be sent to the “App” input port (this applies the values at all input ports but doesn't reset the smoothers — see sections [↑12.4.2, Ports](#) and [↑12.5.2, Ports](#) for more information on the input ports of the Sine and Modal Banks). Hence go ahead and connect the output port of the Value Module to the

“App” input port of the Sine Bank and leave the lower input port of the Value Module disconnected (a disconnected port is the same as having a “0” Constant Module connected to it).

12. Now connect the “2” output port of the Order Module to the “Trig” input port of the Value Module. Your Structure should now look like the one shown below.



13. Right now the “Transpose” Knob has a very limited range. To increase that range you need to go to the Function page of the Knob. Double-click the Knob Module.
14. For our purpose, having a transposition range of 4 octaves in both directions should be enough. 1 octave equals 12 semitones, so 4 octaves equals 48 semitones. Double-click the Max edit field and enter the value “48”.
15. Double-click the Min edit field and enter the value “-48”.
16. Now set the Stepsize parameter to “1”. The Range area of your Module's Function page should now look like the one shown below.



17. One could have the Knob Module appear fully on the Instrument Panel, but since the transposition parameter is rather minor, we would like to only see the value field as a Panel representation. To change this, first click on the View button.



18. Now disengage the Show Picture checkbox.

12.2.3 Creating a Macro for the Amplitude Controls of the Partial

1. We will now turn to creating the controls for the amplitudes of the five partials. Since we will use a similar Structure for sending the parameter changes and the “apply” Event for each amplitude control as we did for the pitch, you can imagine that the Structure can grow quite large. Because of this we will pack the amplitude controls in a Macro. Insert an empty Macro into the Structure by right-clicking the background and selecting the *Macro > _New - Empty* menu entry.
2. Next, go to the Properties of the Macro and change its Label to “Amplitudes”. Your “Amplitudes” Macro should now look like the one shown below.



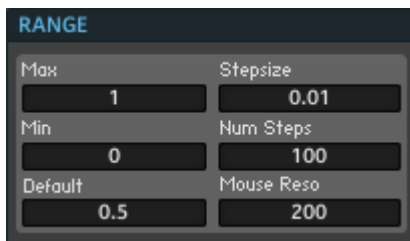
3. Double-click the Macro to enter its Structure.
4. In the previous subsection you learned that in order to make an parameter change that is also recognized by the corresponding partial, you need to send three Events in the following order to the Sine Bank: the Index value to the “Idx” input port, the parameter value to the corresponding input port, and an “apply” Event to the “App” input port. We will generate all three Events in this Macro. Therefore we also need the Macro to have three output ports. Insert three Out Port Modules into the Structure. You can find them in the *Built-In Module > Terminal* submenu.
5. Now change the Labels for the Out Port Modules to the following (listed from top to bottom): “Idx”, “Amp”, and “App”.



Fig. 12.3 When sending a value from an amplitude control to the Sine Bank Module, you need the index value, amplitude value, followed by an “apply” Event (in that particular order). For each new amplitude value, these three Events are sent from the “Idx”, “Amp”, and “App” output ports of the Macro, respectively.

12.2.4 Creating an Amplitude Control for a Partial

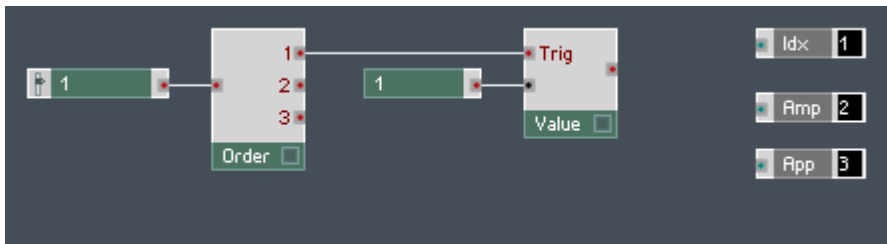
1. We will create a Fader for the amplitude parameter of the first partial. Right-click the Structure background and select the *Built-In Module > Panel > Fader* menu entry.
2. Go to the Properties of the Fader Module and change the Label to “1”. The output of the Fader serves to set the amplitude of the first partial. As briefly mentioned in the previous subsection, the amplitudes for the left and right stereo channels are specified at the “AL” and “AR” input ports of the Sine Bank Module, respectively, and the input value should be in the linear scale.
3. Now look at the Range area of the Module's Function page. A typical range for the amplitude of a partial is [0 ... 1], a suitable range for the linear scale. Additionally, 100 steps should offer a good enough resolution for our purposes. Make sure that the edit fields in the Range area of your Module's Function page have the same or similar values as to the ones shown below.



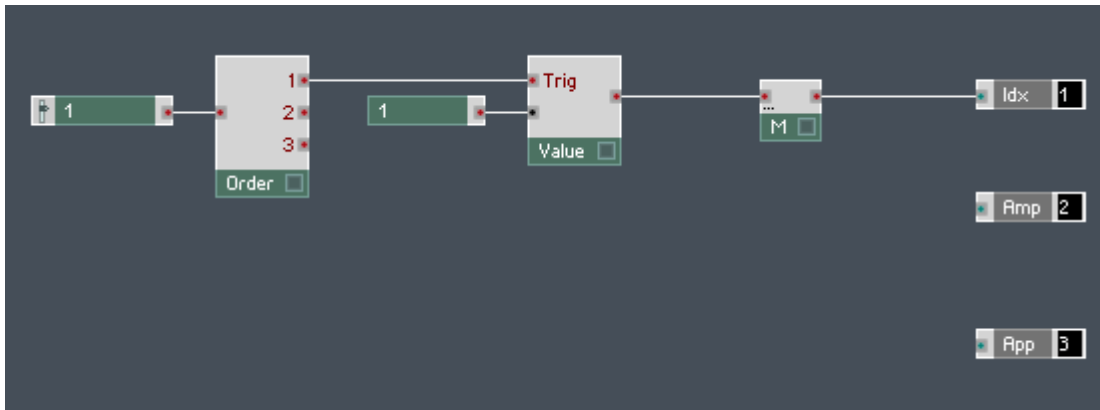
4. As previously in the case of the pitch parameter, we will use an Order Module to send the Events to the Module in the correct order. Insert an Order Module from the *Built-In Module > Event Processing* submenu into the Structure.
5. Now connect the output of the Fader Module to the input port of the Order Module.
6. We remember that the index of the partial has to be sent to the “Idx” input port of the Sine Bank Module before the actual parameter arrives. We will let the “1” output port of the Order Module trigger the index value to be sent. The actual triggering will be done by a Value Module. Insert one into the Structure, you will find it in the *Built-In Module > Event Processing* submenu.



7. The trigger signal from the “1” output port of the Order Module should go to the “Trig” input port of the Value Module. Drag a wire between these two ports.
8. Upon a trigger Event, the Value Module should send the index of the partial. Since the Fader Module that is connected to the Order Module controls the first partial, the “Idx” value is then “1”. Because of this, right-click on the lower input port of the Value Module and select the *Create Constant* menu entry.
9. Make sure that the Constant has the value “1”.



10. Since we will have five controls in all, we need to merge the Events coming from all five sources before being sent to the corresponding Out Port Modules. For this, insert three Merge Modules into the Structure; you will find them in the *Built-In Module > Event Processing* submenu.
11. Place a Merge Module in front of each Out Port. Connect the output port of the Value Module to the input port of the Merge Module and in turn the output port of the Merge Module to the “Idx” Out Port.



12. After sending the trigger for the index value, the Order Module should send the actual value of the amplitude parameter. This should reach the “Amp” Out Port. Therefore drag a wire from the “2” output port of the Order Module to the input port of the Merge Module that sits in front of the “Amp” Out Port and in turn connect the output port of the Merge Module to that Out Port.
13. Last, the Order Module should send a trigger signal that ends up causing the “apply” Event to be sent to the “App” input port of the Sine Bank Module. You already built a Structure for the “apply” Event of the pitch change using a Value Module, so all you need to do is route the trigger Event from the “App” output port of the current Macro to that Value Module. Connect the “3” output port of your Order Module to the input port of the Merge Module that sits in front of the Out Port labeled “App” and in turn connect the output port of that Merge Module to the “App” Out Port Module. The Structure of your Macro should now look like the one shown below.

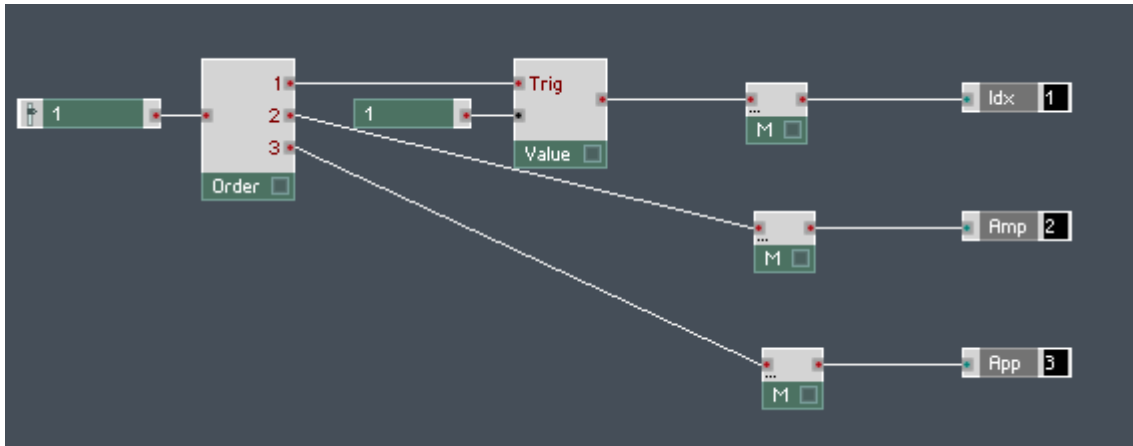
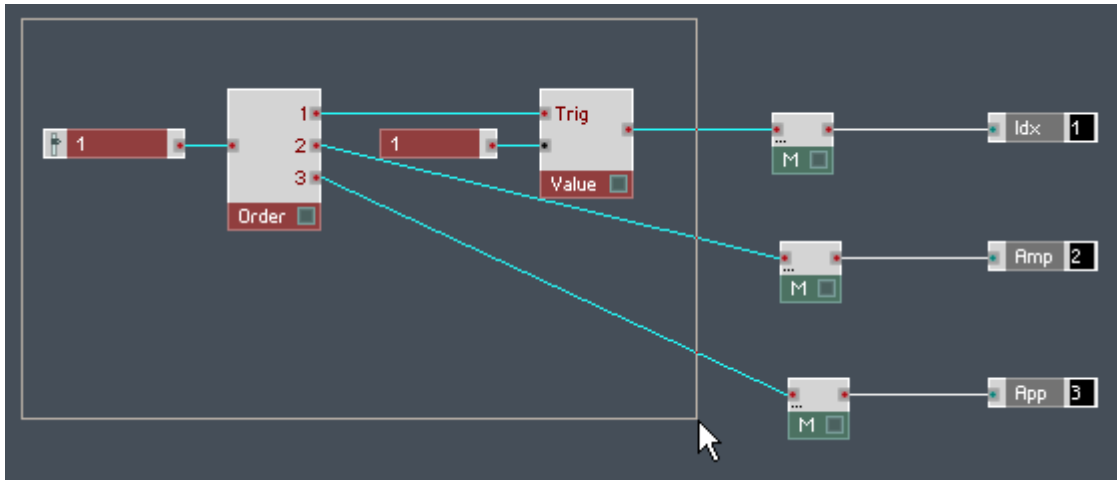


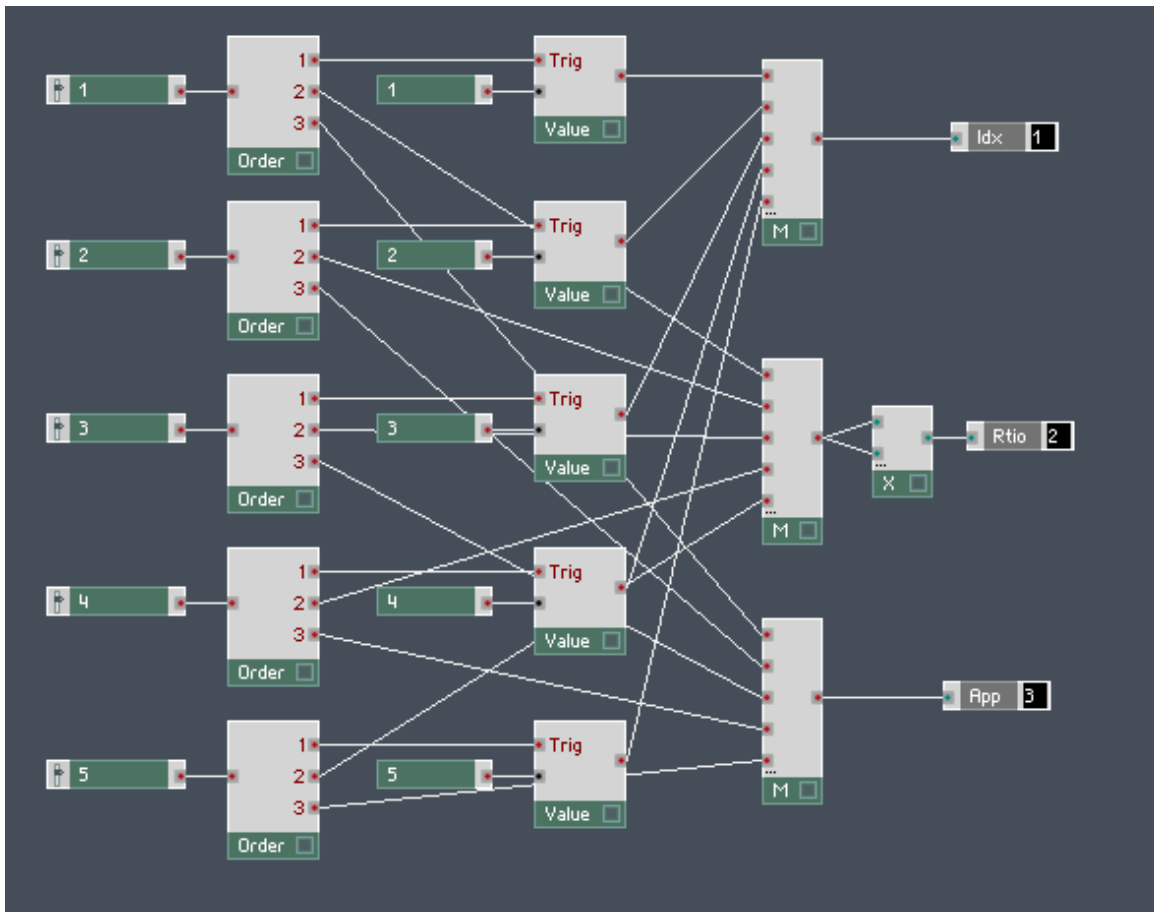
Fig. 12.4 In this state the “Amplitudes” Macro is capable of sending all the Events necessary to set the amplitude of the first partial of the Sine Bank Module.

12.2.5 Duplicating the Structure of the First Amplitude Control

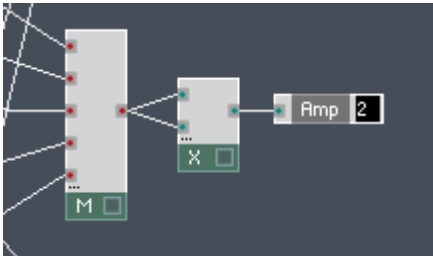
1. Actually, you have already done the main work for the contents of this Macro. Now you can just duplicate the Structure that you built for the first partial for the second, third, fourth, and fifth partials. Select the “1” Fader Module, the Order Module, the “1” Constant, and the Value Module. You can do this by clicking on the corresponding Modules while holding Ctrl for Windows (Cmd if you are on Mac OS X) or by clicking and dragging a rectangle around these Modules.



2. Right-click on one of the selected Modules and choose the *Duplicate* menu entry.
3. Repeat the previous step three more times.
4. Now you should have five copies of the Fader-Order-Constant-Value Structure. Rename the Faders and change the values of the corresponding Constant.
5. Next, make the same connections to the Merge Modules as you did for the original copy of the Structure. This time, to automatically create more input ports for the Merge Module, hold Ctrl for Windows (Cmd if you are on Mac OS X) while dragging the wires to the corresponding input port area of the Merge Module. Remember that the output of the Value Module should be routed to the Merge Module sitting in front of the “Idx” Out Port, the “2” output port of the Order Module should be routed to the Merge Module sitting in front of the “Amp” Out Port, and the “3” output port of the Order Module should be routed to the Merge Module sitting in front of the “App” Out Port. The resulting Structure should look like the one shown below.

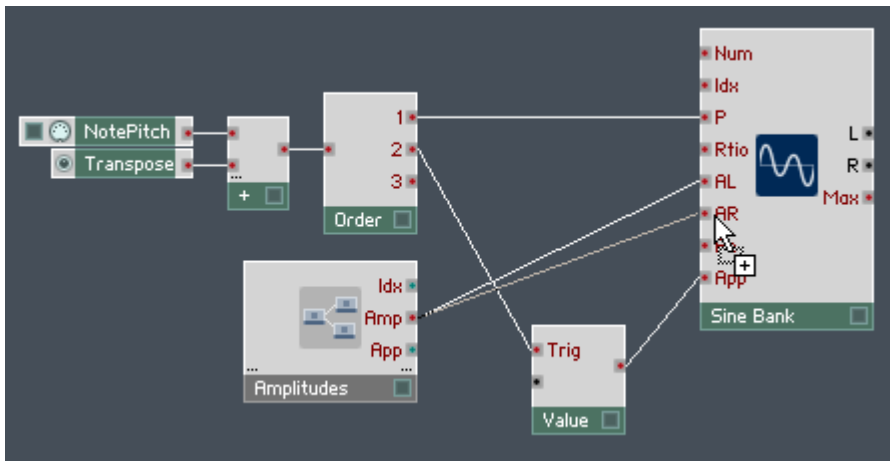


6. As a final detail to this Macro we will scale the amplitude values quadratically. This will give the Faders a more natural feel to them, since they will have higher resolution for low values and less for high values, which is similar to how humans perceive loudness. Insert a Multiply Module from the *Built-In Module > Math* submenu into the Structure.
7. Now connect the output of the Merge Module that sits in front of the “Amp” Out Port to the two input ports of the Multiply Module.
8. Last, connect the output port of the Multiply Module the Out Port.



12.2.6 Routing the Amplitude Control Parameters to the Sine Bank Module

1. Double-click the Macro's Structure background to go to the upper level of the Structure.
2. Now we connect the output ports of the “Amplitudes” Macro with the Sine Bank Module. Let's start with the easiest case, the “Amp” output port, which delivers the amplitude factor for the partials. The Sine Bank Module receives the amplitude values for the left and right stereo channels separately at the “AL” and “AR” input ports, respectively. To keep things simple, we are going to build a mono (as opposed to stereo) synth, so the amplitude values for “AL” and “AR” will be the same. Drag a wire from the “Amp” output port of the Macro to the “AL” and another one to the “AR” input port, as shown below



3. Next, we will connect the index value. Remember though, that we still have to specify the frequency ratios for the partials and these parameters, too, need to be preceded by an index value. Therefore place a Merge Module (*Built-In Module > Event Processing* submenu) near the “Idx” input port of the Sine Bank Module. This way we can merge the index values coming both from the “Amplitudes” Macro and the “Frequencies” Macro that you will soon build.
4. Connect the “Idx” output port of the “Amplitudes” Macro to the input port of the Merge Module and the output of the Merge Module to the “Idx” input port of the Sine Bank Module.
5. Last, we want to route the trigger Event from the “App” output port to the “Trig” input port of the Value Module. Since that input port already has a connection, we will need another Merge Module to guarantee that both triggers (resulting from a change in either pitch or amplitude) reach the “Trig” input port of the Value Module. Insert a merge Module from the *Built-In Module > Event Processing* submenu into the Structure.
6. Connect the “2” output port of the Order Module to the input port of the Merge Module.
7. Drag a wire while holding Ctrl for Windows (or Cmd for Mac OS X) from the “App” output port to the input port area of the Merge Module, creating a second input port.
8. Finally, drag a wire from the output port of the Merge Module to the “Trig” input port of the Value Module. Your Structure should more or less look like the one shown below.

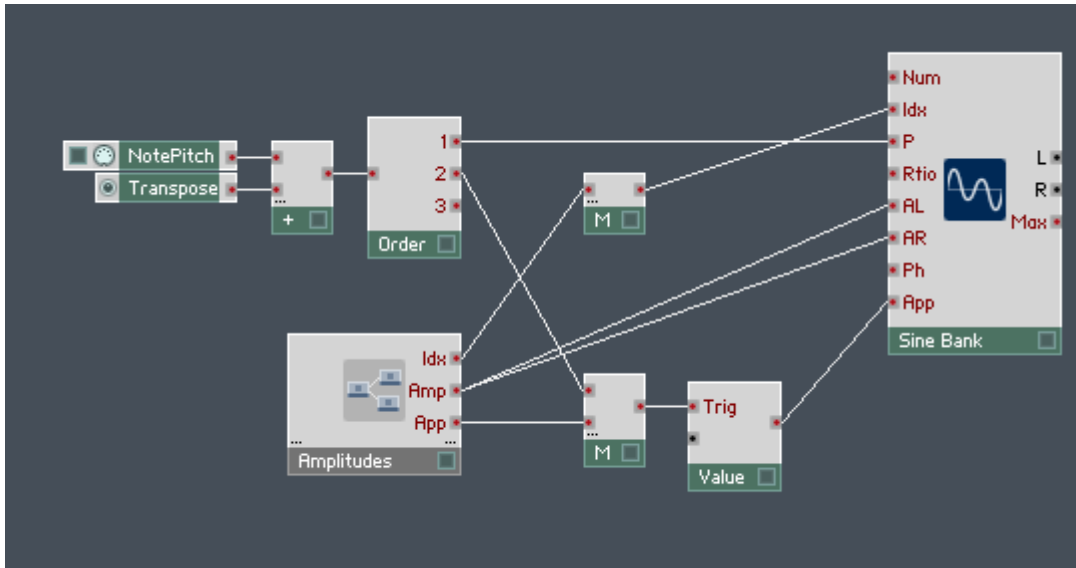


Fig. 12.5 This Structure features two functionalities. One is setting of the fundamental pitch of the Sine Bank. The other, including the Structure in the “Amplitudes” Macro, is setting the amplitudes for the five partials of the Sine Bank Module.

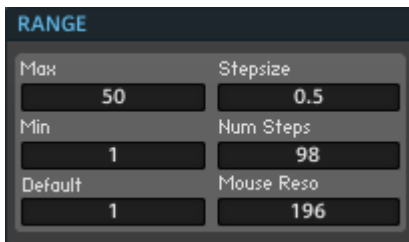
12.2.7 Creating Frequency Ratio Controls for Partial

1. Now the frequency ratios for the five partials need to be set. You will create a Macro with a Structure very similar to the one contained in the “Amplitudes” Macro. Because of this, right-click the “Amplitudes” Macro and select the *Duplicate* menu entry.
2. Next, go to the Properties of the second “Amplitudes” Macro and change its Label to “Frequencies”. Your “Frequencies” Macro should now look.



3. Double-click the “Frequencies” Macro to enter its Structure.

4. First, let's change the Label of the second Out Port from “Amp” to “Rtio”. Go to the Properties by double-clicking the “Amp” Out Port and change the Label in the corresponding edit field.
5. Now all that is left to do is to change the range and resolution of *all* the Fader Modules; we will leave the Labels unchanged. The outputs of the Faders serve to set the ratio of the corresponding partial's frequency to the frequency equivalent to the pitch value that is set at the “P” input port of the Sine Bank Module. Double-click the “1” Fader to see its Properties and then make sure you are seeing the Function page.
6. A typical range for the frequency ratio of a partial goes from “1” (the fundamental frequency, equivalent to the pitch at the “P” input) to a value that depends on the number of partials. Obviously, the more partials one has, the higher an upper limit ought to be. To leave you room for a bit of experimentation, set the range to [1 ... 50]. Additionally, a Stepsize value of “0.5” should offer a resolution where you can easily create both harmonic and inharmonic sounds. Make sure that the edit fields in the Range area of your Module's Function page have the same or similar values as to the ones shown below. Remember to do this for *all* the Fader Modules in the “Frequencies” Macro.



7. Last, delete the Multiply Module from the Macro Structure, since we do not need quadratic scaling for the frequencies.
8. Connect the output of the unconnected Merge Module to the “Rtio” output port.

12.2.8 Routing the Frequency Control Parameters to the Sine Bank Module

1. Double-click on the Macro's Structure background to enter the upper level Structure.
2. Again the three output ports of the Macro need to be connected. First, drag a wire from the “Rtio” output port of the “Frequencies” Macro to the “Rtio” input port of the Sine Bank Module.

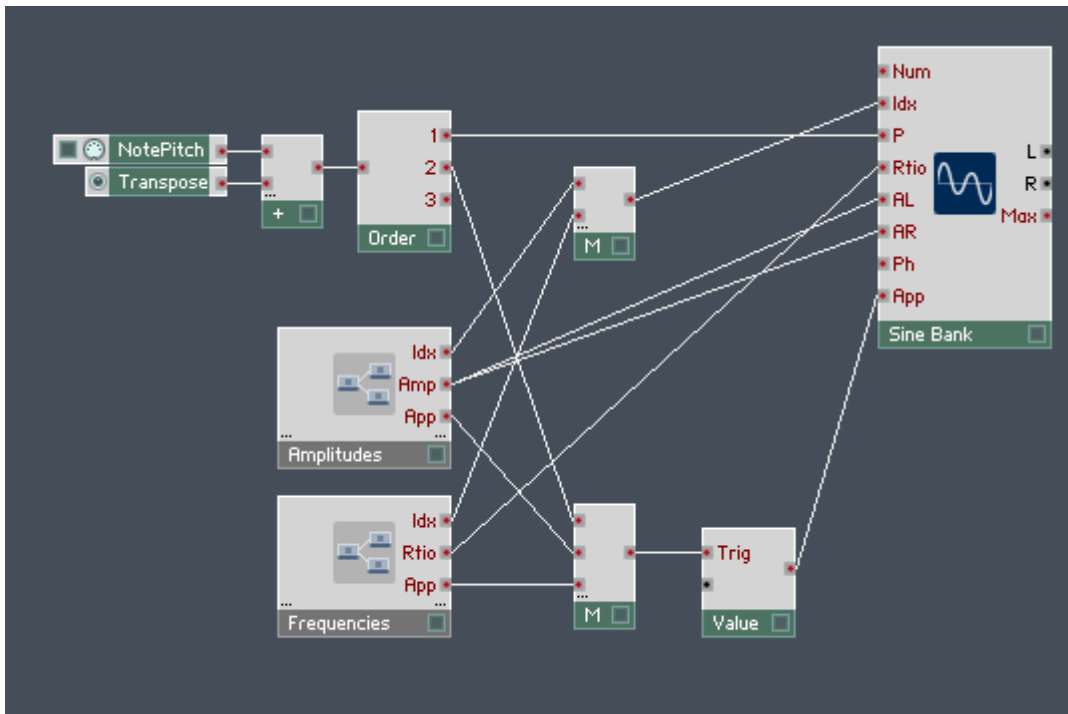
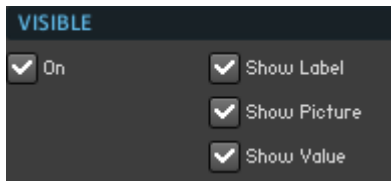


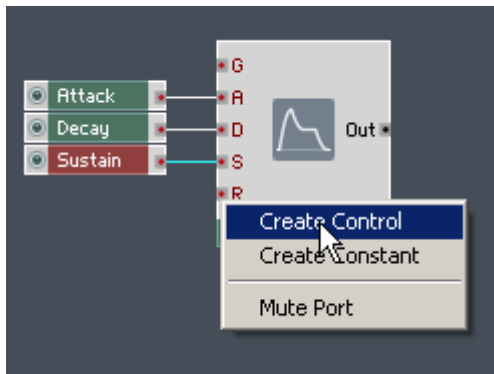
Fig. 12.6 The Structure shown sets the fundamental pitch of the Sine Bank Module and sets the amplitude and frequency ratios of the five partials.

12.2.9 Creating an Amplitude Envelope for the Sine Bank

1. Now that you have your Structure set up to feed the partials of the Sine Bank Module with all the right parameters, you will want to add a simple “ADSR” envelope to the Sine Bank's output signal so that the synth behaves like a real Instrument. Insert an ADSR Module from the *Built-In Module > LFO, Envelope* submenu into your Structure.
2. Since you will probably want to see the envelope graph in the Instrument Panel, go to the View page of the ADSR Module and make sure the *On* and *Show Picture* checkboxes are engaged.



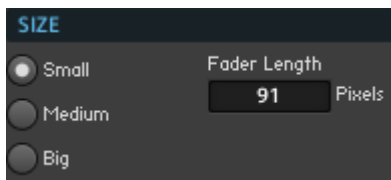
3. To create a control for the envelope attack time, right-click on the "A" input port and choose the *Create Control* menu entry.
4. Apply the same procedure for the "D", "S", and "R" input ports.



5. To make the Panel appearance of the Knob Modules smaller, go to the View page of each Module.

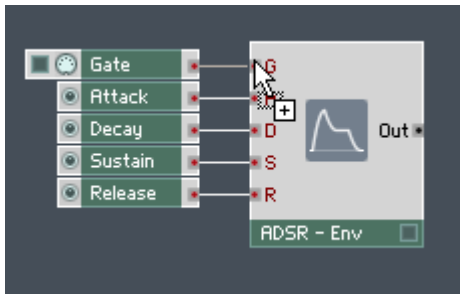


6. Then click on the Small radio button.



7. Now you need a gate signal for the envelope. This should conveniently happen when you press a key on your computer keyboard or MIDI device. For this purpose, insert the Gate Module from the *Built-In Module > MIDI In* submenu into the Structure.

8. Connect the output of the Gate module to the “G” input port of the ADSR Module.



9. To apply the envelope to the Sine Bank Module, we need to multiply its output signal and the envelope signal together. Therefore insert a Multiply Module from the *Built-In Module > Math* submenu into the Structure.
10. Now connect the “L” output port of the Sine Bank Module to one input port, and the output port of the ADSR Module to the other input port of the Multiply Module. Since in the current implementation the left and right stereo channels of the Sine Bank Module are identical, it makes sense to use the output signal from only one output port.
11. Before routing the final signal to the Audio Out Module, you need an Audio Voice Combiner Module (found in the *Built-In Module > Auxiliary* submenu) in your Structure. This Module combines all the Polyphonic Voices into one single Voice. Only Monophonic signals can be sent to the Audio Out Module, because the signal routed to your soundcard has to be one single signal, not several signals running in parallel, as is the case for Polyphonic Voices (see section [↑11.3, Graph Display and Appearance Properties](#) for more information on Polyphony).
12. Connect the output port of the Multiply Module to the input port of an Audio Voice Combiner Module and make sure the output port of that Audio Voice Combiner Module is connected to the input port of the Audio Out Module. Voilà, the Structure of your simple additive synthesizer is now complete; it should look similar to the one shown below.

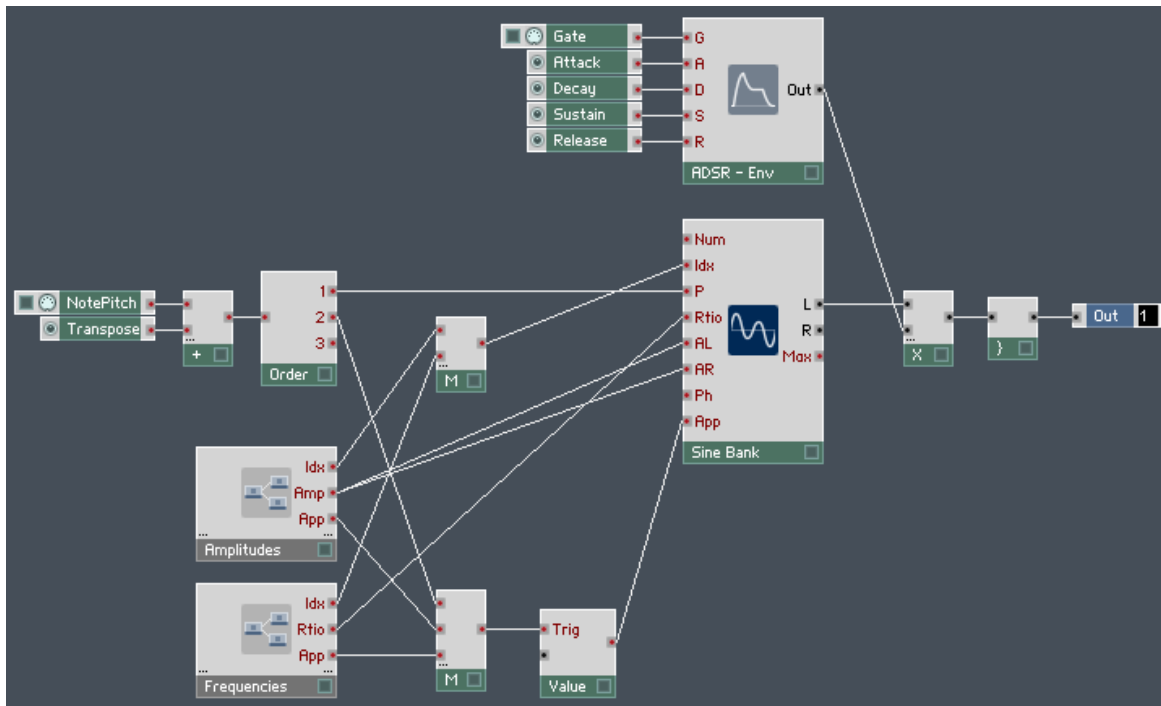


Fig. 12.7 This is the Structure of a simple additive synthesizer featuring five partials and an ADSR envelope. You can set the amplitude and frequency ratios for each partial separately.

12.2.10 Arranging the Instrument Panel of your Additive Synth

1. Although the Structure is now complete, the Instrument Panel still has to be arranged. Go to Panel View, press the Panel Lock button and arrange the Instrument Panel in a way that you find best.



2. Now press the same button again to lock the Panel.
3. Although your layout might be different, your Instrument Panel should look similar to the one shown in the figure below. Now use your keyboard to play this synthesizer! To be sure that the sound that you are hearing from your synth is the result of the fader positions that you see on the Instrument Panel, make sure that you have moved each

fader at least once so that the value has been applied to the corresponding partial. Try out different frequency ratios: only odd harmonic ratios, only even harmonic ratios, inharmonic (non-integer) ratios. A common rule of thumb for partials is that the amplitudes decrease as the frequency ratios increase. But of course, all rules are made to be broken, so feel free to experiment!

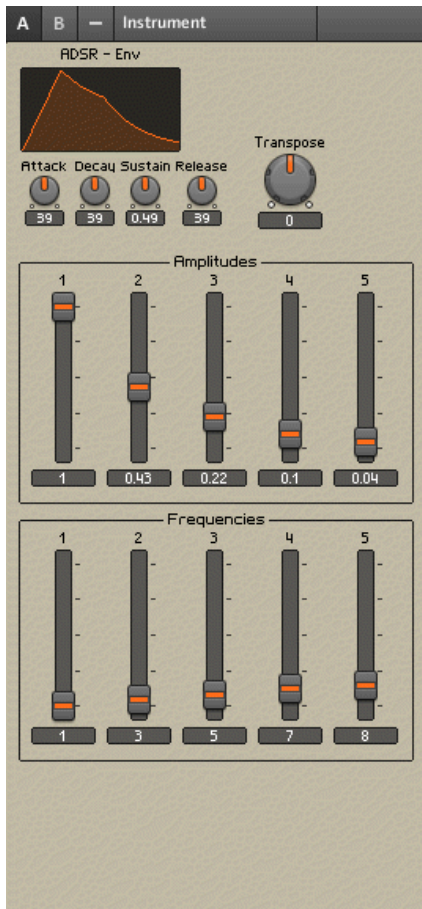


Fig. 12.8 The Instrument Panel of your simple additive synthesizer.

12.3 Building a Simple Structure with the Modal Bank Module

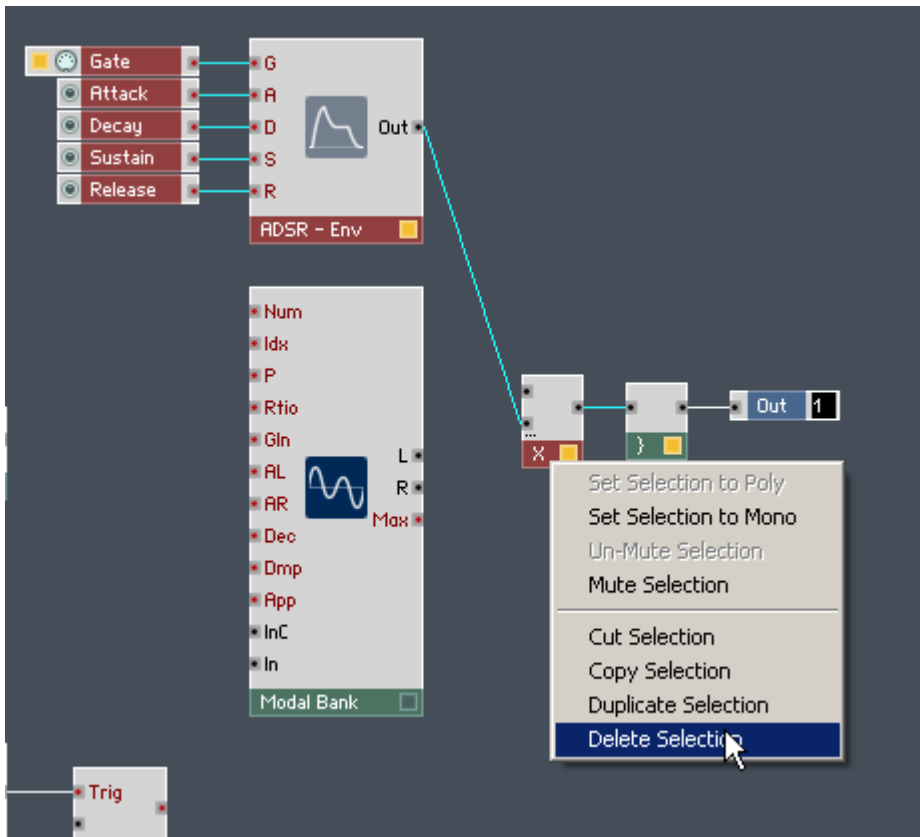
In this tutorial you will learn how to build a very simple modal synthesizer which has 5 partials. For each partial you have three faders controlling the amplitude, frequency, and decay. The principle for feeding the parameters to the partials is identical to that of the Sine Bank Module but the way that the Modal Synthesizer creates sound is much different from the Sine Bank Module. The Modal Bank Module is a set of parallel filters that have a very high resonance setting. Since filter resonance produces a (nearly) sinusoidal waveform, we treat these filters also as “partials”. You have to send an Audio signal into the Modal Bank Module in order to get sound out of it. Sending a short burst of white noise to the Audio input port of the Modal Bank Module causes each of the resonant filters to produce a nearly sinusoidal, exponentially decaying signal at its cutoff frequency. One can control the decay time of these signals at the “Dec” input. Additionally, a damping factor, set at the “Dmp” input, is subtracted from the decay times of all the partials.

12.3.1 Modifying the Additive Synth Structure for the Modal Synth Tutorial

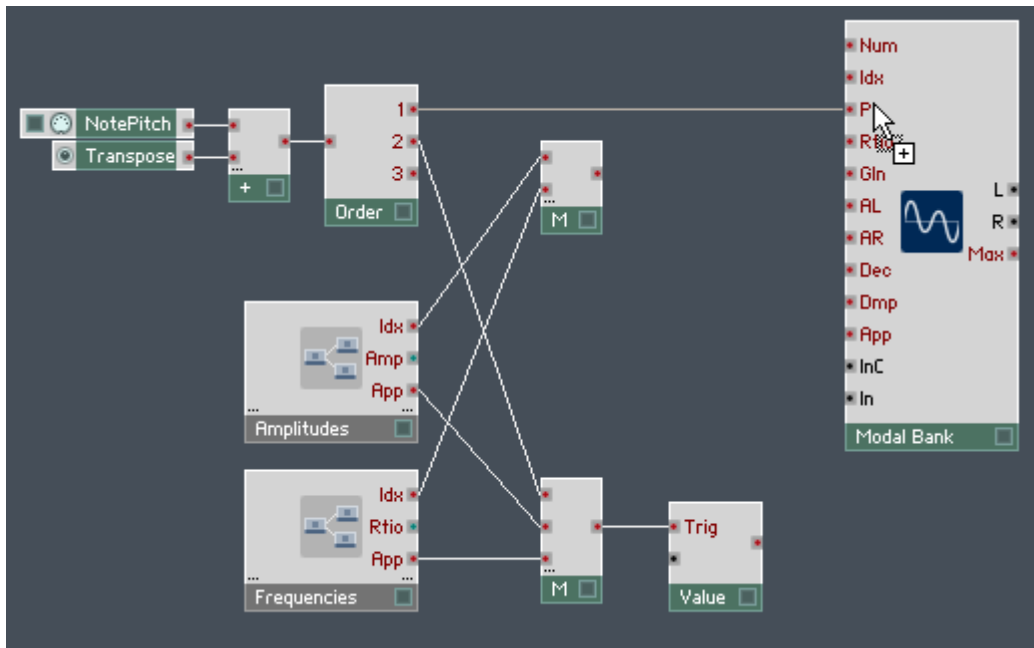
For the following tutorial on building a simple Modal synthesizer it is necessary that you have completed the previous tutorial from subsection [↑12.2, Building a Simple Structure with the Sine Bank Module](#). Perform the following steps:

1. Open the Ensemble that you built in the tutorial in subsection [↑12.2, Building a Simple Structure with the Sine Bank Module](#) and choose the *File > Save Ensemble As ...* menu entry to save a copy of the Ensemble you are about to modify.
2. Go to the Structure View. Your Structure should look like the one from the previous section.
3. First you need to replace the Sine Bank Module with the Modal Bank Module. Right-click on the Sine Bank Module and choose the *Delete* menu entry.
4. Right-click Structure background and choose the *Built-In Module > Filter > Modal Bank* menu entry.
5. Since the partials of the Modal Bank Module decay naturally, there is no need for the ADSR Module in this Structure. Select the ADSR Module and all the Modules that are connected to its input ports, in addition to the Multiply Module.

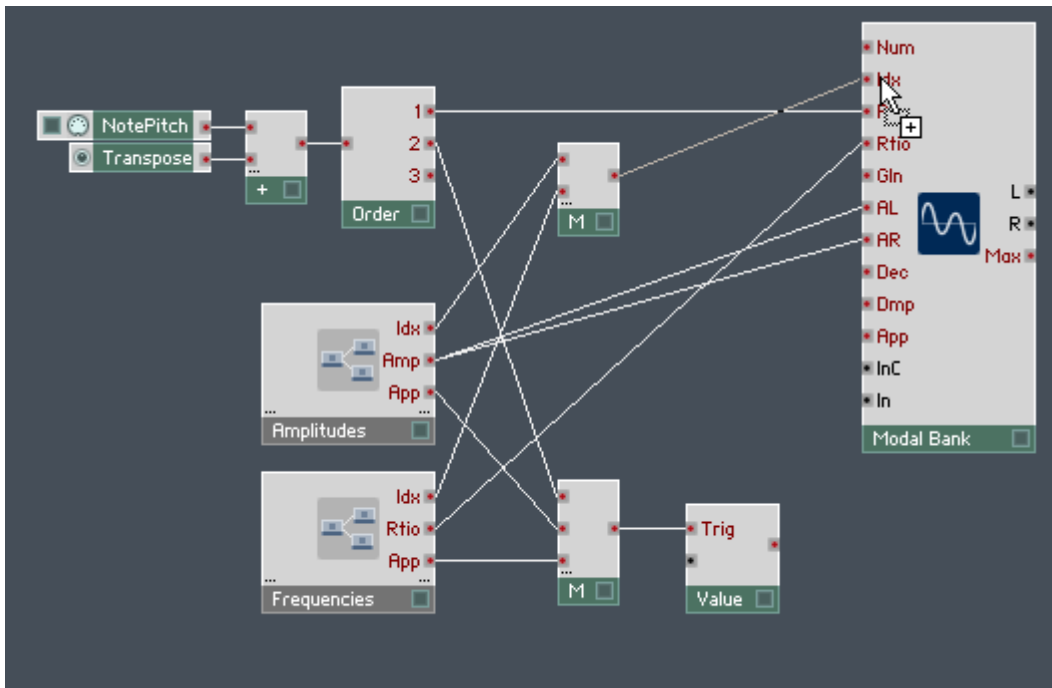
- Next right-click on any one of these Modules and choose the *Delete Selection* menu entry (shown in the figure below).



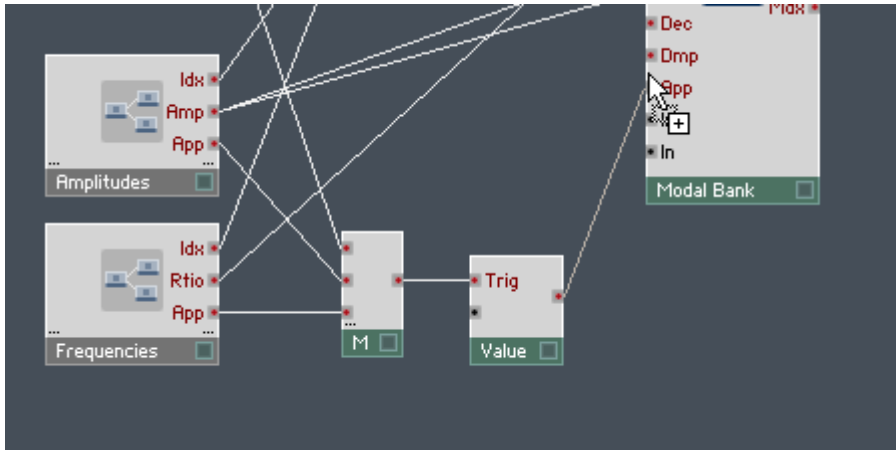
- Now connect the “1” output port of the Order Module (which sends the pitch information) to the “P” input port of the Modal Module.



8. As in with the Sine Bank Module, the “Amp” output port of the “Amplitudes Macro should be connected to the “AL” and “AR” input ports of the Modal Bank Module. Create these connections now.
9. Connect the “Rtio” output port of the “Frequencies” Macro to the “Rtio” input port of the Modal Bank Module as well.
10. Now connect the output of the Merge Module that is connected to the “Idx” output ports to the “Idx” input port of the Modal Bank Module. This is shown in the figure below.

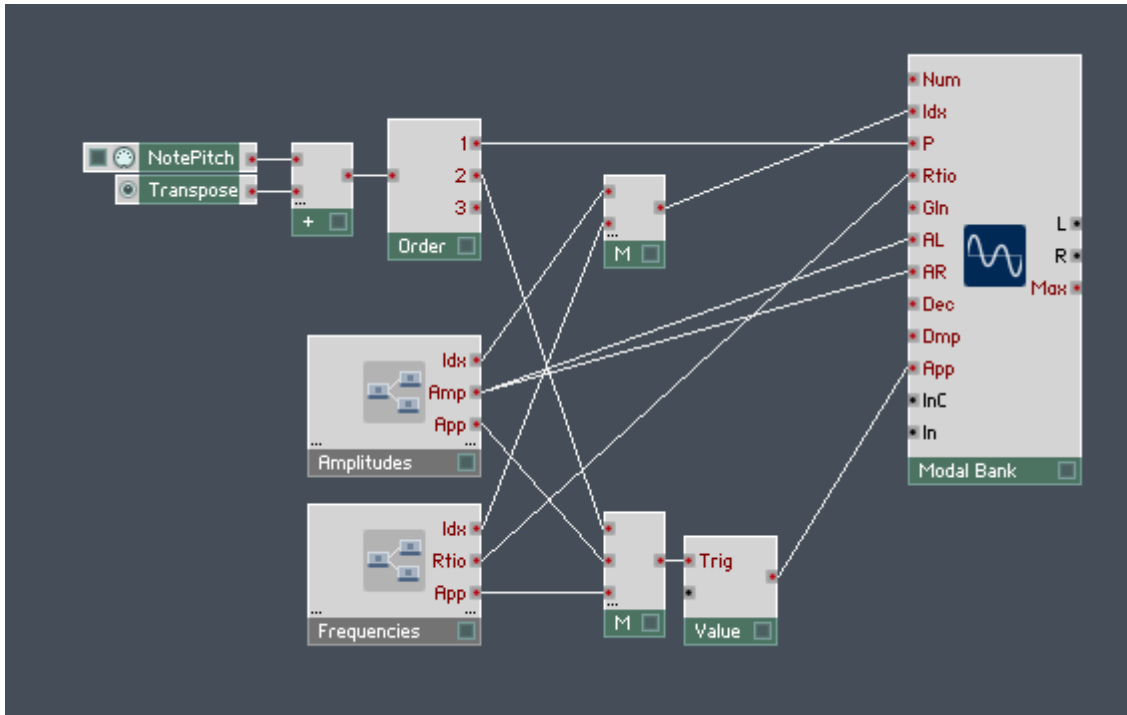


11. Next, connect the output of the Value Module to the “App” input port of the Modal Bank Module, as shown in the figure below.



12.3.2 Creating Decay Time Controls for Partial

1. Once your Structure looks similar to the one depicted in the figure below, you are ready to work on aspects unique to modal synthesis. In this spot we remember that we have the decay parameter which we should specify for the five partials. Again, you need a Macro with a Structure almost identical to the one contained in the “Amplitudes” and “Frequencies” Macros. Because of this, right-click the “Frequencies” Macro and select the *Duplicate* menu entry.

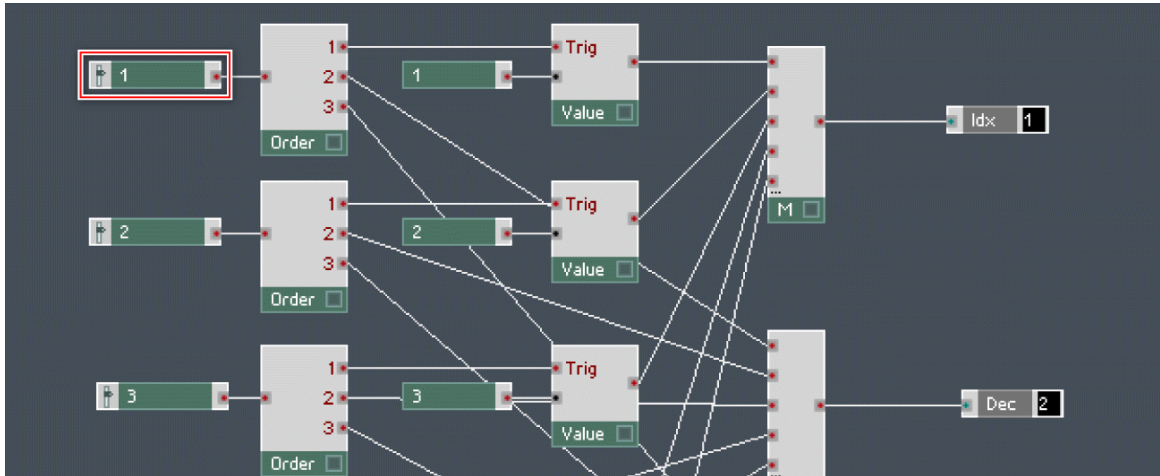


- Next, go to the Properties of the second “Frequencies” Macro and change its Label to “Decays”. Your “Decays” Macro should now look like the one shown below.

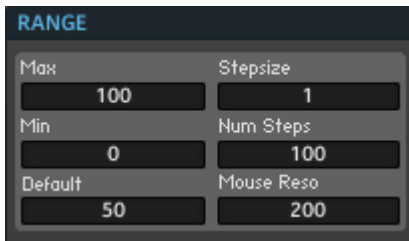


- Double-click the “Decays” Macro to enter its Structure.
- Change the Label of the second Out Port from “Rtio” to “Dec”. Go to the Properties and change the Label in the Label field.

- The outputs of the Faders serve to set the decay parameter of the corresponding partial. You will change the range and resolution to suit the values expected by the “Dec” input port of the Modal Bank Module. To this, double-click the “1” Fader to see its Properties and then make sure you are seeing the Function page.



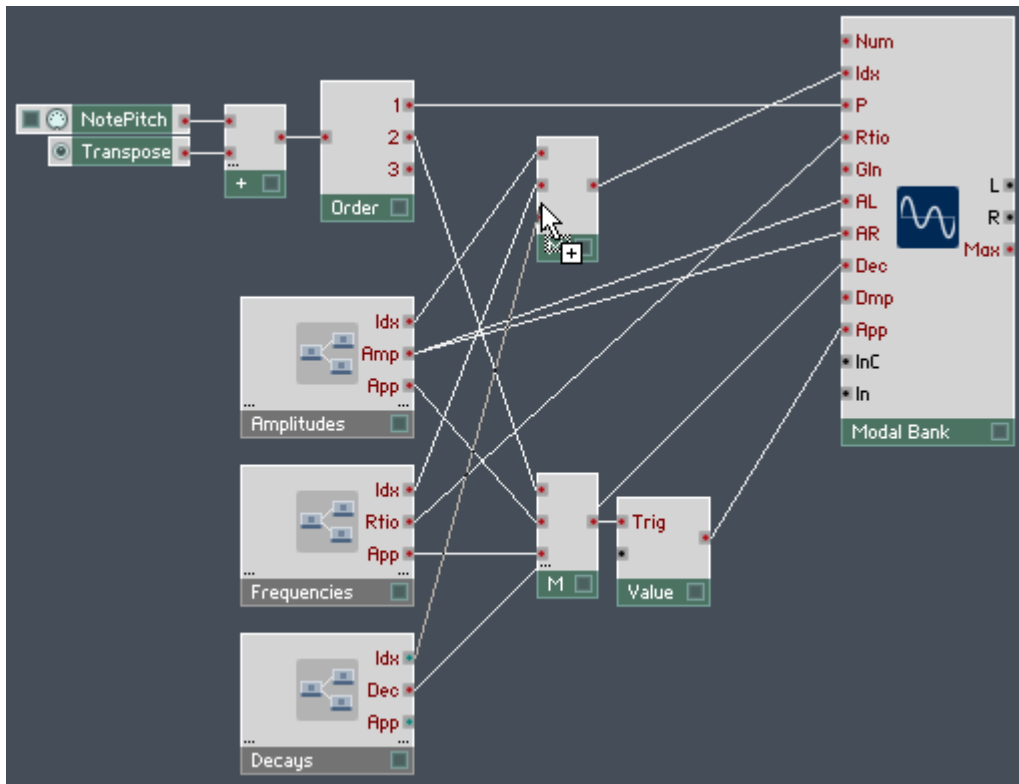
- The decay parameter is the decay specified in the logarithmic scale and in milliseconds, like REAKTOR envelopes. A typical range for the decay time of a partial is [0 ... 100]. Set the Min and Max edit fields to the corresponding values.
- For the resolution, set the Num Steps edit field to “100”.



- Make sure that the edit fields in the Range area of your Module's Function page have the same or similar values as to the ones shown in the figure above. Remember to change the Range settings of *all* of the Fader Modules in that Macro.

12.3.3 Routing the Decay Control Parameters to the Modal Bank Module

1. Double-click on the Macro's Structure background to enter the upper level Structure.
2. Again the three output ports of the Macro need to be connected. First, drag a wire from the “Dec” output port of the “Decays” Macro to the “Dec” input port of the Modal Bank Module.
3. Next, while holding down Ctrl for Windows (or Cmd for Mac OS X), drag a wire from the “Idx” output port of the “Decays” Macro to the input port area of the Merge Module that is connected to the “Idx” input port of the Modal Bank Module. A new input port should appear to accommodate the newly dragged wire.



4. Last, and again holding down Ctrl for Windows (or Cmd for Mac OS X), drag a wire from the “App” output port of the “Decays” Macro to the input port area of the Merge Module that is connected to the Value Module. Again the Module should grow an additional input port to accommodate the new wire. Your resulting Structure should look similar to the one shown below.

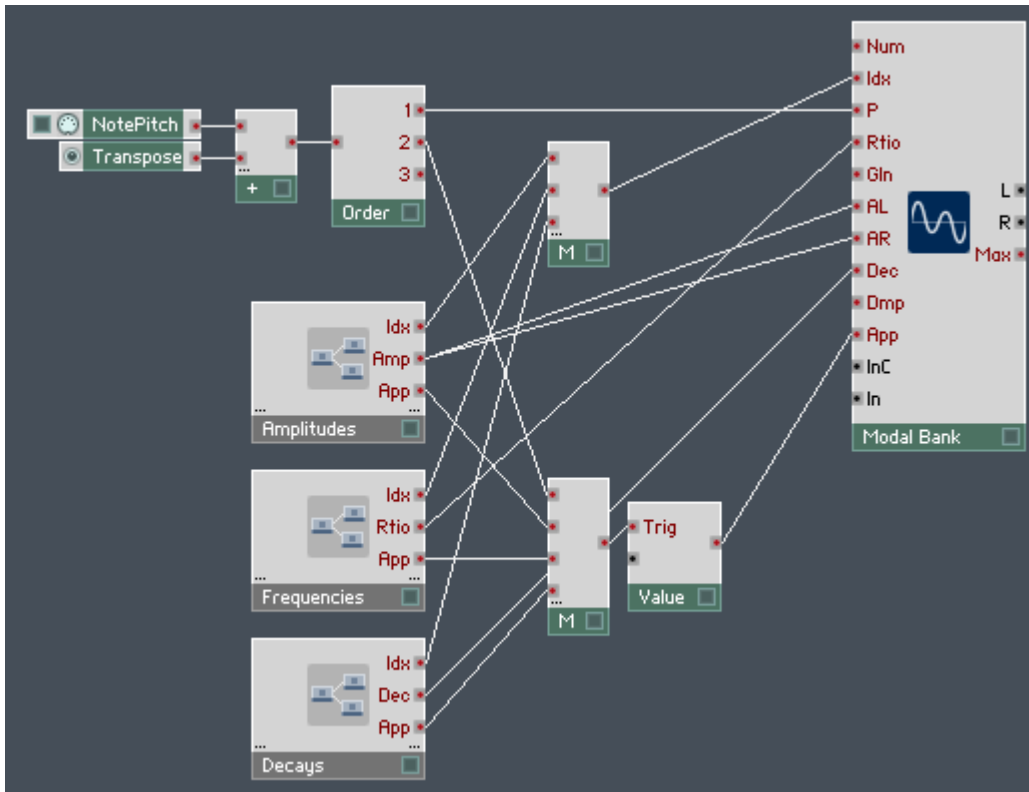
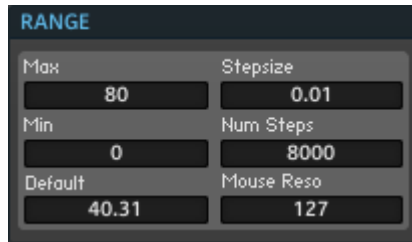


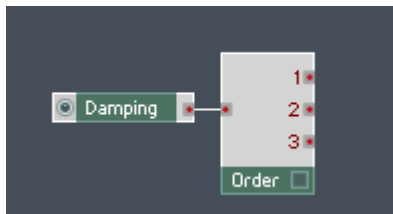
Fig. 12.9 This Structure feeds the Modal Bank Module with the fundamental pitch, partial amplitude, partial frequency, and partial decay rate information.

12.3.4 Sending the Damping Parameter to the Modal Bank Module

1. You already supply your Modal Bank Module's partials with individual decay information. The Modal Bank has also the damping parameter which is globally subtracted from the decay values of all partials to reduce the decay times. This parameter is specified at the “Dmp” input port and you will now create a control for it. Right-click the “Dmp” input port and choose the *Create Control* menu entry. This creates a Knob with the correct Label “Damping” and with a suitable resolution and range, shown below.



2. Just having the Knob connected to the “Dmp” input port is not enough, though. After moving the Knob, the new damping value also has to be applied. So disconnect the “Damping” Knob from the “Dmp” input port. By now you should be familiar with the procedure that follows.
3. Insert an Order Module from the *Built-In Module > Event Processing* submenu.
4. Connect the output of the “Damping” Knob Module to the input of the Order Module you just inserted into your Structure.



5. Connect the “1” output port of that Order Module to the “Dmp” input port of the Modal Bank Module.

6. Last, while holding down Ctrl for Windows (or Cmd for Mac OS X), drag a wire from the “2” output port of the Order Module to the input port area of the Merge Module that is connected to the Value Module. The Merge Module should grow an additional input port to accommodate the wire. Your resulting Structure should look similar to the one shown in the figure below.

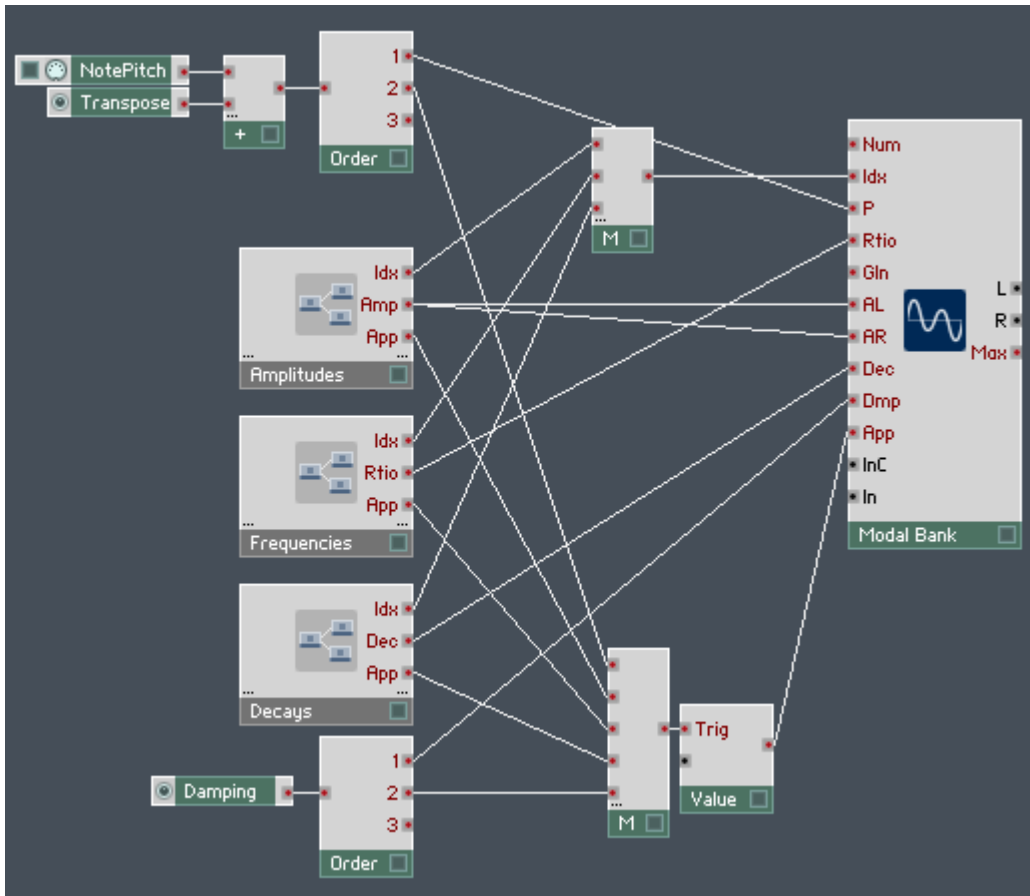
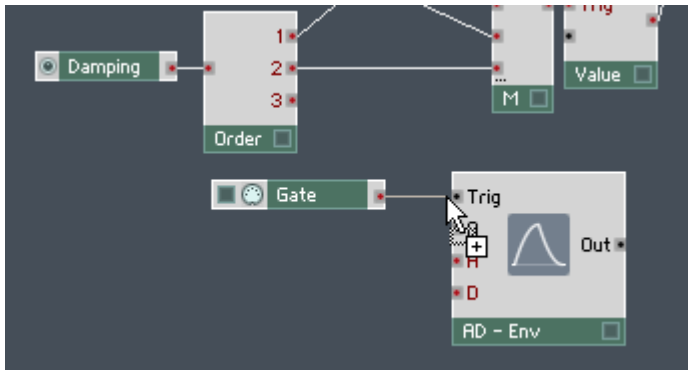


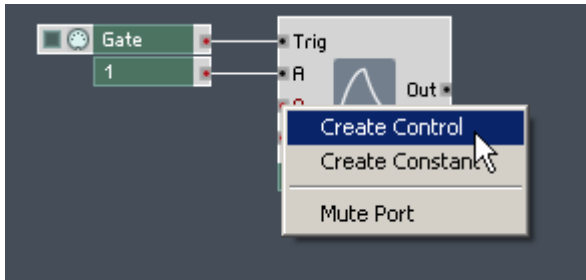
Fig. 12.10 Sending the “Dmp” parameter to the Modal Bank should be followed by an “apply” Event. For this functionality, a signal flow using the Order Module is implemented.

12.3.5 Using a Short Impulse as an Exciter Signal

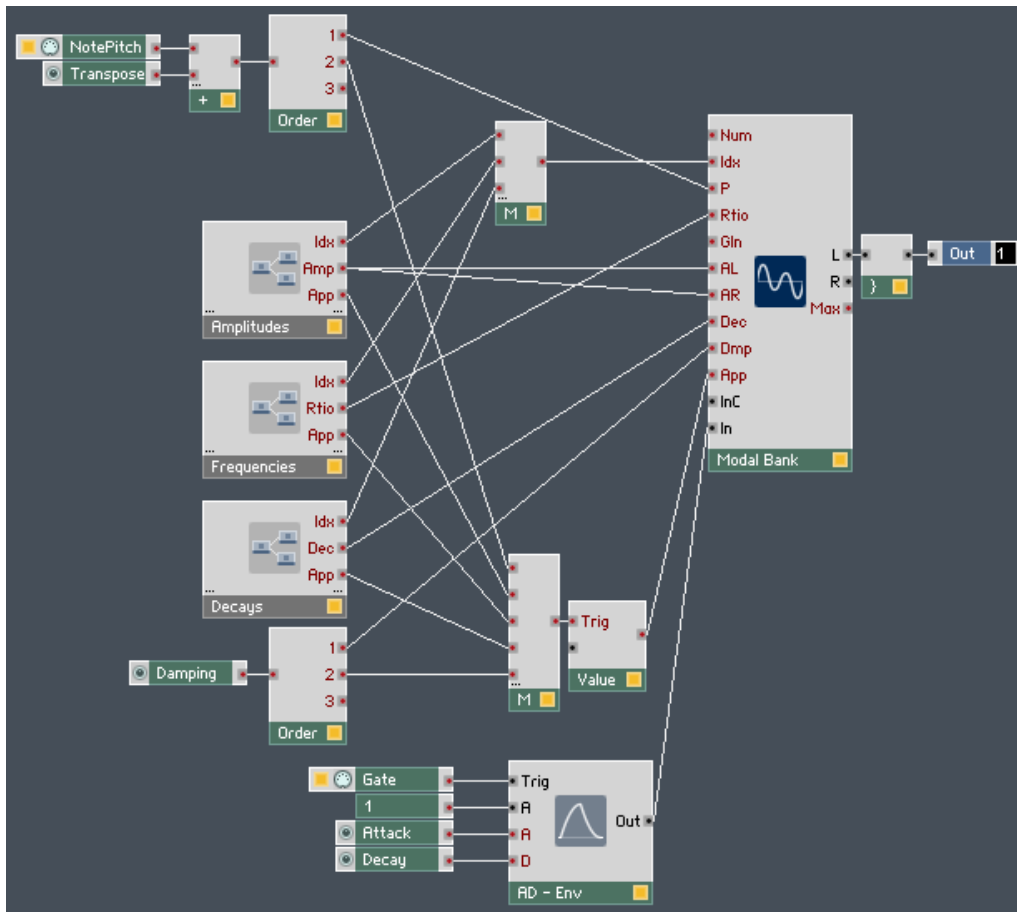
1. You have hooked up all the necessary parameters to your Modal Bank Module. The only thing that is missing now is the sound. Since it's also the main reason for building this Structure, we will move on and get some sound out of that Module! In this tutorial you will see a couple of different ways to feed an Audio signal to the Modal Bank. The first and simplest is to send an impulse to the “In” input port. You will create the impulse using the output signal of an AD envelope. For that, insert an AD Module from the *Built-In Module > LFO, Envelope* submenu into your Structure.
2. Now you need a trigger signal for the envelope that is sent when you press a key on your computer keyboard or MIDI keyboard. For this purpose, insert the Gate Module from the *Built-In Module > MIDI In* submenu into the Structure.
3. Connect the output of the Gate Module to the “Trig” input port of the AD Module.



4. Let's set the maximum amplitude of the AD Envelope to “1”. Right-click the “A” Audio input port of the AD Module and select the *Create Constant* menu entry. Make sure that the Constant Module has the value “1”.
5. Next, right-click the “A” Event input port and choose the *Create Control* menu entry.
6. Repeat the previous step for the “D” Event input port. Now you will have controls on the Instrument Panel for the attack and decay times of the AD envelope.



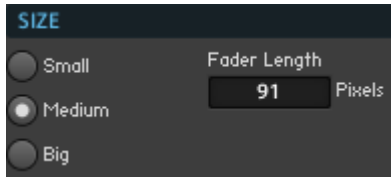
7. Next, connect the output of the AD Module to the “In” input port of the Modal Bank Module.
8. At this intermediate stage it would be nice to hear what effect an impulse signal has on the sound of a Modal Bank Module. In the current implementation the left and right stereo channels of the Modal Bank Module are identical. Therefore it makes sense to use the Audio signal from only one stereo channel to hear the result. Connect the “L” output port of the Modal Bank Module to the input port of an Audio Voice Combiner Module and the output port of that Audio Voice Combiner Module to the input port of the Audio Out Module. Your Structure should now look similar to the one shown below.



9. You have inserted three new Knob Modules into the Structure. You might or might not want these Knobs to have large Panel representations. To reduce the size of the Knobs a bit, go to their View page.

VIEW

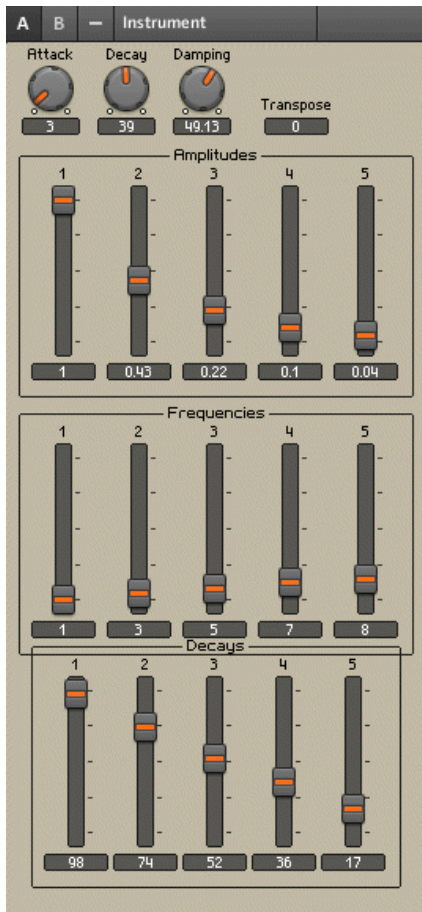
10. Then select the **Medium** radio button. Do this for all Knobs, if preferred—it will help you keep your Instrument Panel tidier.



11. Now go to the Panel View and click the Panel Lock button.



12. Move your controls and Macros around to create a layout that suits you best, but leave room for two more Knobs that we will add in the steps below.
13. Lock the Panel using the same button.

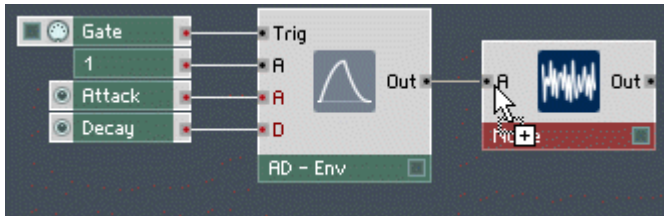


14. Now play around with the Instrument to check out these intermediate results! To be sure that the sound that you are hearing from your synth is the result of the fader positions that you see on the Instrument Panel, make sure that you have moved each fader at least once so that the value has been applied to the corresponding partial. Note how the characteristic sound of the Modal Bank Module differs from that of the Sine Bank Module. Adjust the attack and decay times of the impulse and, of course, try out different frequency ratios: only odd harmonic ratios, only even harmonic ratios,

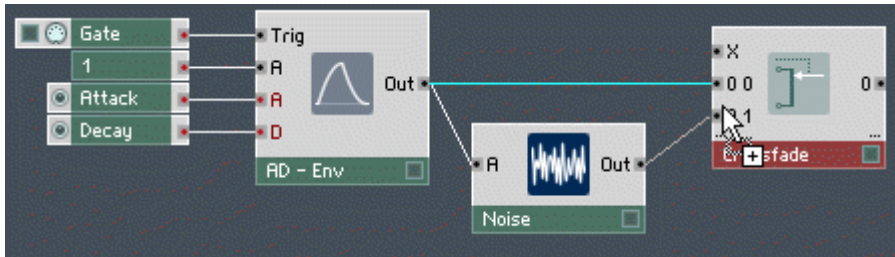
and inharmonic (non-integer) ratios. A common rule of thumb for the partials of percussive sounds made is that the decay times should decrease as the frequency ratios increase.

12.3.6 Using White Noise as an Exciter Signal

1. After hearing the sound generated by the Modal Bank Module in response to an impulse, let's explore other possibilities for the input signal. An effective input for the Modal Module is white noise. White noise has a spectrum that is equally distributed over all frequencies, this means that all partials (filters) of the Modal Bank are excited equally, independent of their frequency. To try out the effect of a short burst of noise at the input of the Modal Module, insert a Noise Module from the *Built-In Module > Oscillator* submenu into your Structure.
2. It is convenient to apply the same AD envelope that you have already in the Structure for the to the white noise signal. Therefore drag a wire from the output port of the AD Module to the “A” input port of the Noise Module.



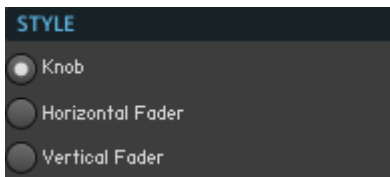
3. Also, crossfading between having the impulse and the noise signal at the input of the Modal Bank Module would be useful for comparing between the two sounds. Insert a Crossfade Module from the *Built-In Module > Signal Path* submenu into the Structure.
4. Connect the output of the AD Module to the “O 0” input port of the Crossfade Module.
5. Next, connect the output of the Noise Module to the “O 1” input port of the Crossfade module.



6. Now right-click the “X” input port of the Crossfade Module and choose the *Create Control* menu entry.
7. Double-click the newly created fader to view its Properties and rename its Label to “Imp - Nse”. Your Structure should now look like the one shown at the end of this list.
8. To reduce change the fader into a knob, go to its View page.



9. Select the *Knob* radio button in the Style area (shown in the figure below).



10. Then select the *Medium* radio button to reduce the Size of the knob.
11. Last, connect the output port of the Crossfade Module to the “In” input port of the Modal Bank Module.
12. Go to the Panel View and using the *Panel Lock* button, find a good location for the newly created Knob Module.



13. Lock the Panel using the same button.
14. Now play the new version of your Modal synthesizer. Actively compare the sounds of the impulse and the noise input but turning the “Imp - Nse” knob while playing notes.

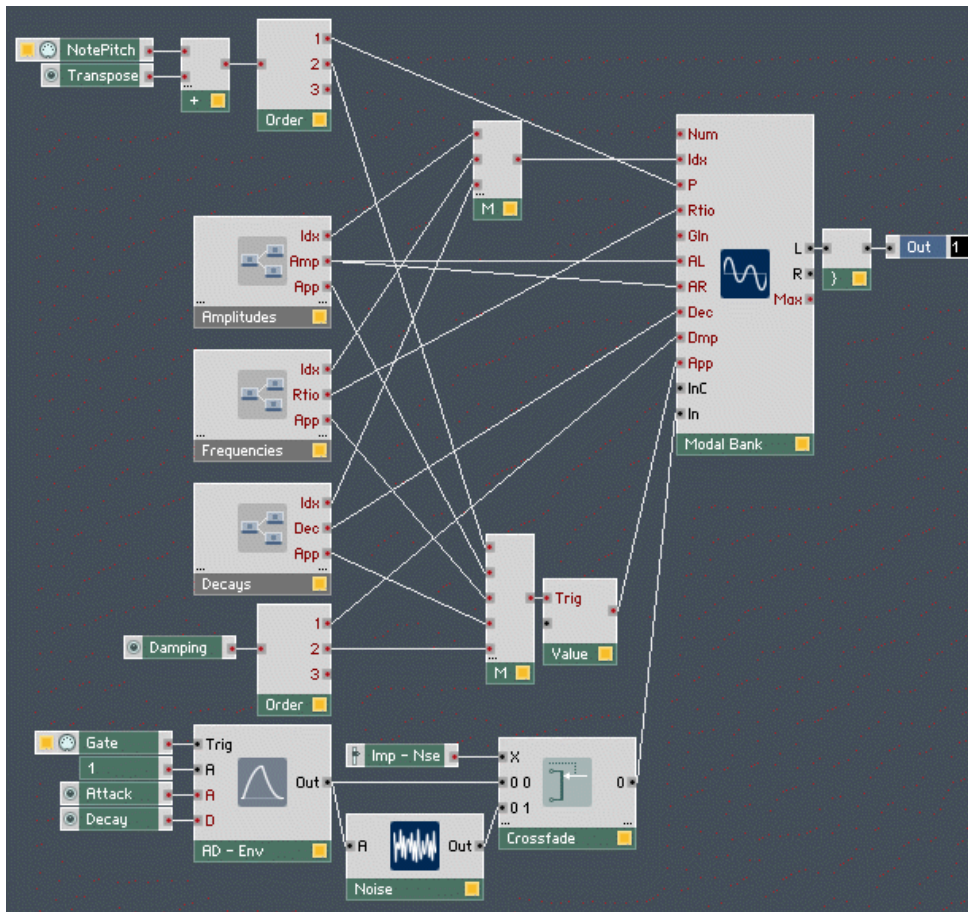


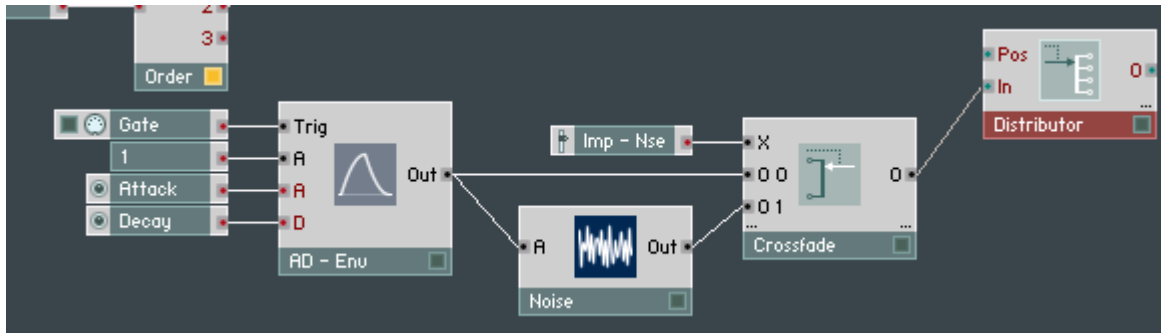
Fig. 12.11 In this version of the modal synthesizer two common input signals are used: an impulse and a burst of white noise.

12.3.7 Using the Compensated Input Port for Sustained Exciter Signals

1. Until now you have used the uncompensated Audio input port, “In”, for the input signal. This input is usually used for impulse-like exciter signals. The “InP” input port is an amplitude-compensated input for the exciter signal, typically used for sustained signals that would be too loud when passed through partial filters with low decay

times (resonance). Let's distribute the signal coming from the output of the Crossfade Module between the “In” and the “InP” input ports. For this, insert a Distributor Module from the *Built-In Module > Signal Path* submenu into your Structure.

2. Connect the output port of the Crossfade Module to the “In” input port of the Distributor Module.



3. Right-click on the “Pos” input port of the Distributor Module and choose the *Create Control* menu entry.
4. Double-click the newly created Fader to view its Properties and rename its *Label* to “Compensate”.
5. To change the fader to a knob, go to its *View* page.

VIEW

6. Select the *Knob* radio button in the *Style* area.
7. Then, to reduce the *Size* of the knob, select the *Medium* radio button.
8. Now set the minimum output ports for the Distributor Module to be two. This is done in the *Function* page of the Distributor Module, so click on the Distributor Module and then *Function* button.

FUNCTION

9. Double-click the *Min Num Port Groups* edit field and type in the desired value: “2”. This is shown in the figure below.

13. Using the **Panel Lock** button, find a good location for the Knob Module you just created and lock the Panel again using the same button.



14. The Instrument Panel should look similar to the one at the end of this list.
Congratulations, you have completed your first modal synthesizer! Enjoy playing the Instrument and remember to compare the sound of the compensated and non-compensated inputs.

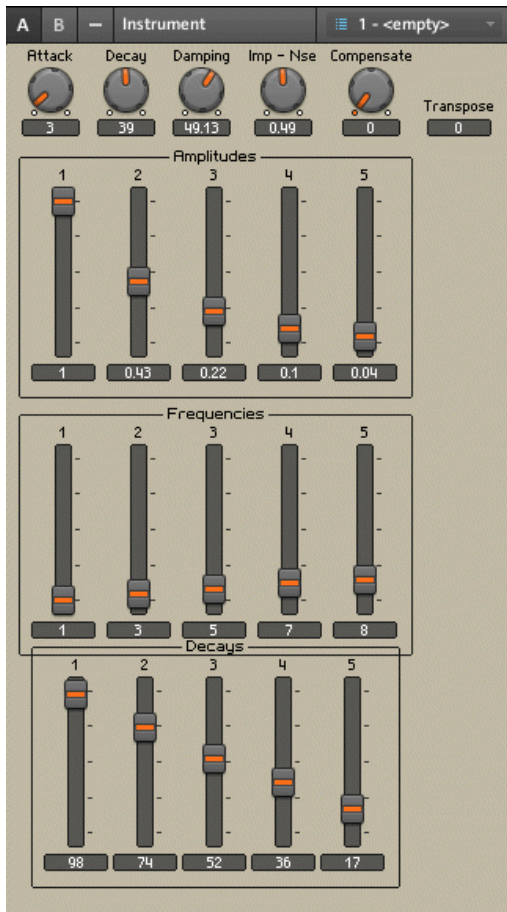


Fig. 12.12 The Instrument Panel of the finished modal synthesizer with five partials.

12.4 The Sine Bank Module

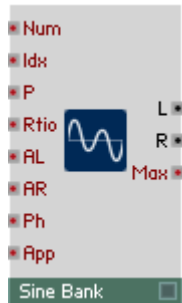


Fig. 12.13 The Sine Bank Module

12.4.1 Overview

The Sine Bank Module is a bank of parallel sine oscillators, called partials. This Module enables you to perform additive synthesis with REAKTOR. For each partial, its frequency ratio to the fundamental, its amplitude, and its phase can be specified at the “Rtio”, “AL” and “AR”, and “Ph” input ports, respectively. Setting these parameters for each partial is done serially and with the use of the “Idx” input port to specify to which partial the incoming parameters should be sent. The fundamental pitch of the Sine Bank is specified at the “P” input; it corresponds to the frequency in respect to which the frequencies of all the partials are calculated. Also, the number of partials that the Sine Bank Module generates has to be specified, either in the Function page of the Module's Properties or at the “Num” input port. After sending all the parameters to the desired partials, an “apply” Event has to be sent to the “App” input port to effectively apply these changes to the partials. The resulting stereo Audio signal is sent to the “L” and “R” output ports for the left and right stereo channels, respectively. Also, in the Function page of the Module's Properties, one can specify the smoothing time and smoothing quality. The smoothing quality setting has an effect on the Module's CPU usage.

The default parameters for the partials are such that the “L” output sends a pulse wave with the pitch of the fundamental (set at the “P” input port) and the “R” output sends a saw wave with the pitch one octave above the fundamental. By adding both signals together you get a saw wave at the pitch of the fundamental.

Application

The Sine Bank Module is meant for additive synthesis. In theory, it is possible to synthesize anything using additive synthesis. In practice, this is often not possible because of the sheer amount of data that would have to be sent to the partials of the Sine Bank Module. Nevertheless, with the right parameters, the synthesis of a very broad range of sounds becomes available. Often a large number of partials are used; the maximum number of partials is 10'000. Usually much less partials than this maximum number are used, mainly because of CPU limitations. In a case where there are a lot of partials it is most effective to make use of the Iterator Module (found in the *Built-In Module > Event Processing* submenu) to supply all partials with the necessary parameters between the processing of subsequent audio samples.

12.4.2 Ports

Input Ports

- **(Num)** “Num” (number of partials) is the Event input port for setting the number of active partials in the Sine Bank Module. This value should be a positive integer in the range [0 ... Max] where “Max” corresponds to the number set in the *Max. Number* edit field in the Function page of the Module's Properties. If this input port is left disconnected, the Sine Bank Module automatically sets the number of partials to the value to “Max”. A higher number of active partials requires more CPU resources. Often the CPU load is increased for multiples of 4 partials.
- **(Idx)** “Idx” (index) is the Event input port for specifying the partial number to which the subsequent parameters arriving at the “Rtio”, “AL”, “AR”, and “Ph” input ports should be routed. This value should be a positive integer in the range [1 ... Num] where “Num” denotes the number of active partials (see “Num” input port).

- **(P)** “P” (pitch) is the Event input port for controlling the fundamental pitch of the Sine Bank Module. The frequency corresponding to the fundamental pitch is the frequency in respect to which the frequencies of all the partials are calculated. The values at this input port are set in the logarithmic pitch scale and are usually in the range [0 ... 127].
- **(Rtio)** “Rtio” (frequency ratio) is the Event input port for setting the ratio of the partial's frequency to the fundamental frequency determined at the “P” input port. If the “Rtio” value of a partial is “1”, it means that the ratio between the partial's frequency and the fundamental frequency is “1”, that is, the partial has the fundamental frequency. If the “Rtio” value of a partial is “2”, then the partial has double the frequency of the fundamental. In general, the frequency of the partial (“f”) can be calculated from the fundamental frequency (“F”) and the frequency ratio (“Rtio”) with the following formula: $f = F * \text{Rtio}$. As can be seen from the formula, negative “Rtio” values cause negative frequencies for the partials. Negative frequencies cause the phase of the oscillator to run backwards.
- **(AL)** “AL” (amplitude, left channel) is the Event input port for specifying the partial amplitude in the left stereo channel. It is the value that the unit amplitude oscillator waveform of the partial is multiplied by before being mixed together with the other partials and forwarded to the “L” output port. The typical range is [0 ... 1], but negative values are also allowed; they cause the oscillator waveform of the corresponding partial to be inverted.
- **(AR)** “AR” (amplitude, right channel) is the Event input port for specifying the partial amplitude in the right stereo channel. It is the value that the unit amplitude oscillator waveform of the partial is multiplied by before being mixed together with the other partials and forwarded to the “R” output port. The typical range is [0 ... 1], but negative values are also allowed; they cause the oscillator waveform of the corresponding partial to be inverted.
- **(Ph)** “Ph” (phase) is the Event input port for specifying the phase (position in the waveform) to which the oscillator is reset when a positive “apply” Event arrives at the “App” input port. The range for the “Ph” input port is [-1 ... 1] where “-1” corresponds to a phase of -180 degrees and “1” corresponds to a phase 180 degrees.

- **(App)** “App” (apply) is the Event input port for applying the latest parameter values that have arrived at the various input ports to the actual oscillators inside the Sine Bank Module. The parameter values sent to the Sine Bank Module are stored in a temporary buffer and only applied to the partials once an Event arrives at the “App” input port. If the value of the Event is greater than zero, the apply operation also applies the latest values that have been sent to the “Ph” input port and forces the smoothers to jump to the values of the corresponding partial amplitude parameters. Thus, a positive valued Event at the “App” input port works like a “reset”. If the “apply” Event is less or equal to zero, all new parameters except the phase are applied.

Output Ports

- **(L)** “L” (left stereo channel) is the output port for the Audio signal that is a mix of all the partials with amplitudes set by the “AL” parameter.
- **(R)** “R” (right stereo channel) is the output port for the Audio signal that is a mix of all the partials with amplitudes set by the “AR” parameter.
- **(Max)** “Max” (maximum number of partials) is the Event output port for the value set in the Max. Number edit field of the Module's Function page.

12.4.3 Properties: Function Page

Setting Module to Mono

To set the Polyphony of the Module to the value “1” (Monophonic), engage the **Mono** checkbox, shown below. Please refer to section [↑9.3, Polyphony](#) for more information on Polyphony.



Fig. 12.14 The Status area in the Module's Function page holds the checkboxes to set the Module to Monophonic Mode and to activate or deactivate the Module.

Muting the Module

To mute this Module, engage the **Mute** checkbox, as shown in the previous paragraph. This also mutes all Modules that are lying upstream of it and are not connected to any other Modules.

Setting the Maximum Number of Partials

The maximum number of partials that the Sine Bank Module generates has to be specified in the **Max. Number** edit field in the Module's Function page, shown in the figure below. The actual number of active partials is specified at the “Num” input port. “Num” values larger than the value set in the **Max. Number** edit field are clipped. If the “Num” input port is left disconnected, the Sine Bank Module automatically sets the number of partials to the value specified by the **Max. Number** edit field. The default value for this edit field is “128”. Also, the value in the **Max. Number** edit field is sent to the “Max” output port. A higher number of active partials requires more CPU resources.



Due to internal optimization, for many computers the CPU load is increased when the number of partials reaches a multiple of 4 partials.

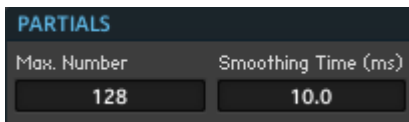


Fig. 12.15 The maximum number of partials and the smoothing time are set in the Partials area of the Sine Bank Module's Function page.

Setting the Smoothing Time

The smoothing time defines the time-interval over which the partial amplitudes are faded between different values. This parameter can be set in the **Smoothing Time** edit field. The units for this edit field are milliseconds and the default value is 10 ms.

Setting the Smoothing Quality

You can specify the quality of the smoothing curves with the **Smoothing Quality** drop-down menu (see screenshot below). Higher smoothing quality causes the Module to respond better to fast modulations of the partial amplitudes. However, increasing the smoothing quality also increases the CPU load.

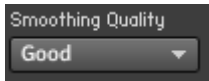


Fig. 12.16 The Smoothing Quality drop-down menu lets you increase the quality of the Module's sound in response to fast modulations of the partial amplitudes.

12.5 The Modal Module



Fig. 12.17 The Modal Bank Module

12.5.1 Overview

The Modal Bank Module is a bank of parallel resonant filters, called partials. This Module enables you to perform modal synthesis with REAKTOR. For each partial, its frequency ratio to the fundamental, its input and output amplitudes, and its decay time can be specified at the “Rtio”, “Gln”, “AL” and “AR”, and “Dec” input ports, respectively. Setting these parameters for each partial is done serially and with the use of the “Idx” input port to specify to which partial the incoming parameters should be sent. The fundamental pitch of the Modal Bank is specified at the “P” input; it corresponds to the frequency in respect to which the frequencies of all the partials are calculated. Also, the number of partials that the Modal Bank Module generates has to be specified, either in the Function page of the Module's Properties or at the “Num” input port. The “Dmp” input port is used for the

damping value which represents a global value that is subtracted from the decay times of all partials. After sending all the parameters to the corresponding input ports, an “apply” Event has to be sent to the “App” input port to effectively apply these changes to the partials. Before the exciter signal reaches a partial, it is multiplied by the “Gln” factor specified at that partials index. After the exciter signal has passed through the filter of the corresponding partial, the resulting stereo Audio signal is sent to the “L” and “R” output ports for the left and right stereo channels, respectively. Also, in the Function page of the Module's Properties, one can specify the smoothing time and smoothing quality. The smoothing quality setting has an effect on the Module's CPU usage.

The default parameters for the partials are such that the “L” output sends a pulse wave with the pitch of the fundamental (set at the “P” input port) and the “R” output sends a saw wave with the pitch one octave above the fundamental. By adding both signals together you get a saw wave at the pitch of the fundamental.

Application

The Modal Bank Module is meant for modal synthesis which is a synthesis method that basically consists of an exciter signal and a resonator. This system can be great for synthesizing percussive sounds ranging from mallets to vibraphones, but also for sustain sounds like woodwinds. The Modal Bank Module can also be used in a feedback circuit, extending its sonic possibilities even further. As with the Sine Bank Module, some sounds require a lot of data to be sent to the partials of the Modal Bank Module. However, this is not the case for all sounds. The strength of the Modal Module lies in the fact that depending on the exciter signal, it is possible to create complex output signals with limited use of real-time modulation of the partial parameters. Often a large number of partials are used; the maximum number of partials is 10'000. Usually much less partials than this maximum number are used, mainly because of CPU limitations. In a case where there are a lot of partials it is most effective to make use of the Iterator Module (found in the *Built-In Module > Event Processing* submenu) to supply all partials with the necessary parameters between the processing of subsequent audio samples.

12.5.2 Ports

Input Ports

- **(Num)** “Num” (number of partials) is the Event input port for setting the number of active partials in the Sine Bank Module. This value should be a positive integer in the range [0 ... Max] where “Max” corresponds to the number set in the `Max. Number` edit field in the Function page of the Module's Properties. If this input port is left disconnected, the Sine Bank Module automatically sets the number of partials to the value to “Max”. A higher number of active partials requires more CPU resources. Often the CPU load is increased for multiples of 4 partials.
- **(Idx)** “Idx” (index) is the Event input port for specifying the partial number to which the subsequent parameters arriving at the “Rtio”, “AL”, “AR”, and “Ph” input ports should be routed. This value should be a positive integer in the range [1 ... Num] where “Num” denotes the number of active partials (see “Num” input port).
- **(P)** “P” (pitch) is the Event input port for controlling the fundamental pitch of the Modal Bank Module. The frequency corresponding to the fundamental pitch is the frequency in respect to which the frequencies of all the partials are calculated. The values at this input port are set in the logarithmic pitch scale and are usually in the range [0 ... 127].
- **(Rtio)** “Rtio” (frequency ratio) is the Event input port for setting the ratio of the partial's frequency to the fundamental frequency determined at the “P” input port. If the “Rtio” value of a partial is “1”, it means that the ratio between the partial's frequency and the fundamental frequency is “1”, that is, the partial has the fundamental frequency. If the “Rtio” value of a partial is “2”, then the partial has double the frequency of the fundamental. In general, the frequency of the partial (“f”) can be calculated from the fundamental frequency (“F”) and the frequency ratio (“Rtio”) with the following formula: $f = F * \text{Rtio}$.
- **(Gln)** “Gln” (gain in) is the Event input port for specifying the gain factor with which the exciter signal (at the “In” or “InP” input ports) is multiplied before it is sent to the corresponding partial's filter. This means that each partial has its own “Gln” value associated with it. Using “Gln” values you can consecutively excite different partials without suppressing the output signals of the already excited partials.

- **(AL)** “AL” (amplitude, left channel) is the Event input port for specifying the partial amplitude in the left stereo channel. It is the value that the unit amplitude oscillator waveform of the partial is multiplied by before being mixed together with the other partials and forwarded to the “L” output port. The typical range is [0 ... 1], but negative values are also allowed; they cause the waveform of the corresponding partial to be inverted.
- **(AR)** “AR” (amplitude, right channel) is the Event input port for specifying the partial amplitude in the right stereo channel. It is the value that the unit amplitude oscillator waveform of the partial is multiplied by before being mixed together with the other partials and forwarded to the “R” output port. The typical range is [0 ... 1], but negative values are also allowed; they cause the waveform of the corresponding partial to be inverted.
- **(Dec)** “Dec” (decay) is the Event input port for specifying the decay time (resonance) of the filter associated with the addressed partial. The decay time is specified in the logarithmic scale, just like the envelope time inputs of the envelope Modules in the Primary Level. It is the time (in milliseconds) that the decay curve requires to reach the factor “1 / e” in respect to its initial (maximum) value. Here “e” denotes the mathematical constant known as Euler's number. It is approximately “2.72”. The typical range for the “Dec” input of the Modal Bank Module is [0 ... 100] where a value of “0” corresponds to a decay time of 1 ms, a value of “20” corresponds to a decay time of 10 ms, a value of “200” corresponds to a decay time of 100 ms, and so on. The formula for calculating the decay time (“T”) from the value at the “Dec” input port is $T = 10 ^ {(\text{Dec} / 20)} \text{ ms}$.
- **(Dmp)** “Dmp” (damping) is the Event input port for specifying the global damping parameter. The damping value is subtracted from the “Dec” values of all partials. Positive damping values reduce the decay times. Usually the damping parameter is zero at the beginning of the attack of the exciter signal and then is set to a positive value to “damp” the envelopes of the outgoing signal at some later point. Of course, after sending the new value to the “Dmp” input port, an “apply” Event needs to be sent to the “App” input port in order for the damping value to take effect in the behavior of the Module.

- **(App)** “App” (apply) is the Event input port for applying the latest parameter values that have arrived at the various input ports to the actual filters inside the Modal Bank Module. The parameter values sent to the Modal Bank Module are stored in a temporary buffer and only applied to the partials once an Event arrives at the “App” input port. If the value of the Event is greater than zero, the apply operation sets the state variables of all the filters to zero, thus working like a “reset”. If the “apply” Event is less or equal to zero, all new parameters are applied but the filters retain their state variable.
- **(InP)** “InP” (amplitude compensated audio input) is the Audio input port that is usually used for sustained exciter signals that would be too loud when passed through partial filters with low decay times (resonance). The amplitude compensation was calibrated with a white noise signal of constant amplitude which has equal power at all frequencies. Feeding oscillator signals (like a saw wave) into this input port will not necessarily yield good amplitude compensation since the spectrum of such an oscillator is very different from that of white noise.
- **(In)** “In” (audio input) is the Audio input port for that is usually used for impulse-like exciter signals. This input was calibrated using one cycle of a sine wave as an impulse signal.

Output Ports

- **(L)** “L” (left stereo channel) is the output port for the Audio signal that is a mix of all the partials with amplitudes set by the “AL” parameter.
- **(R)** “R” (right stereo channel) is the output port for the Audio signal that is a mix of all the partials with amplitudes set by the “AR” parameter.
- **(Max)** “Max” (maximum number of partials) is the Event output port for the value set in the *Max. Number* edit field of the Module's Function page.

12.5.3 Properties: Function Page

Setting Module to Mono

To set the Polyphony of the Module to the value “1” (Monophonic), engage the *Mono* checkbox. Please refer to section [↑9.3, Polyphony](#) for more information on Polyphony.



Fig. 12.18 The Status area in the Module's Function page holds the checkboxes to set the Module to Monophonic Mode and to activate or deactivate the Module.

Muting the Module

To mute this Module, engage the **Mute** checkbox. This also mutes all Modules that are lying upstream of it and are not connected to any other Modules.

Setting the Maximum Number of Partial

The maximum number of partials that the Modal Bank Module generates has to be specified in the **Max. Number** edit field in the Module's Function page, shown below. The actual number of active partials is specified at the “Num” input port. “Num” values larger than the value set in the **Max. Number** edit field are clipped. If the “Num” input port is left disconnected, the Modal Bank Module automatically sets the number of partials to the value specified by the **Max. Number** edit field. The default value for this edit field is “128”. Also, the value in the **Max. Number** edit field is sent to the “Max” output port. A higher number of active partials requires more CPU resources.



Due to internal optimization, for many computers the CPU load is increased when the number of partials reaches a multiple of 4 partials.

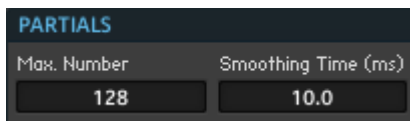


Fig. 12.19 The maximum number of partials and the smoothing time are set in the Partials area of the Modal Bank Module's Function page.

Setting the Smoothing Time

The smoothing time defines the time-interval over which the partial amplitudes are faded between different values. This parameter can be set in the **Smoothing Time** edit field. The units for this edit field are milliseconds and the default value is 10 ms.

Setting the Smoothing Quality

You can specify the quality of the smoothing curves with the **Smoothing Quality** drop-down menu. Higher smoothing quality causes the Module to respond better to fast modulations of the partial amplitudes. However, increasing the smoothing quality also increases the CPU load.

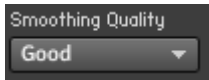


Fig. 12.20 The Smoothing Quality drop-down menu lets you increase the quality of the Module's sound in response to fast modulations of the partial amplitudes.

13 Instruments: Interfaces for Your Structures



Fig. 13.1 The Instrument Object as seen in the Structure View

A REAKTOR Instrument usually represents a musical instrument or effect. This means that it not only holds a Structure, but also equips it with the necessary interfaces for communication with the outside world. In this chapter you will learn about these interfaces, in particular MIDI and OSC. Other Instrument relevant topics are discussed elsewhere in this manual where the context has been found more appropriate. These topics and their locations are found in the following section.

13.1 Why You Should Care About Instruments

Instruments are not only containers for your Structures, but they also provide the necessary interfaces for the Structure to serve a musical purpose:

- For one, each Instrument has its own Instrument Panel which holds the controls that enable you to directly interact with the Instrument. The Instrument Panel is discussed thoroughly in chapter [↑8, Panel Editing](#).
- You can save and recall the states of the Panel controls using Snapshots (also called presets, patches or programs). Each Instrument has its own list of Snapshots. In the Function page of the Instrument's Properties you can specify how saving and recalling Snapshots is done in relation to other Instruments (see section [↑4.8, Defining Snapshot Behavior for Instruments](#) for more information on this).
- REAKTOR supports polyphonic playing of Instruments. Settings that determine among other things the number of Polyphonic Voices are set in the Properties of the Instrument. Each Instrument can have its own Voice settings. For more information on Polyphony, please refer to section [↑9.3, Polyphony](#).

- Also, Instruments enable your Structures to communicate with other Instruments or external devices using MIDI or OSC. MIDI and OSC are discussed in the sections [↑13.2, Setting up MIDI Connections](#) and [↑13.4, Setting up OSC Connections](#), respectively.
- These are the fundamental differences between Instruments and Macros. Although Macro also are containers for your Structures, they don't provide separate Instrument Panels, Snapshots, or means of communication via MIDI or OSC.



Fig. 13.1 The Panel of an Instrument is shown. The gray bar at the top of the Instrument is the Instrument Header.



The MIDI Activity display in the Instrument Header is described in section [↑3.5, Other Common Operations](#). The Voices display in the Instrument Header is described in [↑9.3.2, Changing the Number of Polyphonic and Unison Voices](#).

Structure-Specific Features of Instruments

Instruments can be placed into Structures much like other Objects. You can even place an Instrument inside another Instrument. However, an Instrument's ports only send and receive Monophonic signals. This means that unless the number of Voices of the Instrument is set to “1”, you always have to place a Voice Combiner Module in front of the Out Modules in the Instrument's Structure. For more information on Polyphony, please refer to section [↑9.3, Polyphony](#).

What is the Difference between Instruments and Ensembles?

In any given Structure, there is a top-level Instrument. When you are looking at the highest hierarchy level of a Structure (where the Audio In and Audio Out Modules lie), you are actually still inside an Instrument — the top-level Instrument. This top-level Instrument is called an Ensemble. The Ensemble is everything that an Instrument is, but with a few additional features.

First you might notice that instead of conventional In and Out Modules the Ensemble has Audio In and Audio Out Modules. Thus, an Ensemble is the only Object in REAKTOR which has a direct connection to your soundcard. Furthermore, when saving an Ensemble (as an *.ens file), you also save global settings such as Sample Rate, Control Rate, and Clock BPM.

Ensembles can be inserted into Primary Level Structures as Instruments by right-clicking the Structure and opening the *Insert Ensemble* submenu. In this case the inserted Ensemble's Audio In and Audio Out Modules will be replaced by conventional In and Out Modules and its global settings will be discarded. You can also import Instruments to be loaded as Ensembles using the *File > Import Instrument* menu entry. In this case the In and Out Modules will be replaced by Audio In and Audio Out Modules.

13.2 Setting up MIDI Connections

MIDI (musical instrument digital interface) is a communication protocol that operates with integers in the value range [0 ... 127]. Using the MIDI protocol, REAKTOR Instruments can communicate with external devices (such as a host sequencer or MIDI keyboard) and among each other. This section describes how to configure your Instruments and their Panel controls to send and receive MIDI messages.

13.2.1 MIDI In

This subsection describes how to configure your Instrument to receive MIDI messages. Note that by choosing you Instrument to be a Voice & MIDI Slave to another Instrument, the most MIDI In settings are overridden by the MIDI In settings of the Voice & MIDI Master Instrument. Please refer to subsection [↑9.3.3, Changing an Instrument's Voice Allocation Settings and Tuning](#) to learn how to set an Instrument to be Voice & MIDI Slave to another Instrument. The MIDI In Settings that are overridden are:

- External MIDI In Device (replaced by the entry “*Slave Mode*”)
- MIDI In Channel
- Upper Limit and Lower Limit edit fields
- Sustain Ctrl and Hold Ctrl edit fields

Choosing a MIDI In Device

Before your Instrument can receive any MIDI messages, you need to choose their origin. For example, in stand-alone mode you can choose your MIDI keyboard or another sequencer Instrument to control a synthesizer Instrument. To see how to select the device or devices to control your Instrument, follow these instructions:

1. Make sure you have two Instruments open: a synthesizer and a sequencer such as Monoliner. You will find Monoliner in the Insert *Instrument* > *Sequencers* submenu.
2. Go to the Connections page of the synthesizer Instrument.

A rectangular button with a thin border and the word "CONNECT" in all caps.

3. To choose an external MIDI In device, select the appropriate device from the External Device drop-down menu. This action is shown below.



Fig. 13.2 With the External Device drop-down menu you can choose which MIDI device your Instrument should receive MIDI messages from. Note the entries “None” and “All”.



Only MIDI devices that were activated in the MIDI and Audio Settings dialog box will show up in the *External Device* drop-down menu. Note that you have to activate MIDI In and MIDI Out devices separately. Please refer to the Getting Started Manual for more information on setting up MIDI hardware.



When building an Instrument that should receive its MIDI signals from external devices, it is often useful to select the *All* menu entry from the *External Device* drop-down menu. This way if you load the Instrument into the plug-in version of REAKTOR it is guaranteed that the Instrument will pick up on the MIDI signals that the host sequencer sends to REAKTOR.

- If you are running REAKTOR in plug-in mode, you have two entries in the *External Device* drop-down menu. When you select the *plugin* menu entry, then the Instrument receive MIDI from the host sequencer. If *None* is selected, then no MIDI signals from the host sequencers directly reach the Instrument.
- In addition to external MIDI devices, you can also choose open Instruments from which to receive MIDI signals. To let the Monoliner sequencer control your synthesizer Instrument, choose the Monoliner menu entry from the *Internal Instrument* drop-down menu. Note that choosing an Instrument such as a sequencer as an internal MIDI In Device also causes the target Instrument (such as a synth) to be selected in the sequencer Instrument's *Internal Instrument* drop-down menu in its MIDI Out area.



Fig. 13.3 The Internal Instrument drop-down menu

Now press the **Start Clock** button to start the Monoliner sequencer. You should now see (and hear) the synthesizer being sequenced by Monoliner.



Fig. 13.4 The Start Clock button



Note that you can select multiple MIDI In sources for an Instrument.

Choosing the MIDI In Channel

After selecting the external MIDI device from which your Instrument should receive MIDI signals, you need to make sure that the correct MIDI channel is chosen. MIDI signals can be transmitted over 16 distinct channels. For example, if your MIDI keyboard is configured to send its MIDI messages to channel 1 and your Instrument is configured to channel 2, then your Instrument will not receive the MIDI messages sent from the MIDI keyboard. The MIDI channels of the sending and receiving parties need to match. You select the MIDI channel using the **Channel** edit field shown below. By entering “0” into the **Channel** edit field you set the MIDI channel to Omni. This means that your Instrument accepts MIDI signals from all 16 MIDI channels. In that case it doesn’t matter what MIDI channel your MIDI device is set to.



Fig. 13.5 You can select MIDI channels 1 to 16 by entering the appropriate number in the Channel edit field. By entering “0” into the Channel edit field you set the MIDI channel to Omni.

Determining the MIDI Note Range

Commonly external MIDI devices and sequencer Instruments such as Monoliner are used to tell REAKTOR Instruments which notes to play. These messages arrive at the Instrument's MIDI interface and carry values in the MIDI scale. Each MIDI note carries its note

number (equivalent to pitch) and its velocity value. With the **Upper Limit** and **Lower Limit** edit fields, you can choose to have your Instrument only react to MIDI notes that are within the range set by them.

An application is when you want to use one keyboard to control multiple instruments. For example, you can choose one Instrument to have a **Lower Limit** of “C3” and an **Upper Limit** of “C4” and a second Instrument to have a **Lower Limit** of “C4” and an **Upper Limit** of “C5”. This way if you play notes in the octave C3-C4 you address the first Instrument and when you play in the octave C4-C5 you address the second Instrument.

- To enter the **Lower Limit** and **Upper Limit** values into the edit fields, first double-click the edit field, then type either the MIDI note number or the name of the note (such as “B1”).

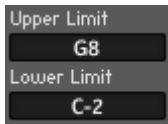


Fig. 13.6 You can choose to limit the range of the MIDI notes that your Instrument reacts to using the **Lower Limit** and **Upper Limit** edit fields.



For more information regarding the MIDI scale, please refer to subsection [↑9.4.1, The MIDI Scale](#).

Transposing the Incoming MIDI Notes

You can transpose all incoming MIDI notes up or down a fixed number of semitones. This can be useful when you are using the split keyboard setup discussed under the previous heading and want both Instruments play in the same octave, although addressing them with keys that belong to different octaves of the MIDI keyboard.



Note that in addition to the **Note Shift** edit field, the **Tune** edit field in the Instrument's Function page also affects the tuning of the Instrument. See subsection [↑9.3.3, Changing an Instrument's Voice Allocation Settings and Tuning](#) for more information on the **Tune** edit field.

- To transpose the incoming MIDI notes, enter the desired number into the **Note Shift** edit field, shown below.



Fig. 13.7 Here all incoming MIDI notes are transposed upwards 12 semitones which corresponds to an octave.



For more information regarding the MIDI scale, please refer to subsection [↑9.4.1, The MIDI Scale](#).

Specifying Common MIDI Controller Numbers

REAKTOR Instruments let you reserve MIDI controller numbers for three special controls: the sustain pedal, the hold pedal, and Snapshot Morphing. In order for these features to be activated, you have to do two things:

1. Engage the checkbox that is next to the edit field belonging to the desired control.
2. Enter the desired MIDI controller number into the corresponding edit field.

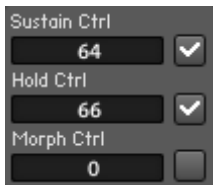


Fig. 13.8 There is an edit field to specify the MIDI controller number for each special control: the sustain pedal, the hold pedal, and Snapshot Morphing.

Details on each of these controllers are listed below:

- **Sustain Ctrl** specifies the number of the MIDI controller that functions as a sustain pedal (called “hold” or “damper pedal” by some MIDI equipment manufacturers). Its standard controller number is “64”. As long as the sustain pedal is turned on, any playing note will be held, even after its key has been released.
- **Hold Ctrl** specifies the number of the MIDI controller that functions as a hold pedal (called “sostenuto” by some MIDI equipment manufacturers). Its standard controller number is “66”. All notes that are playing when hold is turned on will be held even after their key has been released until hold is turned off. Notes that are played when hold is already on are not affected.
- **Morph Ctrl** defines the controller number used for Snapshot Morphing. For a detailed description on Snapshot Morphing, please refer to section [↑4.5, Morphing Between Snapshots](#).

13.2.2 MIDI Out

This subsection describes how to configure your Instrument to send MIDI messages to other Instruments and external MIDI devices.

Choosing a MIDI Out Device

You can send MIDI signals to other Instruments and external MIDI devices. To see how to select the device or devices to which to send MIDI messages, follow these instructions:

1. Make sure you have two Instruments open: a synthesizer and a sequencer such as Monoliner. You will find Monoliner in the *Insert Instrument > Sequencers* submenu.
2. Go to the Connections page of Monoliner.

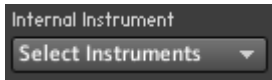


3. To choose an external MIDI Out device, select the appropriate device from the *External Device* drop-down menu.



! Only MIDI devices that were activated in the MIDI and Audio Settings dialog box will show up in the *External Device* drop-down menu. Note that you have to activate MIDI In and MIDI Out devices separately. Please refer to the Getting Started Manual for more information on setting up MIDI hardware.

4. If you are running REAKTOR in plug-in mode, you have two entries in the *External Device* drop-down menu. When you select the *plugin* menu entry, then the Instrument sends MIDI messages to the host sequencer. If *None* is selected, then the Instruments sends no MIDI signals to the host sequencer.
5. In addition to external MIDI devices, you can also choose open Instruments to which to send MIDI signals. To let the Monoliner sequencer control your synthesizer Instrument, go to the *Internal Instrument* drop-down menu and choose the menu entry corresponding the synthesizer you wish to control with Monoliner. Note that choosing an Instrument as an internal MIDI Out Device (in our case Monoliner) also causes that Instrument to be selected in the target Instrument's *Internal Instrument* drop-down menu in its MIDI In area.



6. Now press the **Start Clock** button to start the Monoliner sequencer. You should now see (and hear) the synthesizer being sequenced by Monoliner.



Note that you can select multiple MIDI Out targets for an Instrument.



Only MIDI devices that were activated in the MIDI and Audio Settings dialog box will show up in the **External Device** drop-down menu. Note that you have to activate MIDI In and MIDI Out devices separately. Please refer to the Getting Started Manual for more information on setting up MIDI hardware.



Note that you can select multiple MIDI Out targets for an Instrument.

Choosing the MIDI Out Channel

After selecting the external MIDI device to which your Instrument should send MIDI signals, you need to make sure that the correct MIDI channel is chosen. MIDI signals can be transmitted over 16 distinct channels. For example, if you configure your Instrument to send its MIDI messages to channel 1 and your external device is configured to receive MIDI from channel 2, then the external device will not receive the MIDI messages your Instrument sent. The MIDI channels of the sending and receiving parties need to match. You select the MIDI channel using the **Channel** edit field shown below.



Fig. 13.9 You can select MIDI channels 1 to 16 by entering the appropriate number in the Channel edit field.

13.3 MIDI and the Instrument Panel

In REAKTOR, Instruments provide the MIDI interface for your Structures. A common use of MIDI is to control knobs, faders, and other Panel Elements. These fader and knob controls change their positions according to the received MIDI messages. In this section you will learn how to assign and configure MIDI messages for Panel Elements.



To quickly and automatically assign a MIDI controller to a Panel control, use the “MIDI Learn” feature as discussed in subsection 3.3.2.

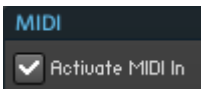
Incoming MIDI Messages for Panel Controls

To assign and configure incoming MIDI messages for Panel controls, perform the following steps:

1. Open an Instrument that has at least one knob or fader on its Instrument Panel.
2. Choose and setup a MIDI In device, as described in subsection [↑13.2.1, MIDI In](#). This is the device from which your Panel controls will receive their MIDI messages from.
3. Double-click on a Panel control such as a knob to open its Properties.
4. Go to the Connections page of the control using the [Connect](#) button.

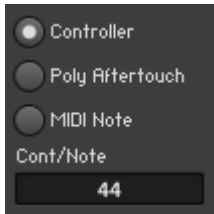
A rectangular button with a thin border and the word "CONNECT" in all caps.

5. First, engage the [Activate MIDI In](#) checkbox (see screenshot below). Now the control can be operated via MIDI.



6. Next you need to choose what type of MIDI signal you want to assign to the control. This is done with the radio button selector shown below. When the [Controller](#) radio button is selected, MIDI controller messages operate the Panel control. In this case the MIDI controller number has to be defined using the [Cont/Note](#) edit field. When the [Poly Aftertouch](#) radio button is selected, MIDI Poly Aftertouch messages operate the Panel

control. Last, if the **MIDI Note** radio button is selected, MIDI note messages operate the Panel control. The MIDI note which operates the Panel control is selected with the **Cont/Note** edit field and the velocity determines the control position.



7. If in the previous step you chose the source of the MIDI messages to be a MIDI controller or a MIDI note, then you need to define the MIDI controller or MIDI note number of the source signal, respectively. Enter the corresponding value into the **Cont/Note** edit field, shown above.
8. Now your Panel control should be configured to receive MIDI messages from the desired source. However, if you chose the incoming MIDI signal to be a MIDI controller, there are some additional properties that let you fine-tune how your control reacts to these messages. The first feature is called “Soft Takeover”. When a fader or knob is operated by an external MIDI controller, it normally jumps straight to the received controller value. Such a jump can be quite noticeable in the sound, depending on the controlled parameter (for example, amplifier level). Thus, such jumps can be undesirable. You can avoid such behavior with the “Soft Takeover” feature. It causes the control to only move when the value received via MIDI reaches or goes past the current position. To activate “soft takeover”, engage the **Soft Takeover** checkbox, shown below.



9. Some MIDI controllers are not conventional knobs or faders, but are dials or rotary knobs. Dials and rotary knobs can be turned continuously — they don't have fixed lowest and highest positions. Such MIDI controllers send increment messages rather than absolute MIDI controller values. If you are operating your Panel control with such a MIDI controller, engage the **Incremental** checkbox, shown above.



When a button control is operated using MIDI, the button is activated when the received MIDI controller or Poly Aftertouch message has a value greater than “63”. When set to “MIDI Note” mode, the buttons are operated using MIDI Note On and MIDI Note Off messages.



When using MIDI controller or Poly Aftertouch messages to operate a switch or list control, the switch or list selects the option that corresponds to the received MIDI value. The range of possible values [0 ... 127] is divided into equal-sized regions according to the number of switch or list entries. For example, a Switch Module with four inputs (that is, four list entries) would have the following four regions: [0 ... 31], [32 ... 63], [64 ... 95], [96 ... 127]. Note that “0” always selects the lowest and “127” always the highest switch or list entry.



When using MIDI controller or Poly Aftertouch messages to operate a switch or list control, the switch or list selects the option that corresponds to the received MIDI value. The range of possible values [0 ... 127] is divided into equal-sized regions according to the number of switch or list entries. For example, a Switch Module with four inputs (that is, four list entries) would have the following four regions: [0 ... 31], [32 ... 63], [64 ... 95], [96 ... 127]. Note that “0” always selects the lowest and “127” always the highest switch or list entry.

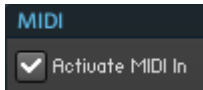
Outgoing MIDI Messages from Panel Controls

There are two ways to send outgoing MIDI messages from Panel controls. The first one is to simply forward incoming MIDI messages to the Instrument's MIDI output. The second one is to convert the Panel control's own output to a MIDI signal and forward that to the Instrument's MIDI output. To choose either or both of these options, perform the following steps:

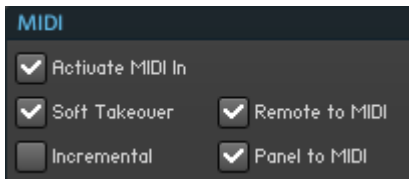
1. Open an Instrument that has at least one knob or fader on its Instrument Panel.
2. Choose and setup a MIDI In and a MIDI Out device, as described in subsections [↑13.2.1, MIDI In](#) and [↑13.2.2, MIDI Out](#), respectively. These are the devices from which your Panel controls will receive their MIDI messages from and send MIDI messages to, respectively.
3. Double-click on a Panel control such as a knob to open its Properties.
4. Go to the Connections page of the control by pressing the **Connect** button.

CONNECT

5. First, engage the **Activate MIDI In** checkbox, shown below. Now the control is open to receive MIDI signals.



6. After assigning a source for the incoming MIDI messages, as discussed under the previous heading, engage the **Remote to MIDI** checkbox (see screenshot below). This forwards the incoming MIDI messages to the MIDI output of the Instrument.



7. If you want to control another Instrument or an external MIDI device with your Panel control, you need to convert its output to the MIDI scale and forward it to the output of the parent Instrument. To do this, engage the **Panel to MIDI** checkbox, shown above.

13.4 Setting up OSC Connections

OpenSound Control (OSC) is a protocol for communication between computers and applications using a variety of network protocols, including TCP/IP and LANs. It was designed with musical purposes in mind. OSC messages can be sent between REAKTOR and other OSC clients. At the time of this writing OSC was supported by devices such as Jazzmutant Lemur, Monome, and the TouchOSC application for iPod Touch and iPhone. Also, you may create OSC connections between instances of REAKTOR running on different computers or REAKTOR and some other OSC compatible application (running on the same or some other machine). OSC communication is only possible when REAKTOR's audio processing is active, and as a consequence, you need an audio card or built-in audio capability on your computer to activate OSC. This section discusses some use cases for OSC and how to configure REAKTOR to make use of this feature.

! The OSC protocol only works in REAKTOR Standalone mode.

13.4.1 Using REAKTOR with an OSC Device

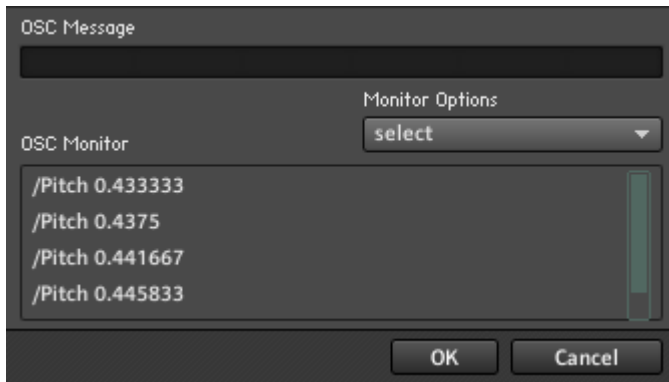
In this subsection you will learn how to setup REAKTOR to exchange OSC messages with an external OSC device. Before you start, please thoroughly read your OSC device's documentation. There you should learn how to establish a connection between the device and your computer on which you are running REAKTOR. To create the actual OSC connection between the device and REAKTOR, perform these steps:

1. Open an Instrument that has at least one knob on its Panel.
2. Open the OSC Settings dialog box by choosing the *File > OSC Settings ...* menu entry.
3. To globally activate the OSC protocol for REAKTOR, engage the *Activate OSC* checkbox, shown below. The Activate OSC status is preserved between REAKTOR sessions.



4. Next, look at the *Local IP Address* display. This shows the current IP address of your computer. It is recognized automatically and cannot be edited. Usually the external OSC device or application requires you to enter the IP Address of the computer that is running REAKTOR. Enter the address that you see in the *Local IP Address* display into the OSC device's edit field that is usually labeled as "Host", "Host IP Address", or "Target IP Address".
5. The *Local Port* edit field, shown in the above figure, lets you enter the port number for OSC signals incoming to REAKTOR. This value is crucial when controlling REAKTOR with an external OSC device. It is the sub-network identifier by which other OSC clients recognize your system when they scan the network (REAKTOR's own *Scan* button is discussed below). Only certain ports are scanned, and you should use a number between 10'000 and 10'015.

6. Now you need to tell your OSC device at which port REAKTOR is awaiting its OSC messages. Your device probably has an edit field that is called something like “Outgoing Port” or “Remote Port”. This is where you enter the port value that you specified in the previous step in the *Local Port* edit field.
7. Now the incoming connection to REAKTOR should have been established. If you send OSC messages to REAKTOR from your device by moving one of its controls, for example, they should show up in the OSC Monitor display, shown below.



8. Use the *Monitor Options* drop-down menu, shown below, to change the way the messages are displayed. Choosing the *New Line* menu entry causes the each new message to be displayed on the first line. The *Append Line* menu entry causes each new message to be displayed on a new line. You can clear the OSC Monitor display by choosing the *Clear* menu entry and with the *Off* menu entry you can turn off the display altogether.



9. Next you should configure the outgoing connection from REAKTOR to the OSC device. REAKTOR can send OSC messages to several targets at once. However, the identifier, IP address, and port of each target OSC device needs to be configured separately. To configure the outgoing connection to an OSC device, first enter the identi-

fier of your OSC device into the Identifier edit field, shown below. Sometimes your OSC device lets you choose this identifier, other times it doesn't and you can make up your own name.

Identifier	IP Address	Port
OSC Device	10.1.10.128	10001

Buttons: Scan, Edit, Delete

Identifier: OSC Devi Remote IP Addr.: 10.1.10.128 Remote Port: 10001 Apply

10. After choosing an identifier, enter the IP Address of your OSC device into the Remote IP Addr. edit field, shown above. This IP address is determined by your OSC device. There you will usually find it in a text field labeled "Local IP Address" or just "IP Address".
11. Next you need to tell your OSC device at which port to expect OSC messages from REAKTOR. For this purpose your device probably has an edit field that is called something like "Incoming Port" or "Local Port". Enter the port value in that edit field. It is a good idea to choose this port number to be one value higher than the port number you specified in instruction step 5.
12. Now you need to tell REAKTOR to which port it should send its OSC messages, it is the port number from the previous step. Enter it to the Remote Port edit field, shown above.
13. Now click on the Apply button, shown above. You should see the OSC device's information appear as an entry in the OSC Members list, shown above.
14. To delete a device from the list of REAKTOR's outgoing OSC connections, select its entry in the OSC Members list and click the Delete button.
15. To edit a device's information, select the corresponding entry in the table and click on the Edit button. The device's information appears in the Identifier, Remote IP Addr. and Remote Port edit fields below the table, as shown above. Now you can edit the information and when you are done, click the Apply button.
16. For some devices, entering the correct IP address and port number for REAKTOR's incoming OSC messages (as instructed in steps 4 to 6) is enough for REAKTOR to be able to find the device, if the following conditions are met. Then all you have to do is

press the **Scan** button, shown above. If REAKTOR finds the device, it will automatically add an entry with the identifier, IP address, and remote port information to the **OSC Members** list.

To use the “scan” functionality, these conditions have to be met:

- REAKTOR must be running on your computer with an active audio engine.
- OSC has to be activated in the REAKTOR OSC Settings dialog box (see instruction step 3).
- In the REAKTOR OSC Settings dialog box a port address between 10'000 and 10'015 must be set (see instruction step 4).
- The OSC client must be located within the same subnet as your computer running REAKTOR.



If you want to connect two computers that are not located in the same subnet (for instance if you want to establish an OSC connection via the internet), you will have to manually enter the identifier, IP address, and port number of the other computer or OSC device into the edit fields below the OSC Member list and then press the **Apply** button.

13.4.2 Defining OSC Synchronization Settings

When you are using OSC to communicate between REAKTOR and another application with a global clock such as a host program, then it can be desirable to have the global clocks of the two applications synchronized. This can also be the case when using OSC to communicate between two instances of REAKTOR. In this subsection you will learn about the OSC Synchronization:

- To have REAKTOR send an OSC clock signal to other OSC clients, engage the **Clock Sync (Master)** checkbox, shown below. The OSC clock works exactly like a MIDI clock; it will be sent to all clients in the OSC Members list.
- Alternatively you can use the “OSC Time Sync” feature to synchronize with clients in the OSC Members list. Time Sync is a control circuit system: the client constantly polls the application that has been set to “master” for the time stamp. It compares the received time stamp with its own and adjusts it if necessary. Engage the **Time Sync (Master)** checkbox, shown below, to set REAKTOR to be the Time Sync “master” application.

- When both the *Clock Sync (Master)* and *Time Sync (Master)* checkboxes are disengaged, you can use the *Select Master* drop-down menu (shown below) to choose a clock source. REAKTOR then synchronizes its clock to that of the source. Select the *Clock Sync* menu entry to synchronize to incoming Clock Sync signals. Alternatively you can choose a menu entry corresponding to another OSC member to Time Sync with that client.

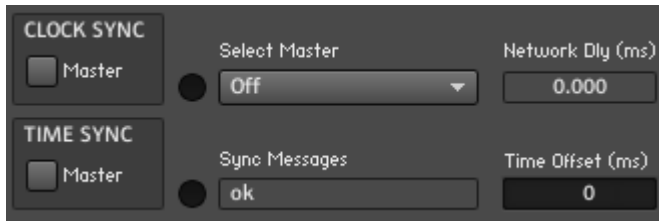


Fig. 13.10 The OSC Synchronization area provides you with settings to synchronize the clocks of REAKTOR and the other OSC clients.

- To add a time offset to each OSC message that is sent to clients, enter the corresponding value in milliseconds into the *Time Offset* edit field, shown above. For example, if you enter the value “1000” into this edit field, each message will be received one second later by the client. This feature only applies if the participating clients are operating in Time Sync Mode.

There are small LEDs to the right of the *Clock Sync* and *Time Sync* checkboxes, shown in the screenshot below. These indicate when a synchronization signal is received or sent. To monitor the status of synchronization messages, view the *Sync Messages* display, shown in the screenshot above. This field reports synchronization errors.

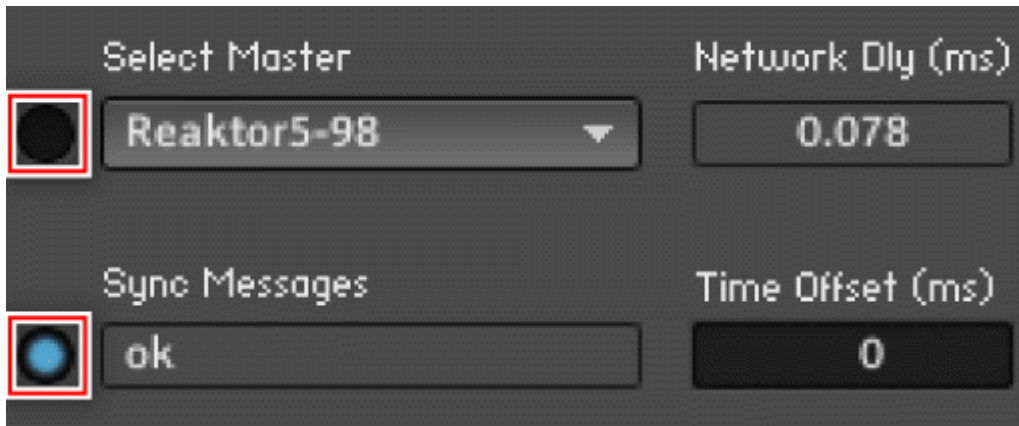


Fig. 13.11 The LEDs next to the Clock Snc and Time Sync checkboxes indicate when a synchronization signal is received or sent.

14 Advanced Operations and Preferences

In this chapter you will learn about important functionalities and preferences in REAKTOR which have not been covered in the previous 13 chapters. First you will learn about CPU resources in the context of debugging and optimizing your Structures. Then you will learn how to use the Internal Connection protocol with the help of IC Send and IC Receive Modules. Following that, you will find out how you can control which Instrument parameters can be automated in hosts (when REAKTOR is running as a plug-in). Last, the remaining options in REAKTOR, in particular the Preferences dialog box, are explained.

14.1 CPU, Debugging, and Optimizing

When building complex Structures in REAKTOR, you will be confronted with several challenges. One challenge is to build Structures which don't use up all of your computer's resources. The most important resource to take into account is the CPU. In this section you will first learn about the ways you can monitor and save CPU resources when using REAKTOR. With complex signal flows it is often helpful to use the debugging tools offered by REAKTOR to monitor the Event Initialization Order and Module Sorting. In this section you will also see how to access these tools.



Note that in order for your Ensembles to function, REAKTOR's audio engine needs to be turned on. An active audio engine is signaled by the white **Audio Engine** button in the Main Toolbar. If an Ensemble requires more CPU resources than are available, a dialog window pops up to inform you of this. Thereafter the audio engine is automatically turned off, as can be seen by the red **Audio Engine** button. Press that button to reactivate the audio engine.



Fig. 14.1 The Audio Engine button (on)



Alternatively to pressing the **Audio Engine** button you can use the keyboard shortcut Ctrl+R if you are in Windows, Cmd+R if you are in Mac OS X.

14.1.1 Monitoring CPU Load

Event and Audio signal processing in REAKTOR requires CPU resources. There are two separate threads which process these signals: the Audio Thread and the GUI Thread. Although both cause a load for the CPU, it is the CPU load of the Audio Thread which is displayed in the CPU Meter.

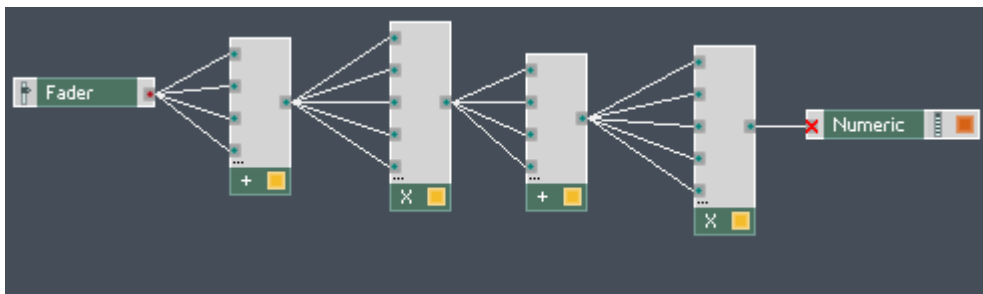


Fig. 14.2 The CPU Meter shows the CPU load produced by the Audio Thread.

Whether a signal is processed in the Audio Thread or GUI Thread depends on the source of the signal. As the name implies, signals which are processed by the GUI Thread originate from the graphical user interface (or GUI, in short). These signals are initiated by mouse clicks, mouse movements, or the computer keyboard. On the other hand, signals which are initiated by audio processing functions (a sine oscillator function, for example) or by incoming MIDI and automation messages are processed in the Audio Thread.

Let's look at the Fader Module as an example. To see the GUI Thread and the Audio Thread in action, perform the following steps:

1. Create a new Instrument. In its Function page, set its number of Voices to something rather large: "200" might be a good number, depending on your computer.
2. Open the Instrument's Structure and insert a Fader Module.
3. Connect the output of the Fader Module to a series of simple operations and at the end connect a Numeric Display Module (see Structure below).



4. Next go to the Numeric Module's Function page and engage the *Always Active* checkbox. All the Modules in the Structure should become active (please refer to section [↑7.4, Creating an Active Signal Flow](#) to see how to activate objects).
5. Now go to the Instrument Panel and use your mouse to move the knob while looking at the CPU Meter. You should see that moving the knob has no effect on it. Although Events from the knob are processed by the Modules downstream and therefore cause a load for your CPU, it is not shown on the CPU Meter. This happens because operations which are triggered by a mouse action are processed by the GUI Thread. Remember that the GUI Thread does not show up on the CPU Meter.



Note that when a high number of Events is triggered from the GUI, a significant CPU load could be induced. However, because this CPU load is caused by the GUI thread, it does not show up on your CPU Meter.

6. To see the processing happen in the Audio Thread, we will use MIDI to move the knob. Operations triggered by incoming MIDI messages are processed in the Audio Thread. Connect a MIDI Device to REAKTOR (or use a host sequencer). Please refer to section [↑13.2, Setting up MIDI Connections](#) to learn how to set up MIDI connections for your Instrument.
7. Now assign the knob to a MIDI controller such as a knob on a MIDI keyboard or an automation parameter in the host sequencer.
8. Last, move the knob in your Instrument using your MIDI device. Notice how the CPU load increases. This is because each time the knob is moved, it sends out an Event which is processed by the Modules downstream and in the Audio Thread.



Generally you cannot explicitly see if a Module does its processing in the Audio Thread or the GUI Thread. It is something you have to figure out on a case to case basis.

Note that generally an Event causes more CPU load than the same operation processed in the Audio Thread. However, as opposed to Audio signals which are permanently updated at the Sample Rate, Events only trigger operations when they are sent. If for certain tasks you use Events (sparingly) instead of the Audio Signal, you can significantly reduce your CPU load.

There is one more thing to be aware of. Audio signals are always processed at the Sample Rate (no faster, no slower). However, with the use of the Iteration Module you can cause a flood of many Events. Unless the *Limited Speed* checkbox in its Function page is enabled, all

of these Events are sent between the processing of two audio samples. This means that with Events you can operate much faster than the Sample Rate. If in such a case your Events are tied to CPU hungry operations, your computer's CPU can be brought to its knees. This can cause clicks or even dropouts of the audio output signal. For this purpose, the CPU Meter has a yellow bar that is capable of detecting such peaks in CPU usage. Again, it only monitors the Audio Thread. The screenshot below shows how the CPU Meter looks when there is a spike in the CPU load. CPU usage without seeing an increase in the CPU Meter.

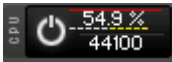


Fig. 14.3 The yellow bar of the CPU Meter helps you detect (potentially Event induced) spikes in CPU usage.



Sometimes your Iteration Module is triggered by a mouse or computer keyboard action from the Instrument Panel. If in such a case a large number of operations are triggered by these Events, you could have significant spikes in CPU load which don't show up on the CPU Meter. This is because these operations are processed by the GUI Thread.

14.1.2 REAKTOR's Debugging Tools

The REAKTOR Primary Level offers three tools which help you gather information about your Structure which is otherwise hidden. These Debugging Tools are accessed from the *Options > Debug Structure* submenu or, as shown in the screenshot below, by clicking the Menu button and browsing the *Options > Debug Structure* submenu.

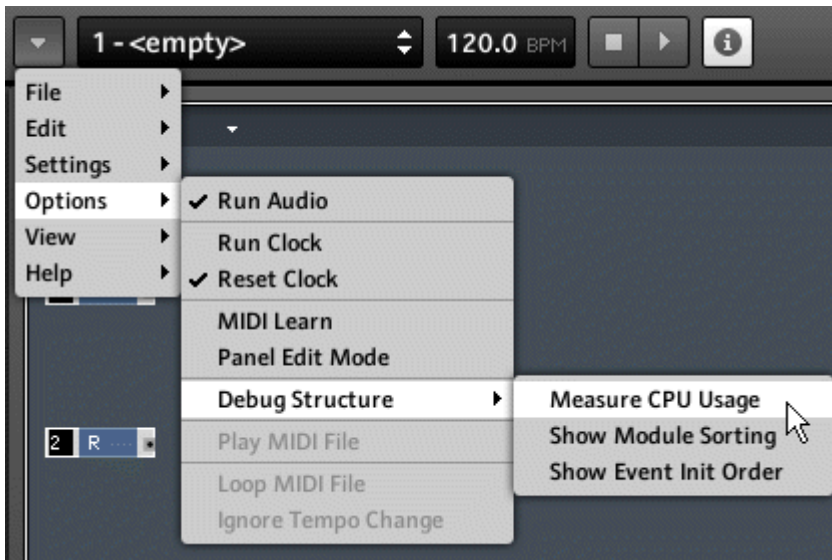


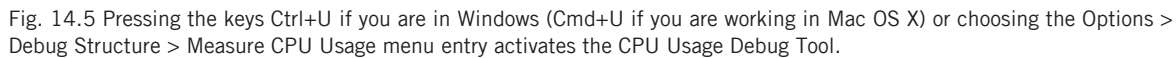
Fig. 14.4 You can access the Debugging Tools from the Options > Debug Structure submenu.

Measuring CPU Usage

In the Primary Level you can measure the CPU usage of individual Modules. Although this is a rough estimate of the actual CPU load induced by the Modules, it helps you get an idea which Modules are using significantly more CPU in relation to others. This way you know what areas of your Structure are worth optimizing. When you are using the CPU Usage Debug Tool, a black display for each Module shows its CPU usage in percent of the total CPU resources. Modules which use less than 0.1 % don't have a display because their CPU usage is generally negligible. This is shown below.

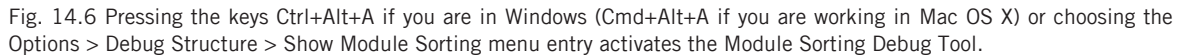
! Note that using the CPU Usage Debug Tool causes an increase in the overall CPU Usage, as seen on the CPU Meter.

- To measure CPU usage of Primary Level Modules, choose the *Options > Debug Structure > Measure CPU Usage* menu entry or press the keys Ctrl+U if you are in Windows (Cmd+U if you working in Mac OS X). You can also click on the Menu button and choose the *Options > Debug Structure > Measure CPU Usage* menu entry.



In Primary Level when performing signal processing with Audio signals, the order of operations becomes important. This is particularly crucial if you have feedback Structures utilizing the Unit Delay Module. Use the Module Sorting Debug Tool to see in which order the operations are performed on your Audio signal. With this Debug Tool active, a blue display for each Module shows its place in the sequence in which the operations are performed on the Audio signal. This is shown in the Structure above.

- REAKTOR 5.5 - Application Reference - 419



In more complex Structures, Event Initialization becomes a crucial part of your Structure. Some Modules like the Multi Display, Modal Bank, and Sine Bank Modules require Events to arrive in a certain order at their input ports before functioning in the intended way. This is also relevant for the Initialization procedure of Events. Please read subsection [↑9.2.5, Initialization](#) for more information on how Event Initialization takes place in the Primary Level. To see the order in which Modules that process or send Events are initialized, use the Event Initialization Order Debug Tool. With this Debug Tool active, a purple display for each Module shows its place in the sequence in which Event functions are initialized. This is shown in the Structure below.

- To activate the Event Initialization Debug Tool, choose the *Options > Debug Structure > Show Event Init Order* menu entry or press the keys Ctrl+Alt+E if you are in Windows (Cmd+Alt+E if you working in Mac OS X). You can also click on the **Menu** button and choose the *Options > Debug Structure > Show Event Init Order* menu entry.

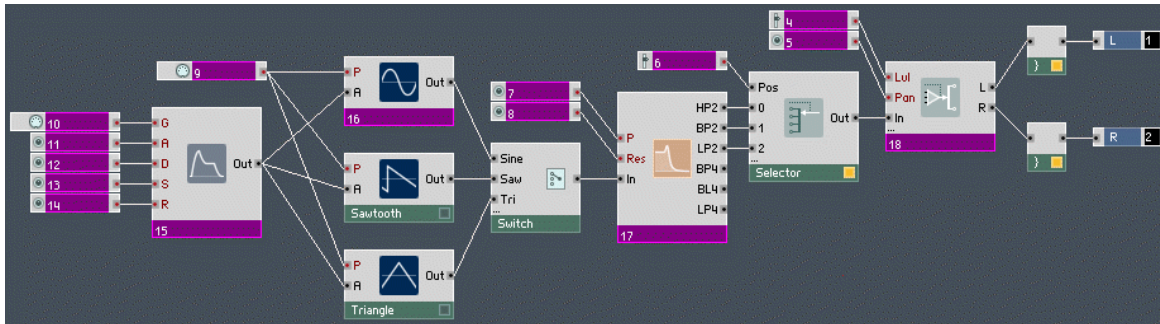


Fig. 14.7 Pressing the keys Ctrl+Alt+E if you are in Windows (Cmd+Alt+E if you are working in Mac OS X) or choosing the Options > Debug Structure > Show Event Init Order menu entry activates the Event Initialization Order Debug Tool.

Testing Initialization Using the Audio Engine Button

Although the Event Initialization Order Debug Tool is helpful when determining the Initialization behavior of your Structures, it is often more useful to see the Initialization take place when debugging Initialization related issues. Although the most definite way to determine how your Structure acts upon the loading of the Ensemble which holds it is to reload the whole Ensemble, most issues can be debugged using the Audio Engine button.



Fig. 14.8 The Audio Engine button (on)

- To reinitialize your Structure, simply click twice on the Audio Engine button. When the button is in its Off state, it turns red.

Event Loops

In general, feedback loops consisting of Event signals are allowed in REAKTOR. However, Events often trigger other Events. Therefore, an Event feedback loop can cause an infinite chain reaction where an Event from an initial source triggers an Event which in turn triggers another Event from the same initial source, and so on. Such a pernicious Structure can quickly lead to a stack overflow which in turn results in prompt termination of the REAKTOR application. Although not all Event feedback loops are fatal, it is definitely not a recommended programming solution because of the potentially fatal results for your Instrument. Therefore REAKTOR offers you a sort of protection from Event Loops by letting

you turn the Structures capability for Event Loops off altogether. This is a property which can be activated in REAKTOR's global Preferences dialog window or for individual Instruments and Macros basis. If an Event Loop is triggered in an Instrument when the Instrument's Event Loops have been disabled (analogously in the case of a Macro) or Event Loops have been globally disabled altogether, a dialog box pops up alerting you of the fact that an Event Loop has been triggered in the Structure. As shown in the screenshot below, it will ask you if you want to be navigated to the part of the Structure where the Event Loop has taken place.

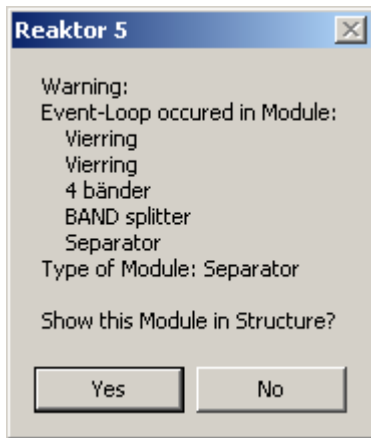


Fig. 14.9 When Event Loops have been disabled and an Event Loop is triggered, a dialog box pops up to alert you of the fact that an Event Loop has occurred. It will ask you if you want to be navigated to the part of the Structure where the Event Loop took place.

- To enable Event Loops for an Instrument or Macro, go to the Object's Properties and engage the **Enable** checkbox in the Event Loops area of the Function page, as shown in the figure below.



Fig. 14.10 The Enable checkbox in the Event Loops area of an Instrument's or Macro's Function page lets you activate Event Loops for its Structure. However, this option is overridden if Event Loops have been globally disabled in the REAKTOR Preferences dialog box.

To globally disable Event Loops for all Instruments, perform the following steps:

1. To globally disable Event Loops, go to the REAKTOR Preferences dialog box using the *File > Preferences ...* menu entry.
2. Then click on the Options tab.

A dark rectangular button with the word "Options" in a light-colored sans-serif font.

3. Last, engage the Globally Disable Event Loops checkbox.

A dark rectangular button containing a white checkmark icon followed by the text "Globally disable event loops" in a light-colored sans-serif font.

4. Engaging the Globally Disable Event Loops checkbox overrides the Event Loops settings of all open Instruments. This setting is preserved between opening different Ensembles and also when REAKTOR is closed and reopened.

14.1.3 Tips to Reduce CPU Load

As mentioned in subsection [↑14.1.1, Monitoring CPU Load](#), your computer's CPU is the most critical resource you should consider when building in REAKTOR. This subsection lists a few tips which may help you reduce the CPU load of your Ensembles:

- **Reduce the number of Polyphonic Voices:** REAKTOR allows you to build Polyphonic Instruments. This means that Polyphonic parts of your Structure process as many signals in parallel as there are Polyphonic Voices set for your Instrument. Obviously, the more Voices your Structures processes, the greater the load on the CPU. When building a polyphonic synthesizer, try to keep the number of Voices as low as possible. If you are using a polyphonic synthesizer in a situation where CPU usage is critical and at the same time you don't need to play many Voices at the same time, you can try reducing the number of Polyphonic Voices of the synthesizer to reduce its CPU load. Please refer to section [↑9.3, Polyphony](#) for more information on Polyphony.
- **Only process signals in Polyphonic Mode when necessary:** From the previous remarks it follows, that you should set all Modules, Macros, and Instruments which don't need to operate Polyphonically to Monophonic Mode (number of Voices equal to "1"). For example, often when you are applying effects like reverb or chorus at the output of a polyphonic synthesizer, it is a smart thing to make sure that the effects are running in Monophonic Mode. Reduce Control Rate

- **Automatic Voice Reduction:** In some cases when using Polyphonic Instruments it makes sense to turn off unused Polyphonic Voices. This can be done automatically using the “Automatic Voice Reduction” feature. Note, however, that since this feature has to monitor the CPU load and turn Voices on and off, it also has a CPU overhead which might be larger than the actual benefit gained from reducing the number of Voices. To activate this Instrument-specific feature, go to the Instrument's Function page and engage the Automatic Voice Reduction checkbox.

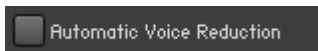


Fig. 14.11 The Automatic Voice Reduction checkbox

- **Process with Events instead of Audio:** In many cases it doesn't make sense to process a signal at the Sample Rate. However, there are hybrid Modules can operate at the Sample Rate (processing Audio signals) or at the rate of incoming Events, when processing Event signals. With a hybrid Module it can happen that you connect an Audio signal to one of its input ports and cause it to operate at the Sample Rate when you actually only need it to update at the Control Rate. This would unnecessarily use up CPU resources. A solution to this could be using the Value Module to latch the Audio signal to the hybrid Module. This way the hybrid Module only does its processing when Events arrive at its input ports.
- **Eliminate superfluous Events:** As described in section [↑9.2, Event Signals](#), even a simple operation like adding the values carried by two Events can be tied to having unwanted Events at a Module's output. In the case of the Add Module you might just be interested in one Event that carries the sum of the values of the two incoming Events. However, each of the incoming Events triggers an Event at the output of the Add Module so you end up with one superfluous Event at the Add Module's output. In Structures where an unwanted Event can trigger a chain reaction of unnecessary signal processing, this one Event can end up using valuable CPU resources. In such a case it is desirable to eliminate the superfluous Events. For the example of the Add Module, a way to inhibit the unwanted Event from proliferating downstream is discussed in subsection [↑9.2.4, Controlling the Event Order](#). This solution makes use of the Order and Value Modules. The Value Module is generally very useful for causing signal values to be sent downstream only when a trigger Events arrives. This technique is called latching. Another useful Module is the Step Filter Module. It blocks all

Events which carry a value that is within a certain tolerance interval from the value carried by the last Event at its input. The tolerance interval is set at the “Tol” input port. With the value “0” at the “Tol” input port, the Step Filter Module blocks all Events that carry the same value as the previous Event.

- **Reduce the Control Rate:** Some Modules such as the LFO Module send Events from their outputs at the Control Rate. If these output Events in turn trigger many CPU-intensive operations, reducing the Control Rate can have its benefits. The Control Rate is changed using the *Settings > Control Rate* submenu.
- **Take advantage of Core:** For Structures that contain a large number of low-level operations such as math operations, relays, separators, unit delays, and so on, it is typically more efficient to build them in Core. With Structures built of larger scale Modules such as Filters this might or might not also be the case. In the Monophonic case or sometimes with a low number of Voices you can expect a similar performance from a single Primary Level Module compared to a Core Structure. However, as the number of Voices grows, a Primary Level Module will generally outperform a Core Cell. It is also important to note that inserting a Core Cell into a Primary Level Structure has an overhead in the CPU load. This means that once you have inserted a Core Cell and are working on Core, it is best to try to pack everything that you want to do in Core in that one Core Cell, if possible.

A Few Additional Notes About Core Cells

In general, it is not more efficient to use a Core Cell with equal functionality instead of a Primary Level Module. In the best case you will get the same performance. Although the factory library Core Cells are not more efficient than their Primary Level counterparts, there are several reasons why you should use them. Core Cells are not completely equivalent in functionality to their Primary Level counterparts as they provide features not found in the latter. For example, although Audio input ports for Core Cells use more CPU resources, they enable you to make modulations that sound much smoother than their Event counterparts. Such aspects, among many others, can all be customized when using Core Cells. Latching, as previously described in this subsection, is a common practice in Core. In particular, it has been used in the Standard Library Core Cells. Latches are not really CPU-intensive and therefore it usually does not make sense to remove latches to reduce CPU load in Core. Usually latches are used to block unnecessary (Core) Events which in

some cases might even reduce the CPU load a little. Depending on the internal Structure of a Core Cell, it might improve its performance if you replace an Audio input port with an Event input port. However, in such a case you would sacrifice the smooth modulation at the Sample Rate. In general. Unless it brings you some direct benefits, you shouldn't convert your Structures to use Core Cells instead of Primary Level Modules. For now there are no plans to drop the existing Primary Level Modules.

14.2 Using the Internal Connection Protocol

The Internal Connection protocol (IC protocol, in short) is a feature that lets you directly link any two controls on your Instrument Panel(s). Furthermore you can also send and receive Events between of those controls and the corresponding IC Module in your Structure. Perform the instruction steps in this section to learn how to work with the IC protocol.



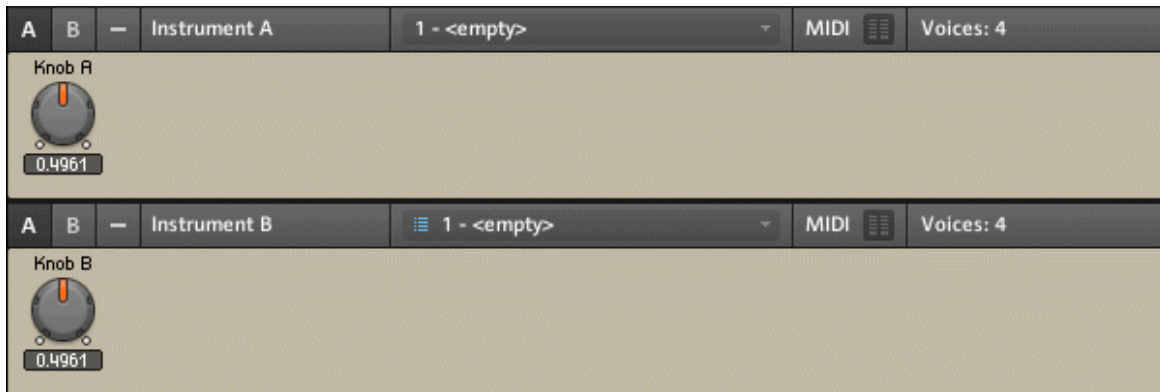
The IC protocol only works with Event streams; it is not compatible with Audio signals. If you want to use wireless connections in place of Primary Level Wires, the please use the Send and Receive Modules found in the *Built-In Module > Terminal* submenu.

14.2.1 Connecting Two Panel Elements Via the IC Protocol

1. Create an Ensemble with two active Instruments. Name the Instruments “Instrument A” and “Instrument B”, respectively.
2. Go to Panel View and unlock the Instrument Panels using the Panel Lock button.



3. For each Instrument Panel, right-click the Panel area and choose the *Insert Built-In Module > Knob* menu entry.
4. Double-click on each knob's Label and rename the knobs to “Knob A” and “Knob B” for Instrument A and Instrument B, respectively.
5. Lock the Instrument Panels by pressing the Panel Lock button. Your Panel view should look similar to the screenshot below.



6. To cause Knob A to send an internal connection signal to Knob B, go to the Connections page of Knob A.



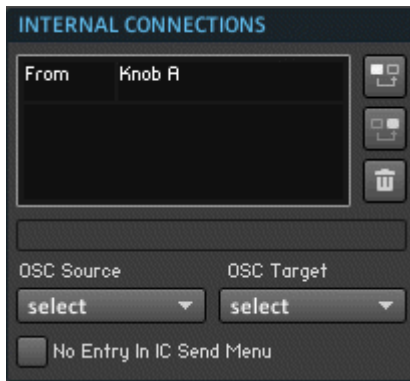
7. Press the IC Send button.



8. The IC Send functionality of Knob A has been activated. Now you need to choose to which control Knob A should send its IC signal. Go to the Connections page of Knob B.
9. To make Knob B receive the IC signal from Knob A, press the IC Receive button.



10. Now the Internal Connections area of Knob B's Connections page should look similar to the one shown below.



11. Test out the Internal Connection you just made. Move Knob A on the Panel and see Knob B snap to the same position as Knob A. Note that this holds even if Knob B has a different value range than Knob A. Also notice that moving Knob B has no effect on Knob A.
12. To see that you can simultaneously also send an IC signal from Knob B to Knob A, go to Knob A's Connections page.
13. Click on the IC Receive button. Knob A will now be linked to the first control that is moved on any Instrument Panel.
14. To make the link, move Knob B. Now you should notice that the two knobs are linked regardless of which one is moved. The Internal Connections area of Knob B should now look similar to the one shown below.

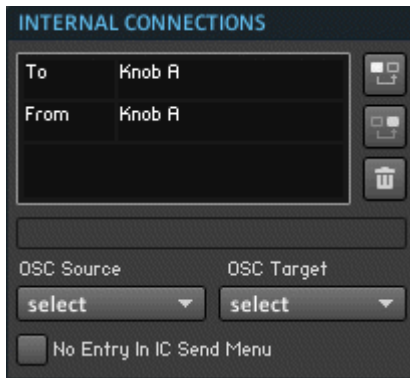
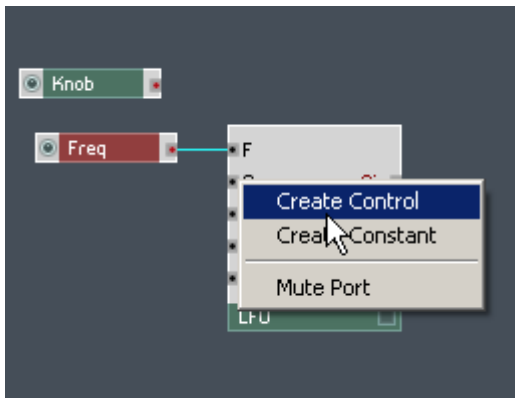


Fig. 14.12 The Internal Connections area of Knob B shows that it sends and receives an Internal Connection signal to and from Knob A, respectively.

14.2.2 Modulating a Control with a Signal from the Structure

In this small tutorial you will see how to modulate a Knob Module with an LFO using the IC Send Module. Perform the following steps:

1. Create a new Instrument with a knob on its Panel and enter Instrument's Structure.
2. Place an LFO Module from the *Built-In Module > LFO, Envelope* submenu into the Structure.
3. Create Knobs for the LFO amplitude and frequency at the “A” and “F” input ports, respectively. Do this by right-clicking the corresponding port and choosing the *Create Control* menu entry. This is shown in below.

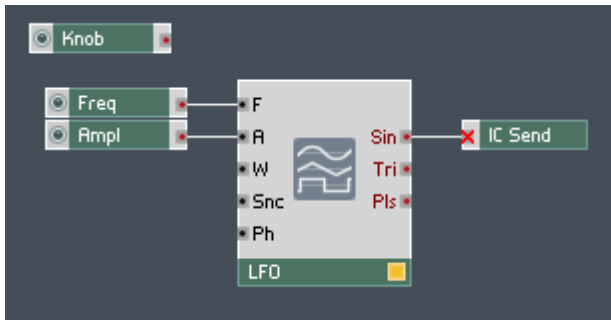


4. Now you are ready to use an IC Send Module to route the output of the LFO Module to the Knob Module. Right-click on the Structure background and insert an IC Send Module from the *Built-In Module > Terminal* submenu into the Structure.
5. To route an LFO signal with a sinusoidal waveform to the IC Send Module, connect the “Sin” output port to the IC Send Module. Notice that both the LFO and IC Send Modules are still inactive.
6. To activate the IC Send Module (and hence the connected LFO Module), go to its Function page.

FUNCTION

7. Engage the Always Active checkbox. You should see the LED of the LFO Module light up.





14.3 Automation ID Management

In addition to the Control ID (discussed in section [↑4.1, Recalling Snapshots](#)), Panel controls in REAKTOR also have an Automation ID. The purpose of the Automation ID becomes apparent when REAKTOR is running as a plug-in in a host sequencer. In that case the Automation IDs serve to communicate to the host which parameters can be automated. This section takes a closer look at Automation IDs when REAKTOR is running as a VST, AU, or RTAS plug-in.

14.3.1 Global Automation IDs

All programs that are used within a host as a plug-in have to conform to a standardized communication protocol like VST or AU. The protocol defines in which way both applications interact, for example in which way MIDI messages or audio signals are interchanged. One part of those protocols deals with the control of a plug-in's sound parameters from outside. Therefore, a plug-in reports a list of parameters to the host application. The host can then send information to the plug-in to set one of the list's parameters to a specific value. This is what is usually understood as automation. Unlike most plug-in applications, REAKTOR provides full control over the label and the ID number which are sent to the host sequencer.

As already mentioned, REAKTOR (as a plug-in) reports one list of automatable parameters to the host application. This list is the Ensemble's global parameter list; each parameter listed here is assigned a so-called Automation ID and can be automated from outside. The

global list comprises subsequent local lists of the Instruments contained in the Ensemble. How these lists (belonging to the individual Instruments) are integrated into the global list (of the Ensemble) is handled in the Connections page of the Instrument Properties.

How the Global Automation ID List Is Put Together

The Base ID (see screenshot below) represents the Automation ID of the Instrument's first parameter within the global automation list. Beginning with that Automation ID a user-definable number of IDs is reserved for that Instrument within the global list. The number of Automation IDs reserved for the Instrument within the global list can be set with the Instrument's Max Id edit field. Below (where you can also see the Max Id edit field) the Instrument has been placed at the beginning of the Ensemble's global list. This is reflected by the fact that the Base Id edit field has the value "0". The Instrument's list of parameters covers 120 the subsequent entries within the global list, although only 105 of these entries have been assigned to controls. The way those entries are used is specified by the Instrument's local parameter list which is explained later.



Fig. 14.13 The Base Id and Max Id in use displays as well as the Max Id edit field are found in the Automation area of the Instrument's Connections page. These fields give you information about the organization of the Instrument's list of Automation IDs.

For example, in the above figure you see that an Instrument's Automation IDs start with the Base Id "0" and go up to the Max Id, "120". If a second Instrument has been or is inserted into the Ensemble, its Base Id starts with the Max Id of the already inserted Instrument: "120". This is shown below. If a third Instrument is inserted into the Ensemble, its Base Id is the sum of the second Instrument's Base Id and its Max Id. In the example at hand it would be "220" (120 + 200 = 220).



Fig. 14.14 The Base Id of the second Instrument is equal to the sum Base Id and Max Id of the previous Instrument.

Changing the Order of Instruments in the Global List

The global list comprises sub-lists of Automation IDs, each belonging to an Instrument. You can change the order of the Instrument's sub-lists within the global list by using the IDs drop-down menu.

- To move an Instrument's automation list upwards (towards "0") within the global list, select the *Instrument Up* menu entry from the IDs drop-down menu. All of the Instrument's Automation IDs are then moved upward together.
- To move an Instrument's automation list downwards (away from "0") within the global list, select the *Instrument Down* menu entry from the IDs drop-down menu. All of the Instrument's Automation IDs are then moved downward together.

! Remember that the communication between host and plug-in is based on the Automation IDs. If the order of Instruments is changed the order of the single entries (within the global list) is also altered. The host, however, will not register this fact; it continues to send values to the parameter at a given Automation ID although the parameter at this "address" has changed. Changes to the Automation IDs almost always ruin existing automation connections. Therefore, think twice before moving Automation IDs around within your Ensemble.

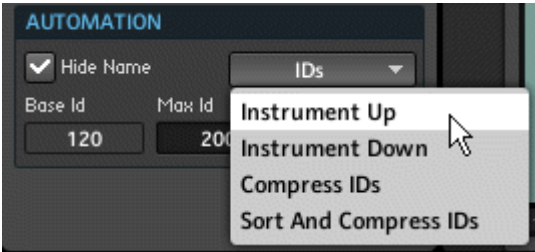


Fig. 14.15 Using the IDs drop-down menu you can move an Instrument's automation list up or down in the global list.

14.3.2 Automation IDs within an Instrument

Each Panel control has its own Automation ID within the Instrument's automation list. These Automation IDs are counted starting from “0”. You see this number in the Id edit field, shown below. When the Instrument's Automation ID is placed in the context of the global list, that Panel control's global Automation ID is counted starting from the Instrument's Base ID. The Instrument, to which the Panel control in below figure belongs, has Base ID “220”. Hence the global Automation ID of the control with an Automation ID of “10” is “230” ($220 + 10 = 230$).

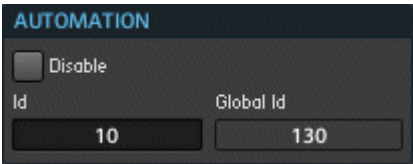


Fig. 14.16 You can manage the Automation IDs of individual Panel controls in the Automation area of their Connections page.

- To change the Automation ID of a Panel control, go to its Connections page and enter the new Automation ID into the Id edit field, shown above.



Fig. 14.17 The Connect button

- If the Automation ID is changed manually, it can happen that the new place within the local list is already occupied by another Panel control. In this case the Automation IDs of the two controls are exchanged. This means that changing an ID manually might change not only the currently edited Panel control's Automation ID but also that of another Panel control.

! Remember that the communication between host and plug-in is based on the Automation IDs. Changes to the Automation IDs almost always ruin existing automation connections.

- To keep a Panel control from appearing the list of automatable parameters that is sent to the host application, engage the **Disable** checkbox in the Panel control's Connections page. This checkbox is also shown in the above screenshot. However, that Panel control's Automation ID would still keep its place in the local and global automation lists within REAKTOR. You might want to use this feature when a control automates another control within REAKTOR; automation chains or feedback loops can cause technical or logical problems.

Max Id in Use

In the Automation area of an Instrument's Connections page you will find the **Max Id in use** display (shown below). This display shows the greatest Automation ID value that is in use by a Panel control within that Instrument. Panel controls with an Automation ID greater than the **Max ID** value do not show up in the global Automation ID list. These controls could therefore not be automated from a host application.



Fig. 14.18 The **Max Id in use** display shows the greatest Automation ID that is in use by a Panel control belonging to that Instrument. If this number is higher than the **Max ID** value, then there are Automation IDs which do not get placed in the global list.

Sorting and Compressing Automation IDs

The list of local IDs can become very chaotic while an Instrument is built within REAKTOR. Each time a new Panel control is created, a new ID is assigned to it. If old Panel controls are deleted, gaps are created within an Instrument's ID list. You can re-assign the IDs manually by entering a new value into the **Id** edit field (as described above), but this

might be time-consuming for large Instruments. Therefore you can also use the *Compress IDs* and *Sort and Compress IDs* menu entries that can be found in the Instrument's Connections page (shown below).

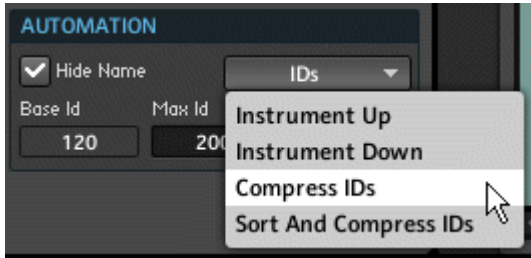


Fig. 14.19 Use the IDs drop-down menu in the Automation area of the Instrument's Connections page to find the *Compress IDs* and *Sort and Compress IDs* menu entries.

Compressing Automation IDs results in filling all gaps within the Instruments automation list. For instance, if the IDs “0” to “5” and “8” to “10” are used by Panel controls while the list's 6th and 7th entries are empty, compressing the list shifts the Automation IDs “8”, “9”, and “10” to the entries “6”, “7”, and “8”, respectively.

- To compress the Instrument's automation list, choose the *Compress IDs* menu entry in the Instrument's Connections page (shown above)
- Sorting Automation IDs places the Automation IDs of Panel controls that are located within one Macro at subsequent entries. Additionally, within the Macro the entries are ordered alphabetically according to the Labels of the Panel controls.
- To sort and compress the Instrument's automation list, choose the *Sort and Compress IDs* menu entry in the Instrument's Connections page (shown above)



Both actions completely re-organize the Instrument's local automation list. Therefore be very careful when choosing to do either of them. One reason is that a parameter can become non-automatable if its ID becomes greater than the Instrument's Max ID value after sorting. Even more important, any existing automation based on the old Automation IDs will be broken, similar to the process described regarding the *Instrument Up* and *Instrument Down* menu entries.

14.3.3 Automatable Modules in REAKTOR

Only a few number of REAKTOR's Modules can be automated. This goes along with the fact that only some Modules are Panel controls, that is, they are displayed on the Instrument Panel and serve as controls for your Instrument. The figure below shows which Modules can be automated. They can all be found in *Built-In Modules > Panel* submenu.

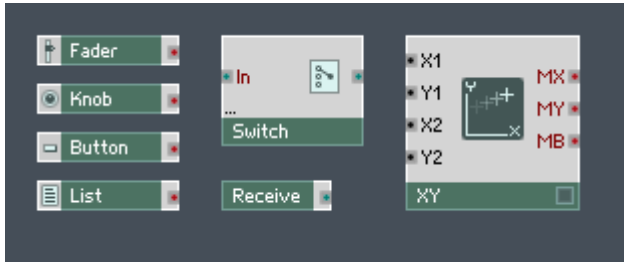


Fig. 14.20 The Modules shown here can be automated by a host application.

Except for the XY Module all of the Modules displayed react similar to automation messages. Like MIDI, automation messages from a host have a fixed range of values. This range is automatically mapped onto the Module's internal range. For instance, imagine a Fader or a Knob Module with a range [0 ... 15]. To set this Panel control to its highest possible value, the host sends the highest possible automation message – and not the value “15”.

If the internal range is smaller than the external range, several external values are mapped onto one internal value. This is also true if the Module can only be set to two states, as is the case for the Button Module. All automation messages below the middle automation value set the Button module to its default state; all messages above the middle invert its state. If the Module provides more than two states and not a continuous range (this is the case for the Switch, Receive, and List Modules), a similar “quantization” takes place. The continuous automation messages are divided into a given number of sub-ranges according to the Module. For example, if a Switch Module provides 3 buttons, low automation values will set it to the first state, middle automation values will change it to the second state, and high values will activate the third state.

14.4 Advanced REAKTOR Preferences

The REAKTOR Preferences Dialog

The REAKTOR Preferences dialog houses a number of preference options that you can set as desired. The Preferences dialog consists of three tabs: Options, CPU Usage, and Directories. This section describes the Options tab and Directories tab. The CPU Usage tab is discussed in subsection [↑9.3.2, Changing the Number of Polyphonic and Unison Voices](#).

- To open the Preferences dialog, press the Menu button and chose the *File > Preferences* menu entry (you can also access the Preferences dialog window using the keyboard shortcut Ctrl+, if you are on Windows (Cmd+, if you are on Mac OS X)).

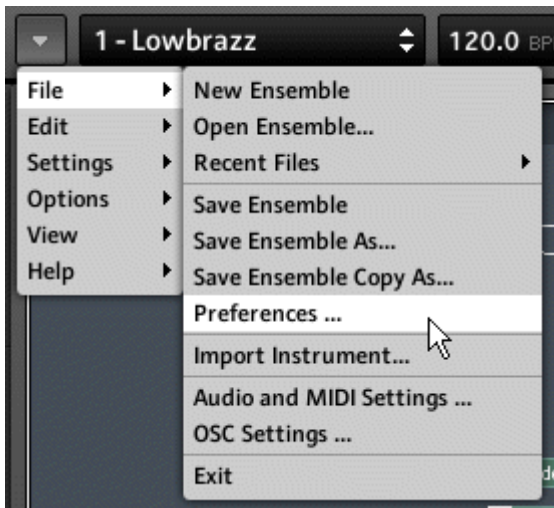


Fig. 14.1 Opening the Preferences dialog window

14.4.1 The Options Tab

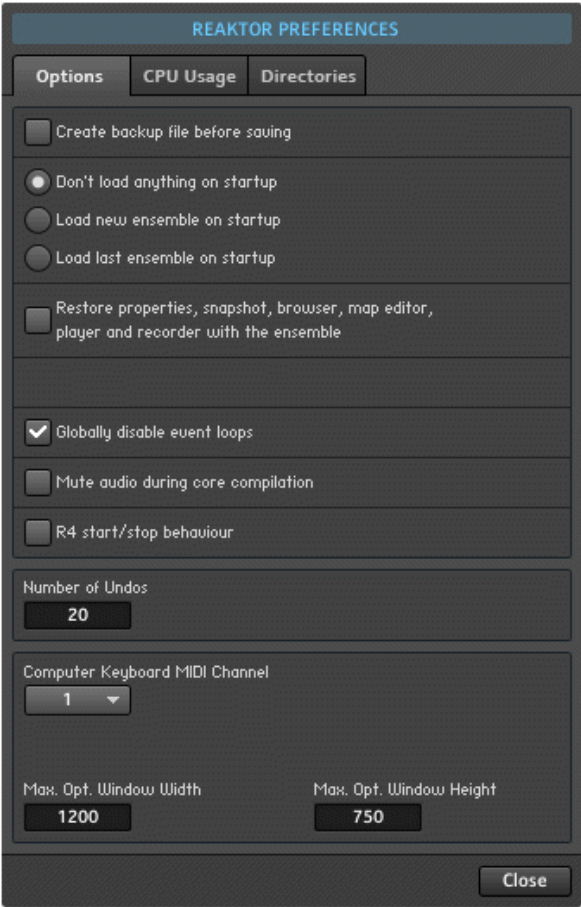


Fig. 14.21 The Options tab holds a number of preferences that globally define the behavior of REAKTOR in specific cases.

The preferences in the Option tab for REAKTOR Standalone mode are the following. Please refer to section [↑3.6, REAKTOR as a Plug-In](#) for additional preferences tied to REAKTOR as a plug-in.

- **Create backup file before saving:** When this checkbox is engaged, REAKTOR will create a backup file of your Ensemble titled “Backup of *Ensemble name*” when you execute the Save Ensemble command.
- **Startup:** The preference in this radio button selector determines what Ensemble, if any, will REAKTOR load when it starts up. The three radio buttons give the following options:
 - Don’t load anything on startup
 - Load new Ensemble on startup
 - Load last Ensemble on startup
- **Restore Properties, Snapshot, Browser, Map Editor, Player and Recorder with the Ensemble:** this option tells REAKTOR to restore an Ensemble's (graphical user interface) GUI state just as it was when the Ensemble was last saved.
- **Globally Disable Event Loops:** this prevents Event signal feedback loops from occurring in Ensembles. Please refer to subsection [↑14.1.2, REAKTOR's Debugging Tools](#) for more information on this feature.
- **Mute Audio During Core Compilation:** when enabled, this option turns off all audio processing during the period of time that it takes to compile a Core Object. This makes compilation faster.
- **R4 start/stop behavior:** Engaging this checkbox activates causes the keyboard shortcuts for the REAKTOR Clock to behave as in REAKTOR 4:
 - Subsequently pressing Space three times causes the following Clock behavior, respectively: Start/Restart/Continue (in REAKTOR 5 the action sequence is Start/Pause/Continue)
 - Subsequently pressing Ctrl+Space in Windows (Cmd+Space in Mac OS X) causes the following Clock behavior: Pause/Stop/Reset (Restart to Song Position zero)
- **Number of Undos:** this edit field lets you set the maximum number of undos that can be performed in a row. Setting the Number of Undos edit field to “20”, for example, enables you to undo the last 20 edits you've made by issuing 20 undo commands in a row. Enter “0” to disable the undo function.



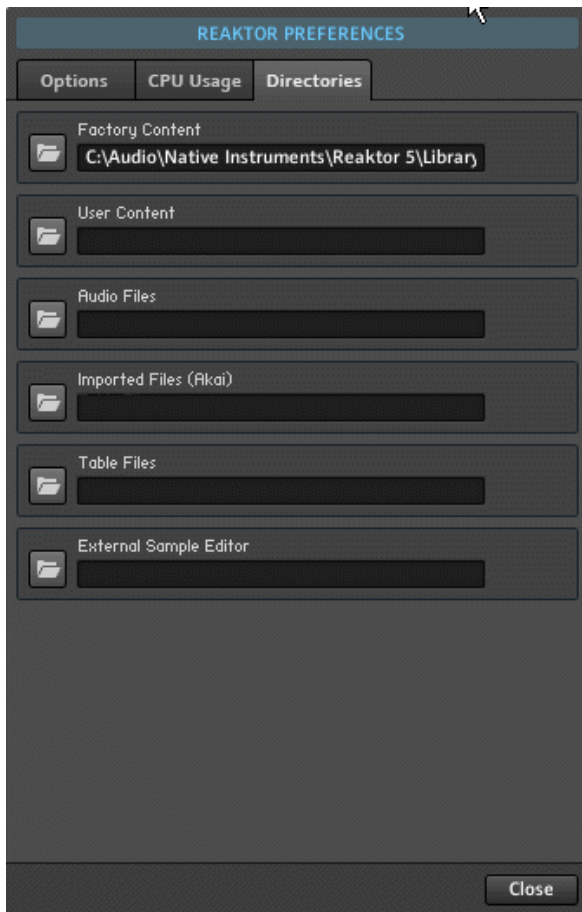
If you work on Ensembles containing large audio files, you will need a lot of computer memory to continue to use the undo function. If you do run out of memory, try reducing the Number of Undos.

- **Computer Keyboard MIDI Channel:** drop-down menu specifies the MIDI channel for MIDI Notes played at the computer keyboard. This is set to “1” by default because REAKTOR Instruments are set to receive MIDI messages on channel 1 by default. If, however, you want to use the keyboard to play an Instrument that receives MIDI on a different channel (for example, “2”), you would set the Computer Keyboard MIDI Channel drop-down menu to this channel (for this example, “2”).

14.4.2 Directories Tab (Setting File Paths)

It is very important that REAKTOR knows where your Factory Library, your User Library and your Sample Library are on your computer or external hard drive. Here you will be guided through telling REAKTOR where the paths to these libraries are located.

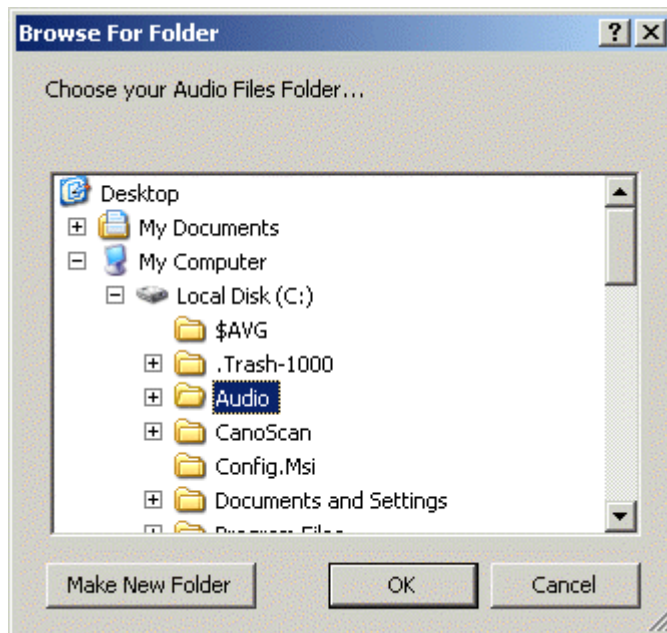
1. Press the **Directories** tab to access the Directories page.



2. The Factory Content edit field shows where your Factory Library has been installed, but we are more interested in setting up the path for your REAKTOR User Library. Click on the Folder button next to the User Content label.
3. You will be asked to “Choose your private Library Folder...”. In this example, the User Library is located under “/My Documents/Native Instruments/Reaktor 5/” (this is a good practice for easily managing your REAKTOR files for years to come). Click “OK” to confirm.



4. Next click on the folder button next to Audio Files label to set up the path to your sample files.
5. You will be asked to “Choose your Audio Files Folder...”. In this example, the audio files are located in the folder “C:/Audio/”



Once you have finished, your REAKTOR Preferences should look something like this:

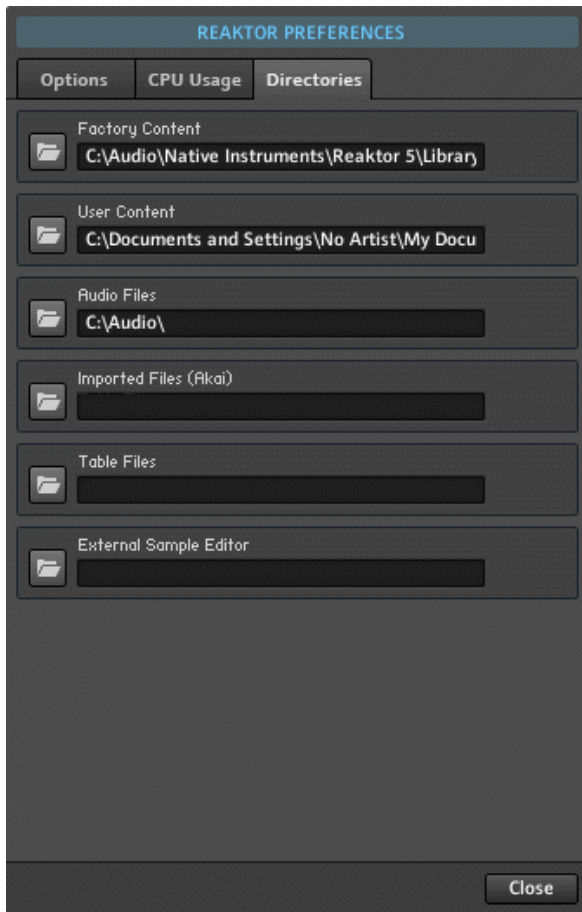


Fig. 14.2 Finished Preferences

- Click on the Close button in the bottom right-hand corner to close Preferences dialog window.