



REAKTOR BLOCKS

Table of Contents

1. Disclaimer	1
2. Welcome to REAKTOR Blocks	2
3. Learning How to Patch	3
3.1. Info Hints	3
3.2. Additional Resources	4
4. Overview	5
5. Basic Use	7
5.1. Loading Racks from a Blocks Product	7
5.2. Loading Racks from the Hard Drive	8
5.3. Switching between Racks	9
5.4. Saving Racks	10
5.5. Creating a New Rack	11
5.6. View Modes	12
5.7. Blocks Missing Dialog	12
5.8. Advanced Features	13
6. Adding and Arranging Blocks	15
6.1. Blocks Library	15
6.2. Adding Blocks to a Rack	16
6.3. Using the Search Box	17
6.4. Arranging Blocks in Panel View	18
6.5. Arranging Blocks in Structure View	19
7. Patching in Racks	21
7.1. Connecting Blocks	21
7.1.1. Making Connections	21
7.1.2. Removing Connections	22
7.1.3. Wire Colors	23
7.2. Panel Controls	24
7.3. Modulation Routing	26
7.3.1. Modulation Depth Sliders	26
7.3.2. Example Routing	27
8. Connections and Signals	28
8.1. Input and Output Types	28
8.2. General Inputs and Outputs	28
8.3. Modulation	29
8.4. Pitch	30
8.5. Gate	30
8.6. Reset	31
8.7. Play	31
8.8. Sync	32
8.9. Pluck	32

8.10. Feedback Connections	33
9. Integrating Hardware Synthesizers with Control Voltages	34
9.1. Routing Control Voltages to External Hardware	34
9.2. Using the Pitch CV Out Block	35
10. User Blocks	40
10.1. Installing User Blocks	40
10.2. Updating User Blocks	41
10.3. Loading User Blocks in Racks	42
10.4. Building Your Own Blocks	43
11. Blocks Reference	44
11.1. Bento Box	44
11.1.1. 4 Mods (SEQ)	44
11.1.2. 8 Steps (SEQ)	46
11.1.3. ADSR Envelope (MOD)	48
11.1.4. CV Processor (PRO)	49
11.1.5. LFO (MOD)	51
11.1.6. Mix (MIX)	52
11.1.7. Oscillator (OSC)	53
11.1.8. Sample & Hold (PRO)	55
11.1.9. SVF (FLT)	56
11.1.10. VCA (AMP)	57
11.1.11. XFade (MIX)	59
11.2. Boutique	59
11.2.1. Dual SKF (FLT)	60
11.2.2. Multiwave OSC (OSC)	61
11.2.3. OSC 5 (OSC)	63
11.3. Digilog	64
11.3.1. Clock Divider (PRO)	64
11.3.2. Quantizer (PRO)	66
11.4. DRIVER	67
11.4.1. DRIVER (EFX)	67
11.5. Kodiak	69
11.5.1. Curve Sequencer (SEQ)	69
11.5.2. Duality Osc (OSC)	71
11.5.3. Flip Gen (OSC)	74
11.5.4. Morph Filter (FLT)	75
11.5.5. Shift Sequencer (SEQ)	78
11.6. Modern	81
11.6.1. Comb Filter (FLT)	82
11.6.2. Paul Filter (FLT)	83
11.7. MONARK	84
11.7.1. MONARK ADS ENV (MOD)	85
11.7.2. MONARK Filter (FLT)	86
11.7.3. MONARK Oscillator (OSC)	87
11.8. ROUNDS	88

11.8.1. ROUNDS Delay (EFX)	89
11.8.2. ROUNDS LFO (MOD)	90
11.8.3. ROUNDS Reverb (EFX)	91
11.9. Util	92
11.9.1. Clock (INT)	93
11.9.2. CV Mix (MIX)	94
11.9.3. Gates & Trigs (INT)	95
11.9.4. Level Mono (MIX)	96
11.9.5. Level Stereo (MIX)	97
11.9.6. Macro Knobs (INT)	98
11.9.7. Macro Switches (INT)	99
11.9.8. MIDI Out (INT)	101
11.9.9. Mix 4 (MIX)	102
11.9.10. Note In (INT)	103
11.9.11. Pitch CV Out (INT)	104
11.9.12. Scope (AUX)	106
11.9.13. Trig In (INT)	107
11.10. West Coast	108
11.10.1. CFG (MOD)	108
11.10.2. DWG (OSC)	111
11.10.3. LPG (FLT)	114
11.10.4. XYS (SEQ)	115
12. Using Blocks in Ensembles	118
12.1. Saving a Rack as Ensemble	118
12.2. Factory Ensembles	119
12.3. The New Ensemble	119
12.4. Play Mode vs. Edit Mode	120
12.5. Navigating between Panel and Structure	120
12.6. Saving Patches and Parameter Settings	121
12.6.1. Saving and Loading Ensembles in the Stand-Alone Application	122
12.6.2. Saving and Loading Ensembles in the Plug-in	122
12.6.3. Storing and Recalling Snapshots	123
12.7. Adding Blocks to an Ensemble	125
12.8. Organizing and Arranging Blocks in an Ensemble	125

1. DISCLAIMER

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.

“Native Instruments”, “NI” and associated logos are (registered) trademarks of Native Instruments GmbH.

Mac, macOS, GarageBand, Logic and iTunes are registered trademarks of Apple Inc., registered in the U.S. and other countries.

Windows and DirectSound are registered trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are the property of their respective owners and use of them does not imply any affiliation with or endorsement by them.

Document authored by: Jan Ola Korte

Software version: REAKTOR 6.4 (07/2020)

2. WELCOME TO REAKTOR BLOCKS

REAKTOR Blocks brings the experience of patching an analog modular synthesizer to REAKTOR 6. Unlike a regular synthesizer, a modular synthesizer does not have a predefined architecture. You can freely arrange and connect the basic building blocks of a synthesizer, or modules, to create patches based on a variety of different synthesis methods.

Blocks patches, called Racks, can be created without prior building experience in REAKTOR. Universal connectivity between the individual modules, called Blocks, allows for any connection to be made, with predictable results. Internally, Blocks are built on a sophisticated framework that supports users who want to modify or build their own Blocks.

By combining this framework with an intuitive user interface, REAKTOR Blocks benefits musicians and builders alike:

- The Rack format allows sound designers and musicians without building experience to start patching in REAKTOR and create their own instruments and effects.
- REAKTOR Primary builders can make a smooth transition to Core by modifying the underlying Core Cells or using the Core Macro Library to build User Blocks.
- REAKTOR Core builders can use the framework and user interface templates to create User Blocks based on their own custom Core Cells.

REAKTOR is fortunate to be supported by a large community of experienced users who actively share their creations in the [REAKTOR User Library](#) on our website—a great source of inspiration, both musically and technologically. We invite you to join this community and are looking forward to seeing and hearing your patches and Blocks!



To support builders in creating their own Blocks, a building template including additional documentation is available in the REAKTOR User Library. For more information, see [Building Your Own Blocks](#).

3. LEARNING HOW TO PATCH

This manual provides an [overview](#) of the Rack and its [basic use](#) along with information about [adding Blocks](#) and [patching with them](#). It also includes information about advanced topics: [controlling hardware synthesizers](#), [using Blocks in Ensembles](#), and working with [User Blocks](#). Additionally, you will find detailed descriptions of the [connections and signals](#), as well as a [reference of all Blocks](#) included in REAKTOR 6. If you want to learn about Blocks while patching, you can use the [Info Hints](#) in REAKTOR.

To get started, you can follow the [Quick Start Guide](#) online, which shows you how to build your first synthesizer using Blocks in a Rack. If you want to take the next steps, we have compiled a list of [additional resources](#) that you can use to learn about patching techniques and apply them using Blocks.

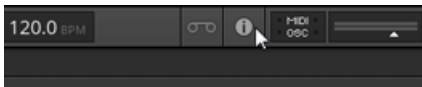


This document does not require prior knowledge of REAKTOR. If you want to learn more about the underlying concepts and features, you can find extensive documentation on the [REAKTOR Downloads](#) page.

3.1. Info Hints

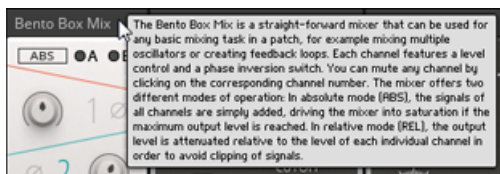
Blocks features comprehensive information about the Blocks as well as their controls, inputs, and outputs in the REAKTOR application. You can view this information in the form of tooltips (called Info Hints in REAKTOR).

- To show the Info Hints in REAKTOR, click Show Info Hints in the REAKTOR header.

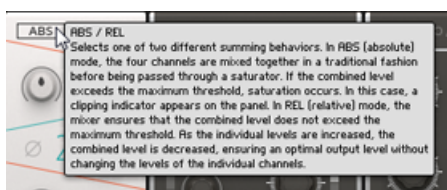


Info Hints are available for each entire Block, the individual controls, as well as the inputs and outputs:

- To view the Info Hint for a Block, place the mouse over its header:



- To view the Info Hint for a parameter, input, or output, place the mouse over the element in the Panel view:



- You can also view the Info Hint for an input or output in the Structure view:



3.2. Additional Resources

If you are new to modular synthesis, or want to expand your knowledge about patching techniques and the technology involved, the following online resources can be of great help.



The websites linked below are owned and operated by third parties. The links are provided for your information and convenience only. Native Instruments has no control over the contents of any of the linked websites and is not responsible for these websites or their content or availability.

- [Introduction to Modular synthesis in Reaktor](#): This tutorial video series by Salamander Anagram for ADSR shows you how to get started with patching in Racks.
- [Patchení s Nikol \(Patching with Nikol\)](#): A video tutorial series published by Eurorack manufacturer Bastl Instruments, explaining both basic concepts and advanced patching techniques that can be applied with any modular synthesizer.
- [Sound On Sound Synthesizer Secrets](#): This collection of articles by Gordon Reid covers many different synthesis techniques, explaining how they can be used to create classic synthesizer sounds or mimic acoustical instruments.
- [Rob Hordijk's Synthesis Workshops](#): Rob Hordijk's Synthesis Workshops are among the most comprehensive resources about sound synthesis available online. The articles and tutorials cover synthesizer theory, history, and practice in detail.
- [Basics of Sounds Synthesis](#): Another helpful resource from Rob Hordijk, covering some of the more basic topics from his Synthesis Workshops in a concise manner. This is especially useful as an introductory document for beginners.
- [Advanced Programming Techniques for Modular Synthesizers](#): This online book by James J. Clark explains many advanced synthesis methods and how they can be patched with a modular synthesizer.

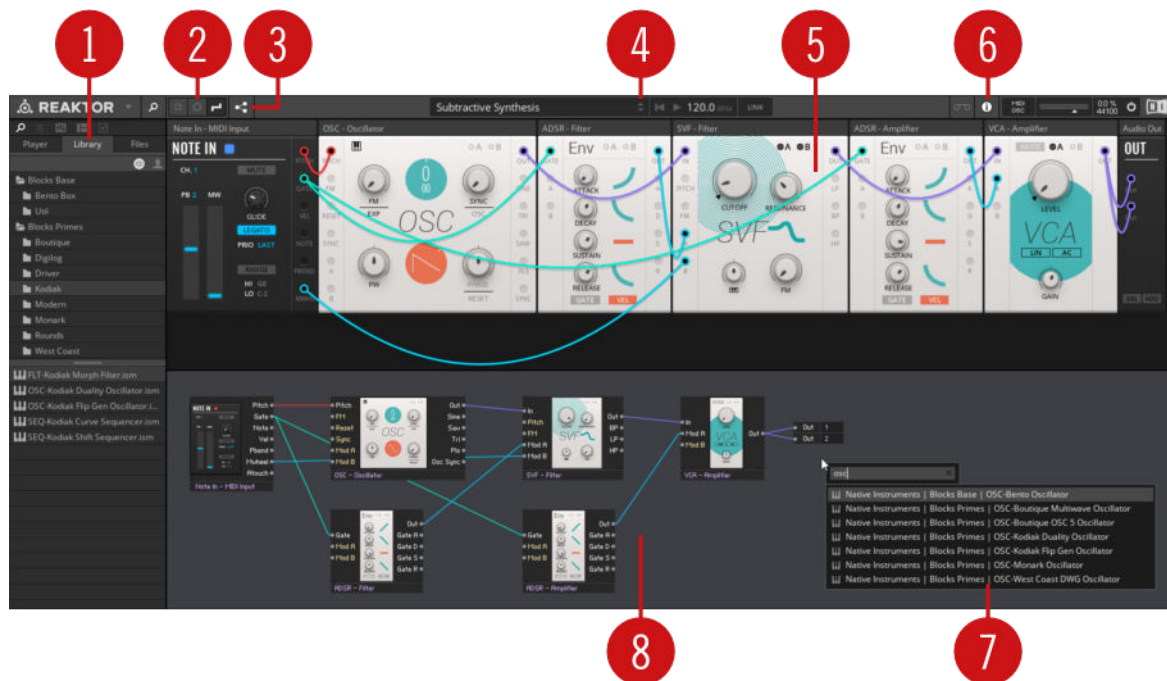
4. OVERVIEW

In a Rack, you use wires to connect any number of individual Blocks, or modules, to form a patch. The structure of this patch constitutes a musical instrument, a sequencer, an effect, or any combination thereof. Each Block takes on a specific role in the structure.

This does not mean each Block always performs the same function. While they all have an intended purpose, Blocks can be used in various interesting ways. There are no wrong connections in Racks, and unusual patches often lead to surprising results.

You can arrange, connect, and control Blocks directly in the **Panel view**. The patch is mirrored in the **Structure view**, allowing you to choose your preferred patching workflow.

The following overview shows you how a Rack is displayed in REAKTOR and where you can find the most important controls.



Overview of a Rack in REAKTOR

(1) **Library:** This tab in the Browser shows all installed Blocks. They are organized in folders. The contents of the selected folder are shown in the lower section of the Browser. You can drag and drop the Blocks from the lower section into the Rack to build your patch. For more information, see [Adding and Arranging Blocks](#).

(2) **View modes:** These three buttons control the visibility of ports and wires in the Rack. You can choose Compact or Ports View to hide the wires and show all panel elements more clearly. For more information, see [View Modes](#).

(3) **Show Structure:** This button shows or hides the additional Structure view (8). This is useful for creating and managing large patches. The Structure view shows all wires even when Compact View or Ports View are selected. To show or hide the Structure view using the keyboard, press [command] + [4] (macOS) or [Control] + [4] (Windows).

(4) **Rack switching:** The arrow buttons in the Rack menu switch between Racks in the selected category. For more information, see [Switching between Racks](#).

(5) **Panel view:** Shows all Blocks in the Rack. You can change the controls on each Block to adjust its parameters. Inputs are located on the left side of a Block, while outputs are located on the right side. You can connect Blocks by clicking on an output and dragging a wire to an input. You can remove a wire by double-clicking on the connected input. For more information, see [Patching in Racks](#).

(6) **Info Hints:** Enables or disables tooltips for all Blocks and their controls. For more information, see [Info Hints](#).

(7) **Search box:** Press the Enter key on your computer keyboard to call up the search box. You can find Blocks by entering their function, family, or name. The search box can be used both in Panel and Structure views. For more information, see [Using the Search Box](#).

(8) **Structure view:** Shows all Blocks in the Rack. Inputs are located on the left side of a Block, while outputs are located on the right side. You can connect Blocks by clicking on an output and dragging a wire to an input. You can click on a wire to select and press [Delete] or [Backspace] to delete it. For more information, see [Patching in Racks](#).

5. BASIC USE

Racks include all settings including parameter values, the arrangement of Blocks, and the connections made between them. They offer a streamlined patching experience including different View modes, but also advanced REAKTOR features like Panel Sets and remote control via MIDI and OSC.

The Blocks themselves are only referenced in the Rack and are recalled from the respective Blocks product when a Rack is loaded. This allows for small file sizes, and makes it possible to fully save and recall Racks with your host sequencer project (host chunk saving).

Racks are saved as *.nksr files. Since they are compatible with Native Kontrol Standard (NKS), they can be loaded as presets in KOMPLETE KONTROL and MASCHINE.

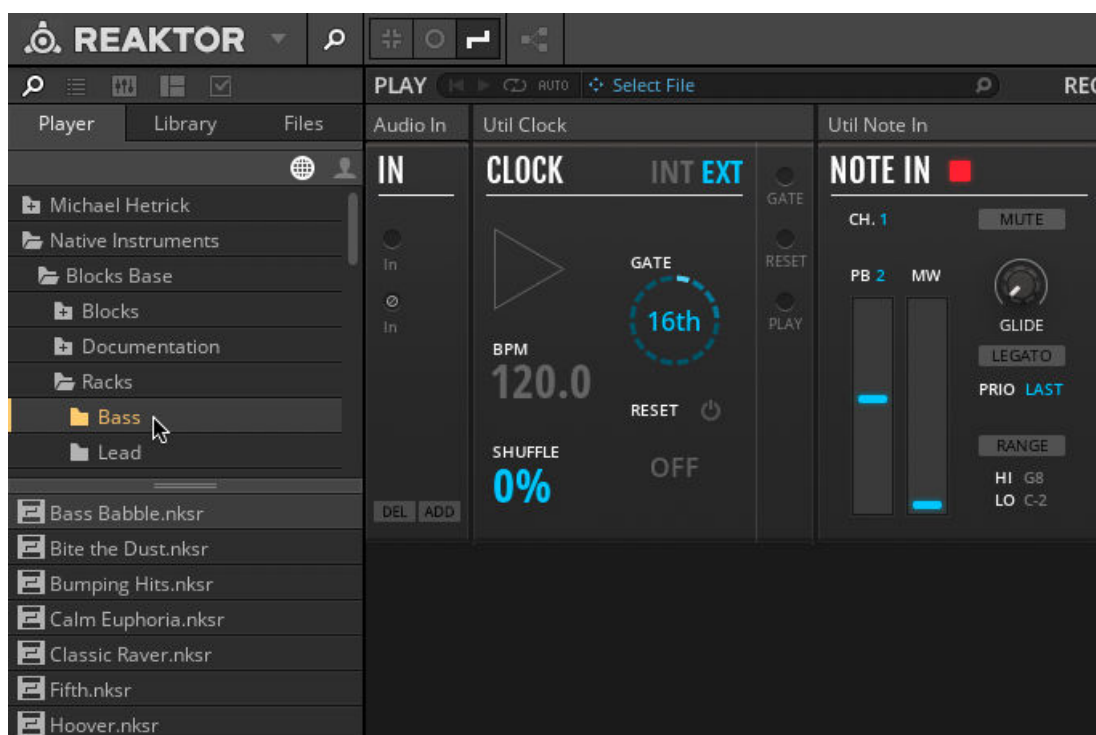
The following sections show you how to load, save, and switch between Racks. Additionally, you can find information about the [View modes](#) and an outline of the [advanced features](#).

5.1. Loading Racks from a Blocks Product

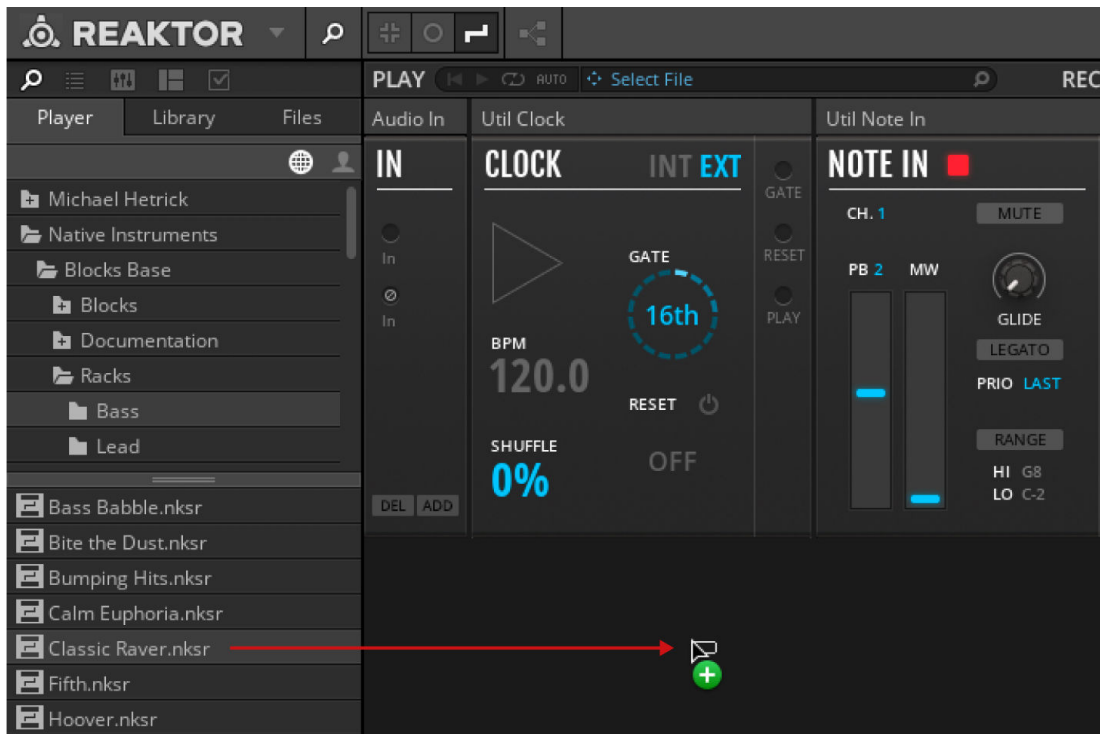
Each Blocks product includes pre-built Racks as factory content. You can load these Racks from the Player tab in the Browser.

To load a Rack included in a Blocks product:

1. Open the folder of the Blocks product in the Browser's **Player** tab and select a category to show the Racks in the lower section.



2. Drag and drop a Rack from the lower section of the **Player** tab to the main area (alternatively, you can double-click it).



The Rack is loaded and ready to be used.



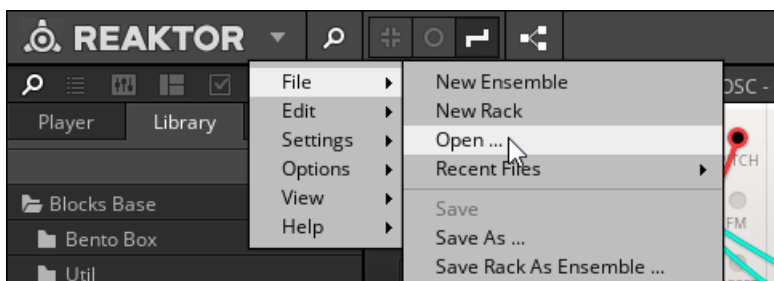
If REAKTOR cannot find all required Blocks when loading a Rack, it will show the Missing Blocks dialog that you can use to relocate them. For more information, see [Blocks Missing Dialog](#).

5.2. Loading Racks from the Hard Drive

You can load Racks from your hard drive, for example, Racks you have saved or ones you have downloaded from the REAKTOR User Library.

To open a Rack saved on the hard drive:

1. Click on the drop-down menu next to the REAKTOR logo and select **Open** in the **File** sub-menu.



2. In the file dialog, browse to the Rack file you saved and select it.
3. Click **Open** to load the selected Rack.

The Rack opens in REAKTOR and is ready to be used.



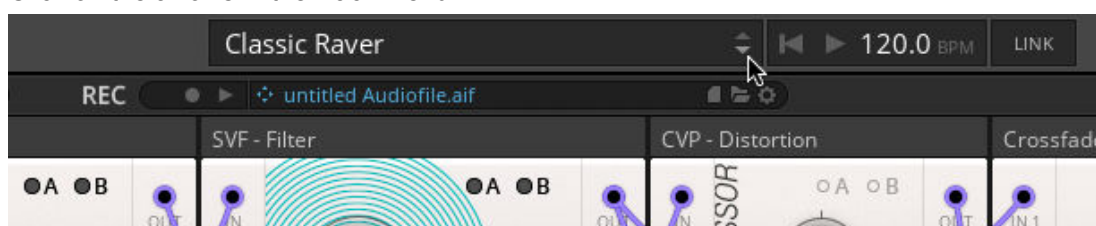
If REAKTOR cannot find all required Blocks when loading a Rack, it will show the Missing Blocks dialog that you can use to relocate them. For more information, see [Blocks Missing Dialog](#).

5.3. Switching between Racks

Once you have loaded a Rack, you can switch between Racks in the same category using the Rack menu in the toolbar or the Preset Browser in the side pane. This is possible with Racks both in the Factory and User Library.

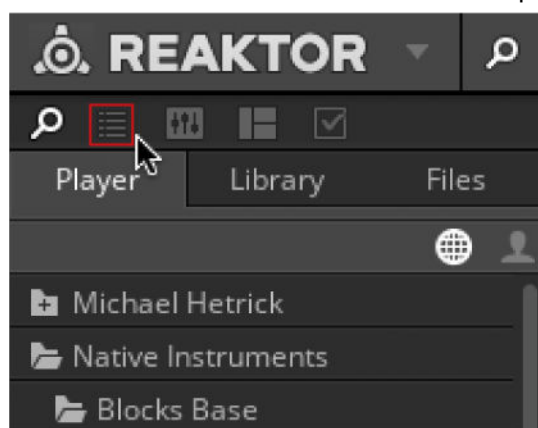
To switch between Racks using the Rack menu:

- Click on the arrows in the Rack menu.

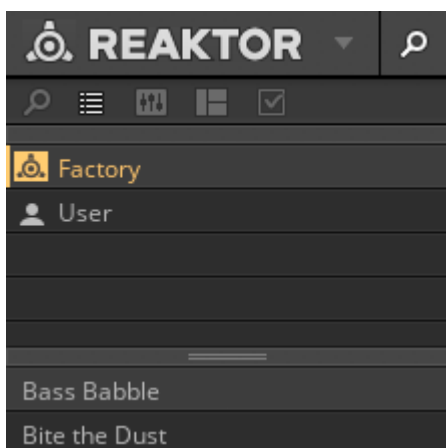


To switch between Racks using the Preset Browser:

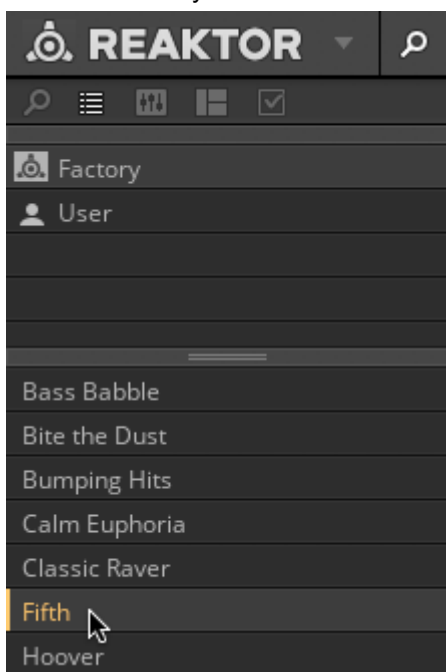
1. Click the Preset Browser button in the side pane.



2. Click either **Factory** or **User** in the Preset Browser to show the respective Racks in the lower section.



3. Click on an entry in the lower section.



The corresponding Rack is loaded.

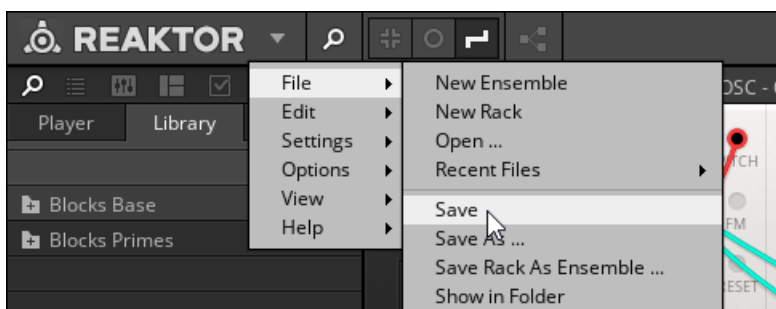
5.4. Saving Racks

When you have created a sound you like, you can save your Rack in a file that you can open at any time, or share with others.



You can use the [REAKTOR User Library](#) to share your Racks and engage with other users.

1. Click on the drop-down menu next to the REAKTOR logo and select **Save** in the **File** submenu.



2. In the file dialog, browse to the location on your hard drive you want to save the Rack in.
 - Using the default Racks folder saves the Rack in the User Library. This makes the Rack available as a preset in the KOMPLETTE KONTROL and MASCHINE Browser. It also enables convenient switching between Racks in the same category. For more information, see [Switching between Racks](#).
 - Alternatively, you can choose any location on your hard drive. For information about loading Racks from the hard drive, see [Loading Racks from the Hard Drive](#).
3. Type in a name for your Rack.
4. Click **Save** to save the Rack.

The Rack is saved in the chosen location on your hard drive.

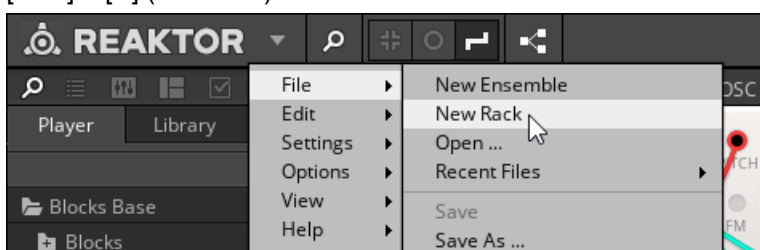


When using REAKTOR as a plug-in, all settings including the arrangement of Blocks and the connections made between them are also saved and recalled with your host sequencer project (host chunk saving).

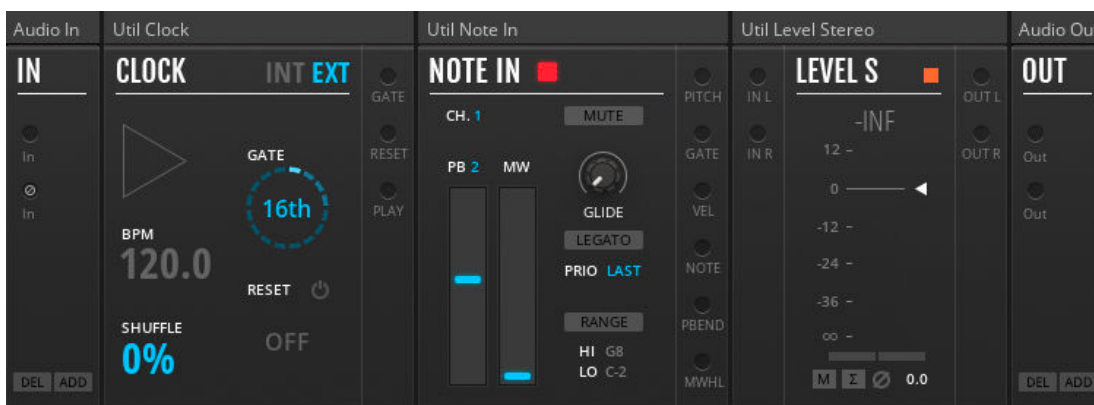
5.5. Creating a New Rack

If you want to create a patch from scratch, you need to open a new Rack.

- Click on the drop-down menu next to the REAKTOR logo and select **New Rack** in the **File** submenu. Alternatively, you can press [Command] + [Shift] + [N] (macOS) or [Control] + [Shift] + [N] (Windows).



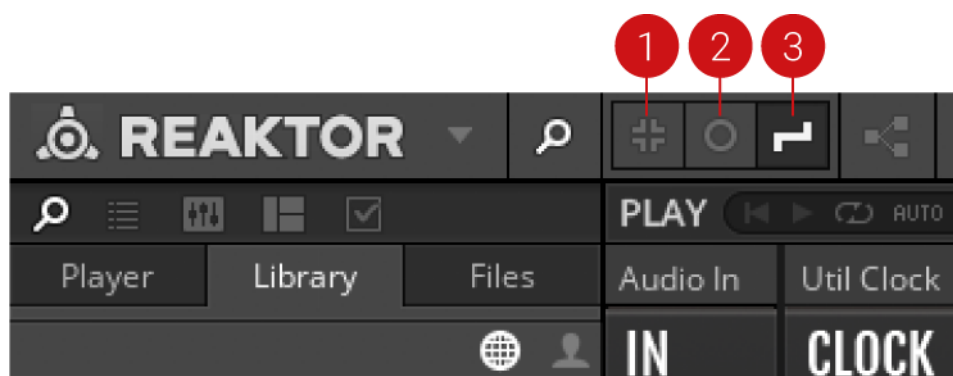
A new Rack opens and is ready to be used. It includes a number of modules that provide the infrastructure for your patch, including inputs and outputs.



To learn about adding and arranging Blocks in the new Rack, see [Adding and Arranging Blocks](#).

5.6. View Modes

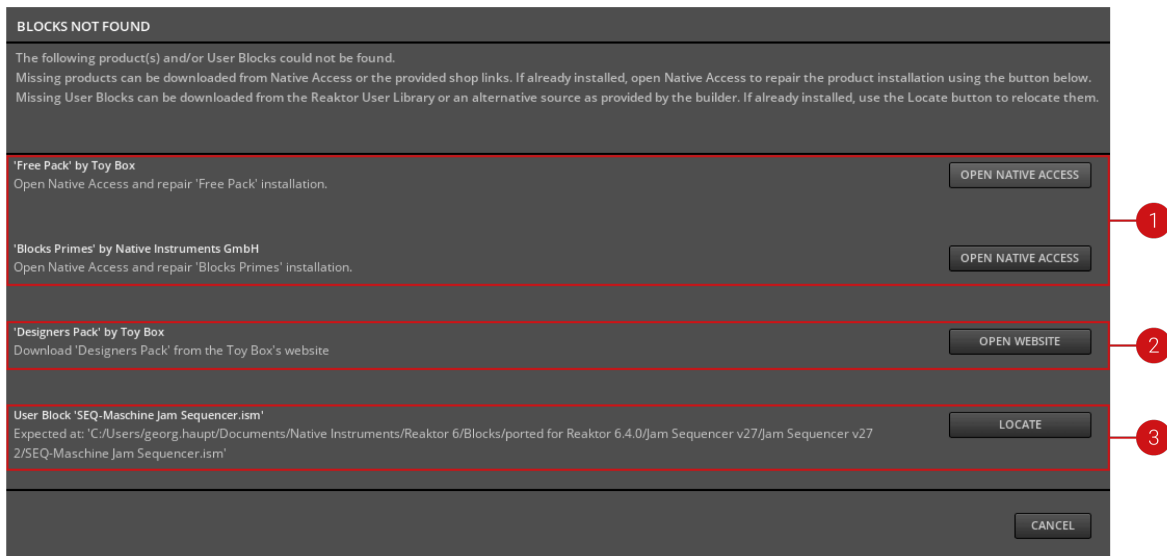
The View modes control the visibility of ports and wires in the Rack's Panel view.



1. **Compact View:** All ports and wires are hidden. You can focus on tweaking the controls on the panel. To select this View mode using the keyboard, press [Command] + [1] (macOS) or [Control] + [1] (Windows).
2. **Ports View:** Only the wires are hidden. You have a clear view of all controls while the ports are still visible, indicating connections in the patch by color coding. To select this View mode using the keyboard, press [Command] + [3] (macOS) or [Control] + [3] (Windows).
3. **Ports and Wires View:** The full patch including ports and wires is visible and can be changed by making new connections or removing existing ones. To select this View mode using the keyboard, press [Command] + [3] (macOS) or [Control] + [3] (Windows).

5.7. Blocks Missing Dialog

The Missing Blocks dialog shows up when REAKTOR cannot find all Blocks required to open a Rack. The following overview shows the different options depending on whether the Block is part of a product, or a User Block.



1. If a Block is missing from an installed product, you can click **OPEN NATIVE ACCESS** to repair the product installation.
2. If a Block is missing from a product that has never been installed, you can obtain it from the vendor by clicking **OPEN WEBSITE**.
3. If a User Block is missing, ensure that you have the respective file available and click **LOCATE** to relocate it. REAKTOR automatically searches for other missing User Blocks in the same directory.



To learn more about User Blocks and how to use them with Racks, see [User Blocks](#).

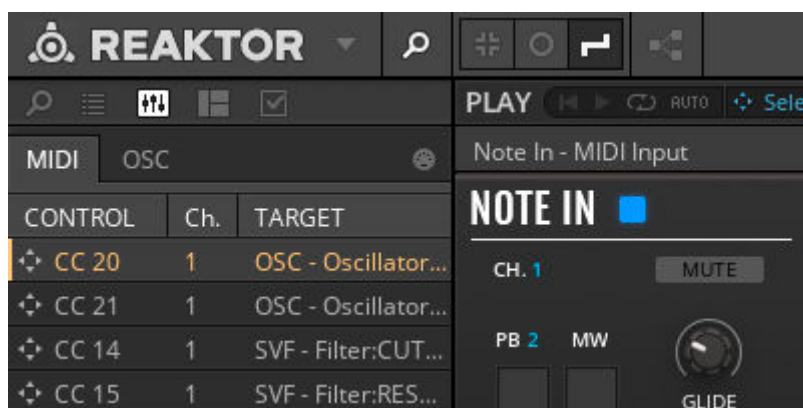
5.8. Advanced Features

The following advanced REAKTOR features are also available in Racks:

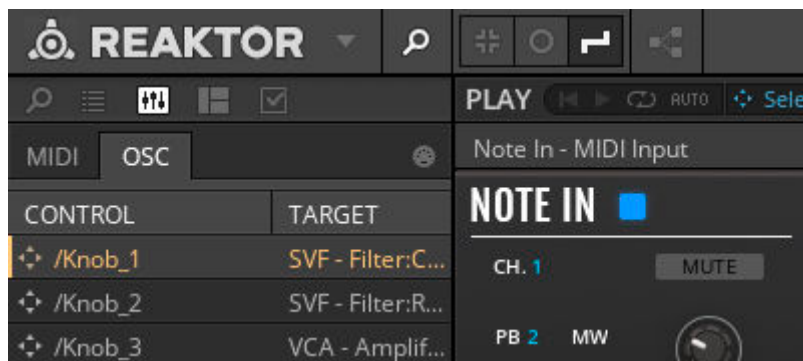
- **Panel Sets:** Panel Sets allow you to save different configurations of the Panel view in the same Rack file. You can show or hide Blocks (while they remain active) or switch between arrangements of Blocks for specific use cases. For more information, refer to the REAKTOR 6 Diving Deeper document, chapter **Panelsets**.



- **MIDI control:** You can assign any of the controls in the Rack to your MIDI controller. For more information, refer to the REAKTOR 6 Getting Started document, chapter **Controlling REAKTOR**.



- **OSC control:** You can also use OSC to control your Rack, or route MIDI and OSC data from and to individual Blocks in the Rack. For more information, refer to the REAKTOR 6 Diving Deeper document, chapter **Controlling REAKTOR with MIDI and OSC**.



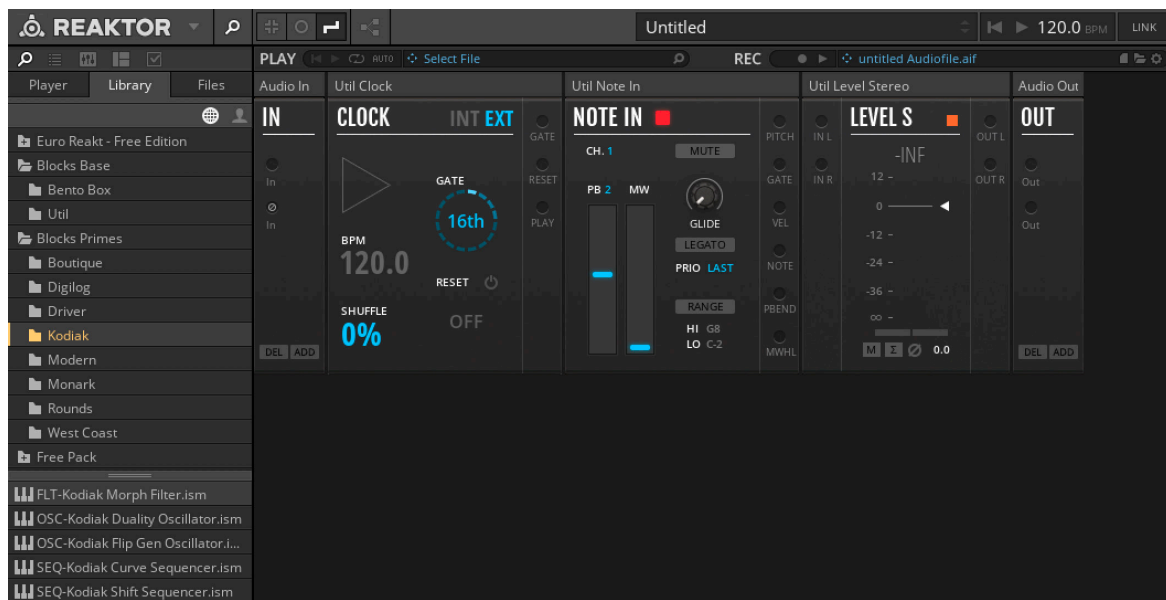
6. ADDING AND ARRANGING BLOCKS

As part of the REAKTOR Library, the [Blocks Library](#) is located in the Browser tab of the side pane. [Adding Blocks to a Rack](#) is done by drag and drop, or by [using the search box](#).

Racks adopt the classic REAKTOR interface paradigm, which is based on the Panel and the Structure view. While the connections are mirrored, the arrangements of Blocks in both views are independent. You can [arrange Blocks in Panel view](#) for optimized playability, and [arrange Blocks in Structure view](#) for improved visibility of the signal flow, for example.

6.1. Blocks Library

When a Rack is open, the **Library** tab in the side pane shows all installed Blocks:



The Library tab in the side pane showing Blocks

In the Library, Blocks are organized in folders. On the first level, they are sorted by product. Each vendor groups the Blocks included in a product in a particular way. Blocks by Native Instruments are available in the two products Blocks Base and Blocks primes, and grouped in the following lines:

- **Bento Box:** Core components of a modular synthesizer, enhanced with special features.
- **Util:** Utilities that provide the infrastructure for your patches.
- **Boutique:** Boutique Blocks are inspired by great synthesizers of the past.
- **Digilog:** Sequencing tools for note processing, complex rhythms, and musical structures.
- **Driver, Monark, Rounds:** Components from well-known NI products, available in Blocks.
- **Kodiak:** Forward-thinking Blocks based on advanced software concepts.
- **Modern:** State-of-the-art Blocks with a contemporary twist.
- **West Coast:** West Coast Blocks are inspired by the Buchla tradition of synthesizers.

All Blocks are associated with a specific category that hints at the Block's most common usage. The category is part of the file name: <category>-<Block name>.ism. The following categories are available:

- **AMP:** Amplifiers control the amplitude of signals.
- **AUX:** Auxiliary (Blocks) provide useful extra functions.
- **DRM:** Drums generate drum sounds.
- **EFX:** Effects process audio signals in interesting ways.
- **FLT:** Filters alter the frequency content of signals.
- **INT:** Integration (Blocks) integrate Blocks with other software and external devices.
- **MIX:** Mixers add signals together.
- **MOD:** Modulators generate signals for controlling other Blocks.
- **OSC:** Oscillators generate audio signals for further processing in a patch.
- **PRO:** Processors process modulation signals in various ways.
- **SEQ:** Sequencers generate or process pitch, gate, and modulation signals.

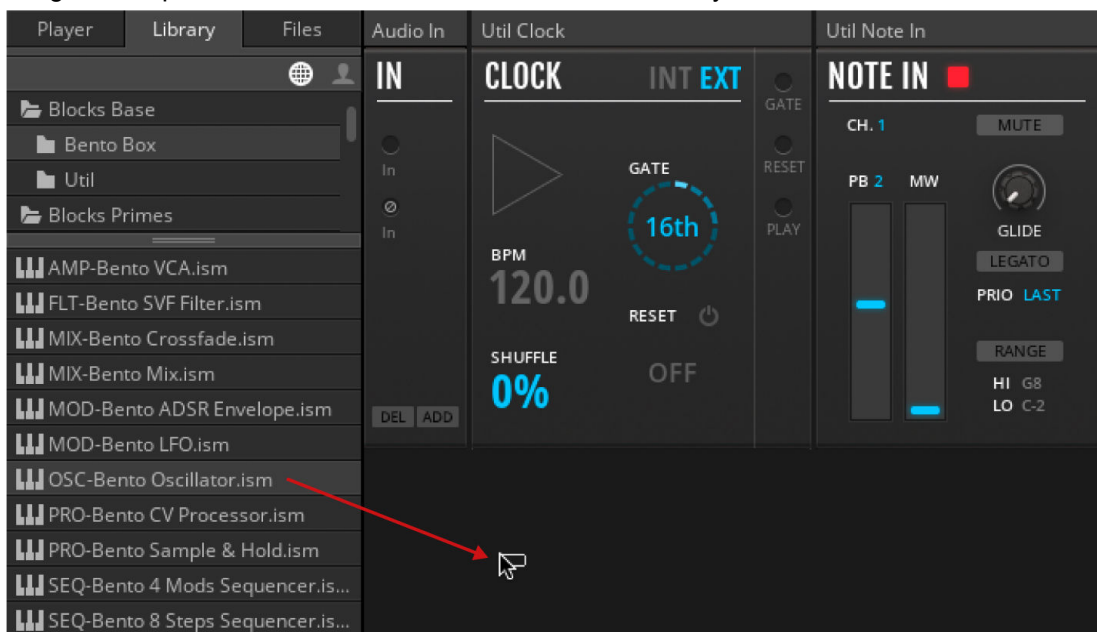


Categories are useful for finding Blocks when adding Blocks to a Rack using the search box. For more information, see [Using the Search Box](#).

6.2. Adding Blocks to a Rack

You can add a Block to your Rack by using drag and drop from the Browser's Library tab:

- Drag and drop the Block from the lower section of the Library tab into the Rack's Panel view.



The Block is added to the Rack.



You can also drag and drop Blocks into the Rack's Structure view.

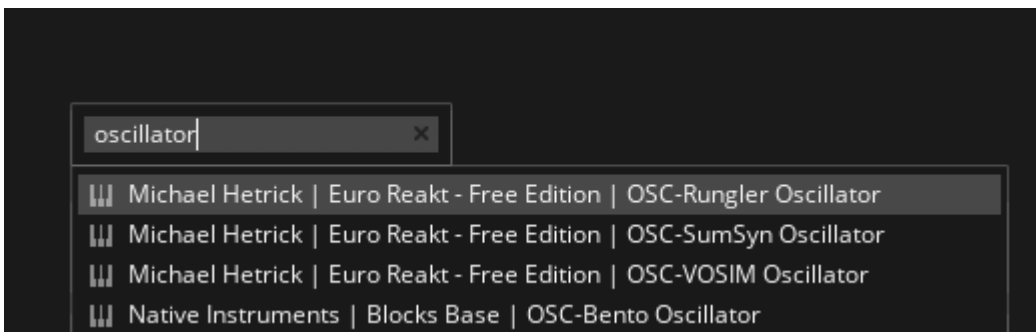
6.3. Using the Search Box

The search box provides a fast and convenient way of finding Blocks and adding them to your Rack:

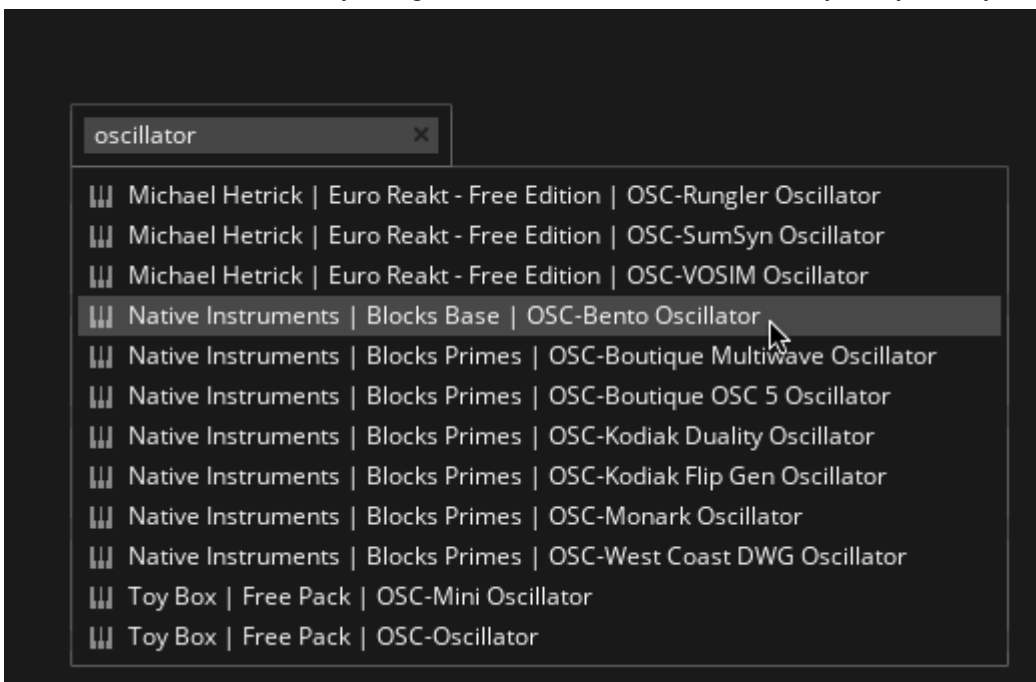
1. Click anywhere in the Rack's Panel view and press [Enter] on your keyboard.



2. Type in the vendor, name, type, or category of the Block (for more information, see [Blocks Library](#)).



3. Select a Block from the list by using the mouse cursor or the arrow keys on your keyboard.



4. Confirm your selection by clicking on the entry or pressing [Enter] on your keyboard.

The Block is added to the Rack.



You can also use the search box in the Rack's Structure view.

6.4. Arranging Blocks in Panel View

In the Panel view, Blocks are organized in a virtual rack that consists of as many rows as you need. All rows of Blocks have the same, fixed height. The width of the rows is determined by the number of Blocks you add to them.

Blocks can be rearranged by using drag and drop. While moving a Block, all connected wires are rerouted automatically to fit the new arrangement. To move a Block to a new position in the Panel view:

- Drag and drop the Block's header to a new position.



The Block is placed in its new position.



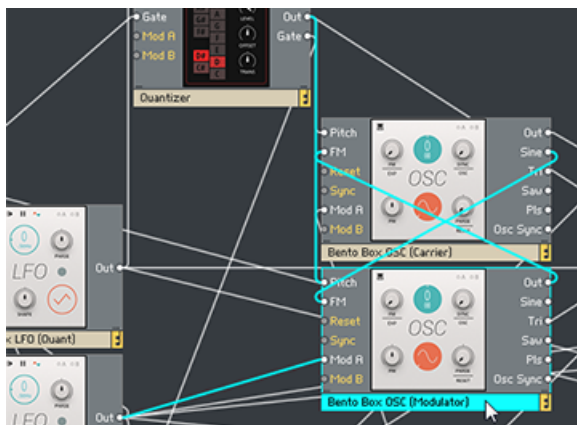
REAKTOR's Panel Sets allow you to save multiple panel configurations for a Rack and recall them at an instant (see REAKTOR 6 Diving Deeper for more information).

6.5. Arranging Blocks in Structure View

In the Structure view, Blocks can be freely arranged, supported by thumbnail pictures and REAKTOR's flexible wiring system. Blocks can be placed on top of each other, however it is recommended to give each Block its own dedicated space in the Structure view to maintain a better overview.

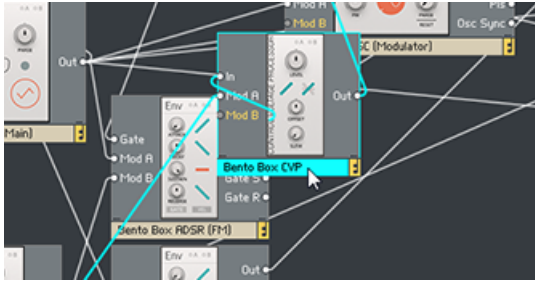
The wires connecting the Blocks always follow the shortest possible path between the ports. If a wire crosses a Block, the overlapping portion of the wire is hidden behind the Block.

- To highlight all wires connected to a Block, including those being hidden behind other Blocks, select it.

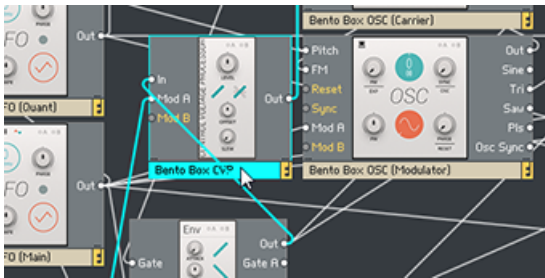


Blocks can be rearranged by using drag and drop. While moving a Block, all connected wires are rerouted automatically to fit the new arrangement. To move a Block to a new position in the Structure view:

- Drag and drop the Block to a new position.



The Block is placed in its new position.



7. PATCHING IN RACKS

Patching in Racks is done by [connecting Blocks](#) in the Panel or Structure view, adjusting [panel controls](#), and [routing modulation](#) to individual parameters. If you are familiar with modular synthesizers, you can adapt your favorite patching strategies to Blocks right away.

While the connections in Panel and Structure view are automatically mirrored, the visual arrangement of Blocks is independent. For example, this allows you to optimize the Panel view for playing your patch, and use the Structure view to create an overview of the signal flow.



If you are new to modular synthesis, the pre-built Racks provided for each Blocks product can serve as a starting point, showing you how Blocks can be combined in a patch. For more information, see [Loading Racks from a Blocks Product](#).

7.1. Connecting Blocks

One of the most important features of Blocks is universal connectivity. You can connect any output to any input and achieve predictable results, regardless of which Blocks are connected.

To establish a connection between two Blocks, you connect an output to an input using a wire. You can do this in the same way both in the Panel and the Structure view. The connections you make in either view are mirrored automatically.

You can remove connections using the context menu or key command. Additionally, you can change the color of wires to improve the visibility of connections or highlight their function in the patch.



For general information about the different types of inputs and outputs see section [Connections and Signals](#). Specific information about each input and output can be found in section [Blocks Reference](#).

7.1.1. Making Connections

To connect an output to an input:

1. Click on the output port you want to connect and keep the mouse button pressed.



2. Drag the mouse onto the input port you want to connect and release the mouse button.



The connection is established, indicated by a wire going from the output port to the input port.



You can also create wires by using drag and drop from an input to an output.

7.1.2. Removing Connections

To remove a connection between an output and an input:

- Right-click a wire and select **Delete Wire**.



Alternatively, you can double-click on an input to remove the wire connected to it.

To remove multiple connections between outputs and inputs:

1. Hold [Shift] and click on the wires you want to delete.



2. Right-click on one of them and select **Delete Wires** to delete the wires.



All selected wires are deleted.



Alternatively, you can press [Delete] or [Backspace] after selecting wires to delete them.

7.1.3. Wire Colors

Racks allow you to improve the visibility of connections or highlight their function in the patch by changing the color of the wires.

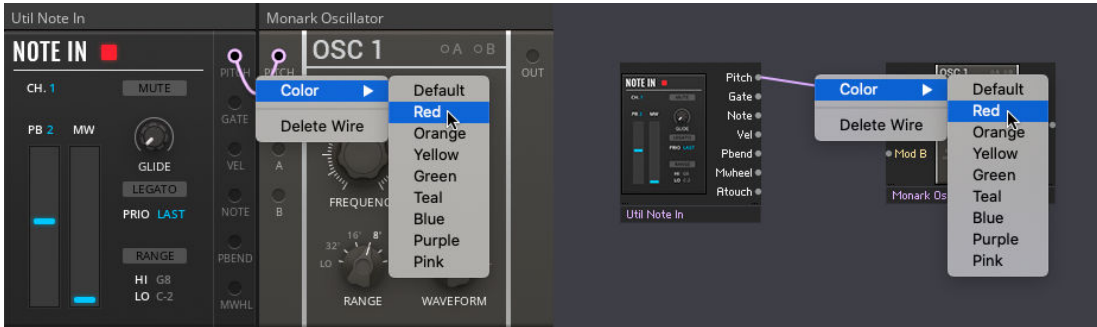
Racks that are compatible with the NKS format by Native Instruments and other vendors use the following color scheme:

- Red: Pitch

- Teal: Gate and clock
- Blue: Modulation
- Purple: Audio

To change the color of a wire:

- Right-click a wire and select a color from the **Color** submenu.



To change the color of multiple wires:

1. Hold [Shift] and click on the wires you want to change the color of.



2. Right-click on one of them and select a color from the **Color** submenu.



The color of all selected wires is changed.

7.2. Panel Controls

Each line of Blocks has its own characteristic design, making it easy to tell one from another. However, most of the panel controls follow the same interface paradigm, which allows you to adjust and play the parameters in an intuitive manner. Below, the most common panel controls are explained briefly.



Find detailed information about each panel control in section [Blocks Reference](#) as well as in the REAKTOR Info Hints (see section [Info Hints](#)).



(1) **Knob:** Knobs are the most common panel control in Blocks. Click and drag a knob up and down to change its value. The current value is shown below the knob while adjusting the parameter. Double-clicking on a knob resets its value to default.

(2) **Circular value control:** These controls can be adjusted like knobs, by clicking and dragging them up and down. They feature a large value display and are used for parameters like oscillator pitch and waveform selection. For pitch, a dual control is used, with coarse tuning at the top and fine tuning at the bottom. Double-clicking on the control resets its value to default.

(3) **Modulation buttons:** The **A** and **B** buttons are used for routing modulation signals in a Block. See section [Modulation Routing](#) for more information.

(4) **Symbol mode control:** These controls are used to select between different modes of a function, in this case different envelope shapes on Bento Box ADSR. Click on the symbol to step through the available modes. The symbol reflects the current setting. Double-clicking on the control resets its value to default.

(6) **Global control:** The options at the top of a Block select between different modes of operation, like key tracking for the Boutique OSC 5 in this case.

(7) **Label control:** On some Blocks, the labels of controls can be clicked to change a related parameter. On the Boutique OSC 5, the waveform label selects alternate waveforms for each of the intervals.

(8) **Fader:** Faders are used as an alternative to knobs and can be adjusted in the same way, by clicking and dragging them up and down. The current value is shown above the fader while adjusting the parameter. Double-clicking on a fader resets its value to default.

(9) **Modulation depth slider:** The modulation depth sliders control the amount of modulation applied to the associated parameter. Double-clicking on the slider resets its value to default. See section [Modulation Routing](#) for more information.

(10) **Button:** Buttons switch between two alternate options or toggle single functions on or off. You can change their value by clicking on them. If the button switches between alternate options, the label will change accordingly. If it toggles a single function on or off, the on-state is highlighted.

(11) **Momentary button:** These buttons are used for gate and reset functions. They light up when a gate arrives at the associated input and triggers the function. You can also click on them to manually set off the function, like starting the envelope on the Bento Box ADSR.

(12) **Control mode button**: Some controls, like **PHASE**, **FM**, and **SYNC** on the Bento Box Oscillator, feature auxiliary buttons next to their labels that allow you change their mode of operation.

7.3. Modulation Routing

Each Block has two modulation buses A and B, represented by the two **A** and **B** buttons on the panel. The buttons correspond to the Block's modulation inputs **A** and **B**. The modulation buses distribute the signals arriving at the modulation inputs to all parameters that can be modulated. Dedicated modulation depth sliders set the amount of modulation applied to each parameter.

The **A** and **B** buttons include a signal indicator that displays incoming modulation signals. It lights up red for positive values and blue for negative values. If no modulation signal is present at the modulation input, the corresponding button is grayed out. However, you can still click the buttons and view or change the modulation amounts.

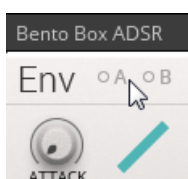


The modulation buses

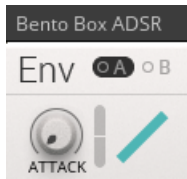
7.3.1. Modulation Depth Sliders

The modulation depth sliders set the amount of modulation from the currently selected modulation bus. All modulation depth sliders are bipolar, meaning that positive (non-inverted) or negative (inverted) modulation can be applied. If you turn up the modulation amount for a parameter, the modulation will also be displayed as a moving arrow next to the panel control.

- Click the **A** or **B** button to show the modulation depth sliders for a modulation bus.

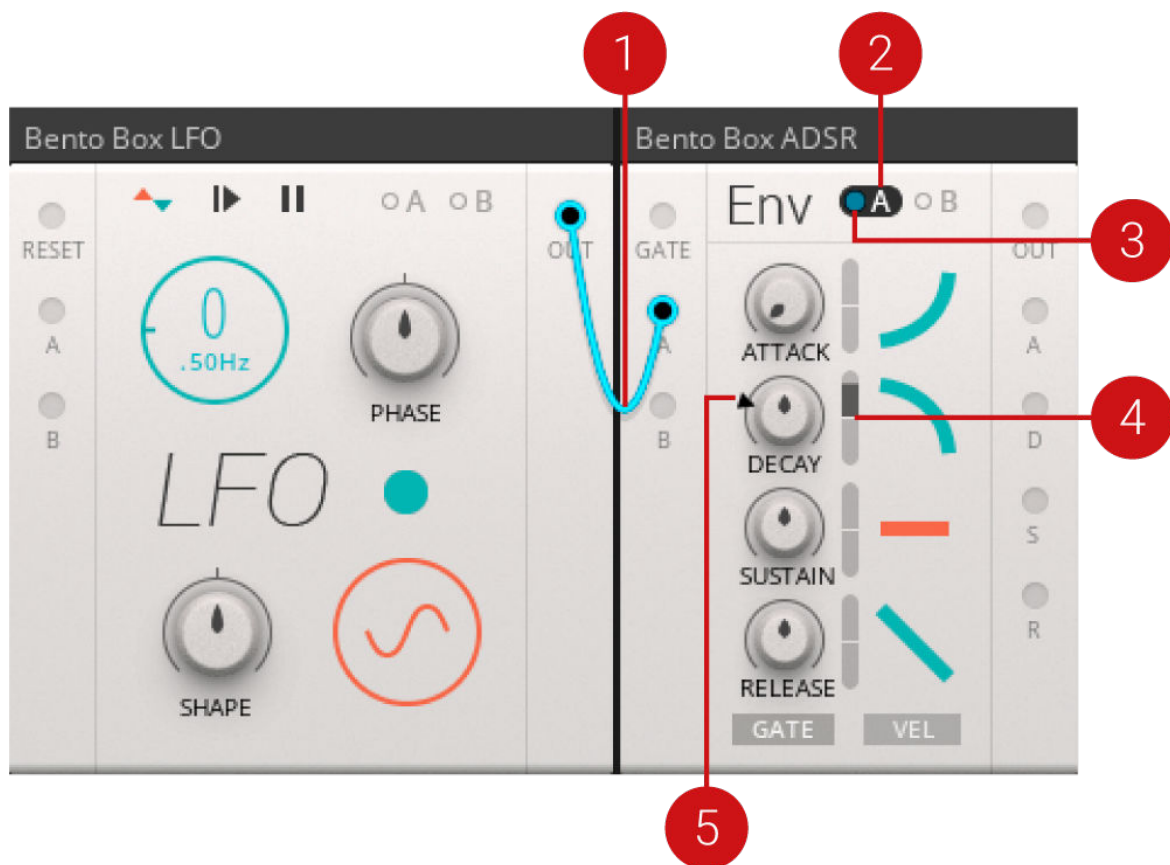


The modulation depth sliders are shown for all parameters that can be modulated.



7.3.2. Example Routing

In the following example, the Bento Box LFO is modulating the Decay parameter of the Bento Box ADSR.



1. The **Out** of the Bento Box LFO is connected to modulation input **A** of the Bento Box ADSR.
2. The **A** button on the Bento Box ADSR is pressed. The modulation depth sliders appear for all parameters that can be modulated.
3. The **A** signal indicator on the Bento Box ADSR is lighting up blue, displaying a negative value of the modulation signal coming from Bento Box LFO.
4. The modulation depth slider for **DECAY** is turned up, routing the modulation signal on modulation bus **A** to this parameter.
5. The arrow next to the **DECAY** parameter shows the momentary position of the modulation relative to the knob setting.

8. CONNECTIONS AND SIGNALS

In order to achieve the desired level of flexibility, there is no distinction between different types of signals in REAKTOR Blocks. All connections are being made with signals within the range of -1 to +1. This allows you to connect any output to any input, without having to worry about signal type or value range. This section provides an overview of the available [input and output types](#), including links to sections that detail all of them. Additionally, you can learn about how to make [feedback connections](#).

8.1. Input and Output Types

Different Blocks require different types of inputs and outputs. Since all signals sent between Blocks consist of the same range, it is necessary to determine how values within that range are converted into a predictable result, depending on what type of input is receiving them.

There are eight different types of input and outputs:

- **General inputs and outputs:** The **In** and **Out** ports can be used for all kinds of signals, depending on the Blocks' functionality and application. For more information, see section [General Inputs and Outputs](#).
- **Modulation:** The **Mod A** and **Mod B** inputs are general modulation inputs. For more information, see section [Modulation](#).
- **Pitch:** **Pitch** inputs are used to control the frequency of oscillators and filters. For more information, see section [Pitch](#).
- **Gate:** **Gate** inputs are used to control a variety of functions in different Blocks, from triggering envelopes or advancing sequencer steps to resetting LFOs. For more information, see section [Gate](#).
- **Reset:** **Reset** inputs function in a similar manner as **Gate** inputs, and are used to reset the playback of sequencers. For more information, see section [Reset](#).
- **Play:** **Play** inputs function in a similar manner as **Gate** inputs, and are used to pause and resume the playback of sequencers. For more information, see section [Play](#).
- **Sync:** **Sync** inputs function in a similar manner as **Gate** inputs, and are used to synchronize the phase of an oscillator to an external source. For more information, see section [Sync](#).
- **Pluck:** **Pluck** inputs function in a similar manner as **Gate** inputs, and are used to trigger or excite special filters and amplifiers, for example the West Coast LPG. For more information, see section [Pluck](#).

8.2. General Inputs and Outputs

The **In** and **Out** ports can be used for all kinds of signals, depending on the Blocks' functionality and application. All inputs and outputs operate at audio rate and in the range of -1 to +1. If a Block has multiple inputs, for example the Bento Box Mix, the inputs are numbered: **In 1**, **In 2**, **In 3**, and so on.

In some cases, inputs or outputs are labeled depending on their specific function. For example, the Bento Box ADSR has four **Gate** outputs that send a gate high signal as long as the corresponding envelope stage is active:

- **Gate A** (gate high during attack)
- **Gate D** (gate high during decay)
- **Gate S** (gate high during sustain)
- **Gate R** (gate high during release)

The same applies to oscillators with multiple waveform outputs, or filters with separate outputs for each filter mode, for example the the Bento Box SVF:

- **Out** (main output with switchable filter mode)
- **LP Out** (low pass mode output)
- **BP Out** (band pass mode output)
- **HP Out** (high pass mode output)

8.3. Modulation

The modulation inputs **A** and **B** are general modulation inputs. They allow for modulation in the full range of -1 to +1 and at any rate. Signals arriving at these inputs can be used to modulate various different parameters on each Block.



For information about routing modulation signals, see [Modulation Routing](#).



The modulation inputs **A** and **B** on the Bento Box SVF

Oscillators and filters often have an additional **FM** input. These frequency modulation inputs can be used to add modulation to the pitch of an oscillator, or the cutoff frequency of a filter. For example, it can be used to modulate the cutoff frequency of a filter with an envelope, or to set up multiple oscillators for FM synthesis. The corresponding panel control sets the depth of modulation.



The FM input on the Bento Box SVF

8.4. Pitch

Pitch inputs, which are used to control the frequency of oscillators and filters, scale an input value between 0 and 1 up to a MIDI note value between 0 and 120. A value of 0 equals MIDI note 0, a value of 0.5 equals MIDI note 60, and a value of 1 equals MIDI note 120.

However, this does not mean that you have to use quantized signals to control your oscillators. If you connect an envelope to the **Pitch** input of an oscillator, the envelope output of 0 to 1 results in a smooth sweep ranging from MIDI note 0 to MIDI note 120 (given that no offset has been dialed in using the frequency controls of the oscillator).

Using this range allows for convenient scaling and offsetting of pitch signals, since the range from 0 to 1 covers exactly 10 octaves. Similar to the 1 V/Oct-standard as found on contemporary modular synthesizers, you are dealing with a value of 0.1/Oct in REAKTOR Blocks. For instance, if you want to transpose a pitch signal two octaves up, you have to add an offset of 0.2 (or 20%) to the signal.

Pitch inputs also accept signals lower than -1 and higher than +1, so pitches above and below the specified range can be achieved, too.

8.5. Gate

Gate inputs are used to control a variety of functions in different Blocks, from triggering envelopes or advancing sequencer steps to resetting LFOs. They operate in the range of -1 to +1, so any signal can be connected. **Gate** outputs usually send out a pulse wave signal in the range of 0 to 1.

Gate inputs look for a positive zero crossing of the input signal. As soon as the input signal rises above 0, it is considered a gate-on message, and once the signal drops to 0 or below, it is considered a gate-off message.

For example, you can connect the output of an LFO to the **Gate** input of an envelope. The LFO has an output of $-/+0.5$. When its output goes above 0, this is considered a gate-on message and the envelope is triggered and sustained. When the LFO returns to 0 or less, this is considered a gate-off message and the envelope is released.

The same applies to any Block with a **Gate** input: Connect the LFO to the **Gate** input of a sequencer, and every time the LFO travels into the positive part of its cycle, the sequencer will advance one step. Since **Gate** inputs operate at audio rate like any connection in REAKTOR Blocks, you can also replace the LFO with an audio oscillator and run the sequencer at audio rate, creating a new interesting sound source.

Velocity

Some **Gate** outputs, for example on the Util Note In and the Bento Box 8 Steps, include velocity information. In this case, the gate signal's amplitude varies depending on the velocity of the played note or the corresponding sequencer step.

To use this information, the receiving input needs to be sensitive to amplitude changes in the gate signal, like the **Gate** input on the Bento Box ADSR, for example. Velocity is derived from how far above 0 the initial increase is. If the input leaps from 0 to 1, the envelope is triggered at maximum velocity. A jump from 0 to 0.5 would trigger the envelope at half velocity, and so on.



Note that velocity sensitivity can have undesired effects when using arbitrary signals. For example, when using a triangle LFO to trigger an envelope with velocity sensitivity, the output level of the envelope will be very low: the LFO cycles above 0, but the initial increase is very small.

8.6. Reset

Reset inputs function in a similar manner as **Gate** inputs, looking for a positive zero crossing of the incoming signal. As soon as the input signal rises above 0, it is considered a reset gate. However, unlike **Gate** inputs, negative zero crossings have no effect.

Reset inputs are commonly found on sequencers (or other counting devices), for example the Bento Box 8 Steps. Every time a positive value arrives, the sequencer is reset to the first step. On the Bento Box 8 Steps panel, you will notice that the **RESET** button lights up when a reset signal is received.

You can use any signal to reset your sequencers, however the **Reset** output on the Util ClockI is specifically designed for this task. It sends out a reset signal every time it is started, and repeats it at an interval set by the **RESET** control on the panel. By distributing the reset signal from the Util Clock across your patch, you can ensure that all sequencers remain synchronous.

8.7. Play

Play inputs function in a similar manner as **Gate** inputs, looking for a positive zero crossing of the incoming signal. As soon as the input signal rises above 0, it is considered a gate-on message, and once the signal drops to 0 or below, it is considered a gate-off message.

Play inputs are found on sequencers, for example the Kodiak Shift Sequencer. As long as the incoming signal remains positive after a gate-on message, the sequencer's playback is enabled, allowing it to advance one step every time a gate-on message is received at the Gate input. When the signal drops to 0 or below, the sequencer's playback is disabled, stopping the sequence at its current position and ignoring gate-on messages received at the Gate input. Additionally, all active sequencer events are reset.

You can use any signal to control sequencer playback, however the **Play** output on the Util Clock is specifically designed for this task. It sends out a constant gate signal when its clock is active (Play button enabled). By connecting it to a sequencer's **Play** input, you can prevent hanging notes caused by active sequencer events, for example **HOLD** events on the Kodiak Shift Sequencer. The **Play** input can also be used to control playback from a keyboard via the Util Note In's **Gate** output, or for interrupting the playback periodically with an LFO.



If no connection is made to the Play input, playback is constantly enabled.

8.8. Sync

Like **Gate** and **Reset** inputs, **Sync** inputs look for a positive zero crossing of the incoming signal. As soon as the input signal rises above 0, a sync event is triggered. Negative zero crossings have no effect.

Sync inputs can be found on oscillators. They are used to synchronize the phase of the signal generated by the oscillator to an external source, for example another oscillator. This is useful for adding stability to complex FM patches with multiple operators, patching classic oscillator sync sounds, or creating interesting new waveforms by mixing signals from multiple synced oscillators.



The Bento Box OSC's **Sync** input has the option to accept a special synchronization signal from the **OSC Sync** outputs found on the Bento Box Oscillator and the Boutique Multiwave Oscillator. This allows for oscillator synchronization with the best possible audio quality.

8.9. Pluck

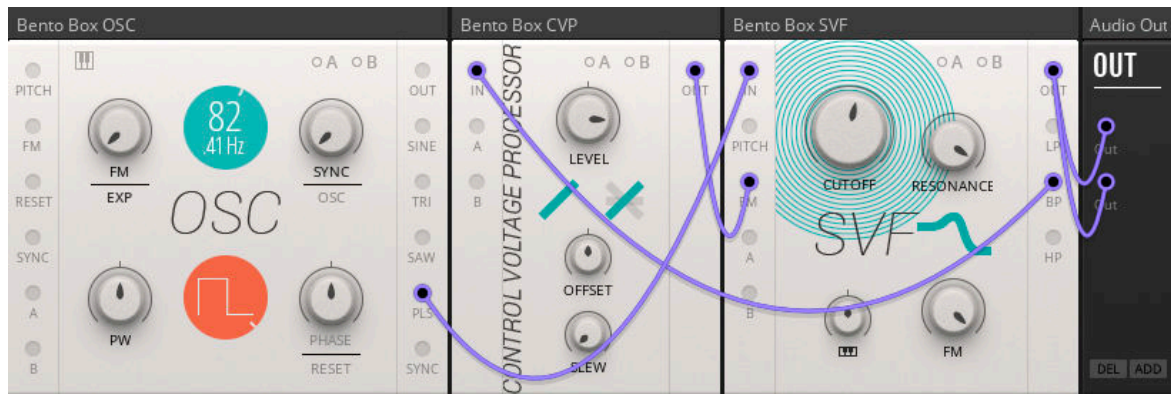
Pluck inputs are found on Blocks that employ an optocoupler to smooth a parameter's response to external control, for example the West Coast LPG. Any signal within the range in the range of -1 to +1 can be connected, however for best results it is recommended to use signals with sharp rising edges, like gates, clocks, pulse and saw wave signals, sample and hold waveforms, or sequences.

The **Pluck** input processes any sharp-edged signal in a way that makes it suitable for briefly exciting the optocoupler when a sharp rising edge is received. This applies a natural sounding, percussive envelope to the parameter involving the optocoupler, for example **LEVEL** on the West Coast LPG. The amount of modulation applied depends on the strength of the rising edge.

8.10. Feedback Connections

Feedback can be patched across any number of modules. However, if you want to make a feedback connection between an input and an output of the same Block, an additional Block needs to be inserted in the feedback loop.

Aside from the Bento Box Mix, which allows you to add the feedback signal to an existing input signal, the Bento Box CV Processor is particularly well suited for this task, providing the means for further processing of the feedback signal.



Feedback connection from the BP output of the Bento Box SVF to its FM input

9. INTEGRATING HARDWARE SYNTHESIZERS WITH CONTROL VOLTAGES

Since the connections and signals used in Blocks follow the same paradigm as analog control voltages (CV), they can be sent directly to DC-coupled converters for CV control of hardware synthesizers. DC-coupled converters or audio interfaces are capable of passing very low frequencies and static voltages at their outputs. They can therefore be used for sending out control voltages from the computer.



Most converters and audio interfaces employ output filtering (AC-coupling) and are not suitable for this purpose. Please refer to the technical specifications of your converter or audio interface to check if it is AC-coupled or DC-coupled.



DC-coupled converters specifically made for sending out control voltages from the computer are available as Eurorack modules by Expert Sleepers.

9.1. Routing Control Voltages to External Hardware

In order to route control voltages to external hardware, you need to enable the outputs of your DC-coupled converter or audio interface in REAKTOR's Audio and MIDI Settings. By adding Out Ports, you can make them available in your patch and connect any modulation gate signal to them:

- To modulate your hardware synthesizer from Blocks, connect any modulation signal to an **Out Port** that is routed to your DC-coupled audio interface or converter.

You can now patch the output of your DC-coupled audio interface or converter to a CV input on your hardware synthesizer.



The [CV Processor \(PRO\)](#) and [CV Mix \(MIX\)](#) Blocks are useful for processing modulation signals before sending them to a DC-coupled audio interface or converter. They allow you to attenuate, invert, and offset signals.

- To trigger envelopes or clock sequencers on your hardware synthesizer from Blocks, connect any gate signal to an **Out Port** that is routed to your DC-coupled audio interface or converter.

You can now patch the output of your DC-coupled audio interface or converter to a gate, trigger, or clock input on your hardware synthesizer.



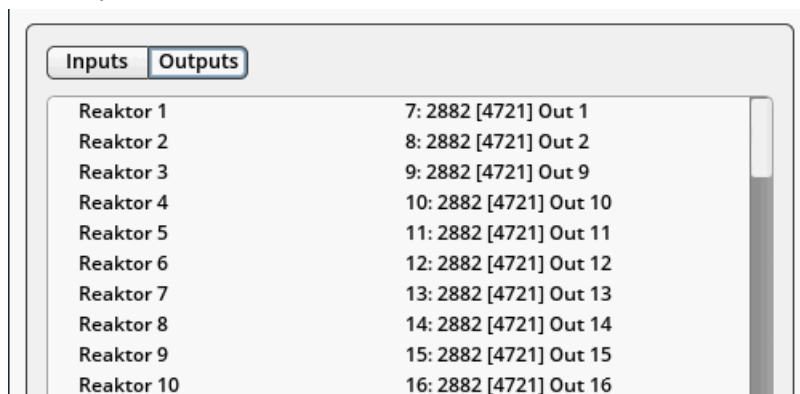
Some hardware synthesizers and synthesizer modules do not respond correctly to gate signals sent from DC-coupled audio interfaces or converters. The [Gates & Trigs \(INT\)](#) and [Pitch CV Out \(INT\)](#) Blocks employ special output processing to prevent these issues and should be inserted before any output used for gate signals.

9.2. Using the Pitch CV Out Block

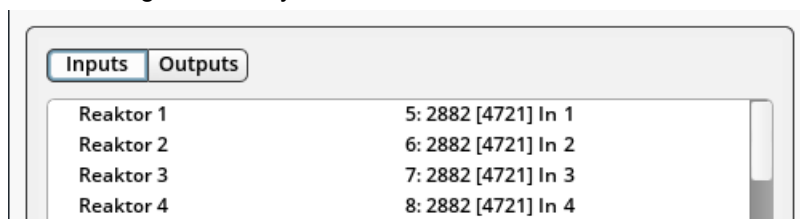
The Pitch CV Out allows you to use Blocks for controlling hardware synthesizers with analog pitch and gate CV (Control Voltage) inputs via DC-coupled audio interfaces or D/A converters. An automatic measurement routine calibrates the control voltages sent to the external oscillator, perfectly matching its response to the pitches in Blocks across the full MIDI note range from C-2 to G8.

To control the pitch of your hardware synthesizer from Blocks, including automatic calibration of the external oscillator, follow these instructions:

1. Open the **Audio and MIDI Settings ...** under **File** in the REAKTOR main menu.
2. Go to the **Routing** tab and click on **Outputs**.
3. Assign the analog outputs of your DC-coupled audio interface to REAKTOR's output channels (**Reaktor 1**, **Reaktor 2**, etc.). Alternatively, assign the digital outputs of your audio interface that are connected to your DC-coupled D/A converter. In this example, the ADAT outputs **Out 9** through **Out 16** of the audio interface are connected to an Expert Sleepers ES-3 DC-coupled converter.



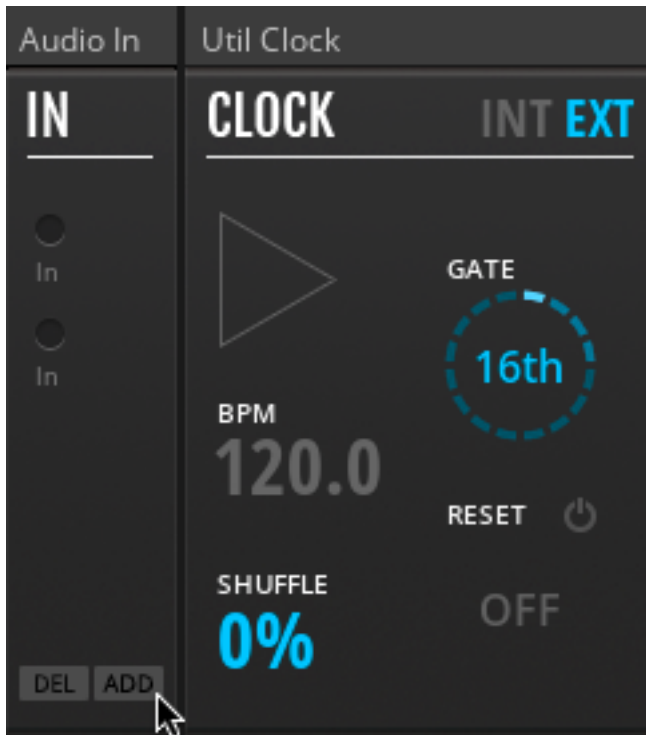
4. Click on **Inputs**.
5. Assign the analog inputs of your audio interface that are connected to the outputs of your hardware synthesizer to REAKTOR's input channels (**Reaktor 1**, **Reaktor 2**, etc.). In this example, the analog inputs **In 1** through **In 4** of the audio interface are connected to the outputs of an analog modular synthesizer.



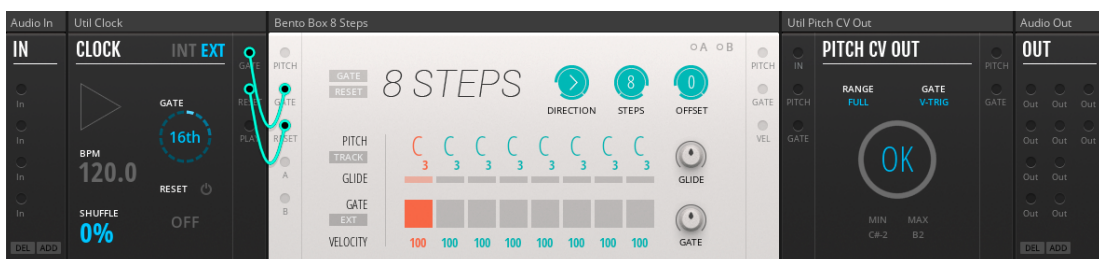
6. Click on **ADD** on the Out Block to add additional outputs to the Rack. Repeat for all input channels assigned in step 3. The outputs on the In Block correspond to the REAKTOR output channels with the same number.



- Click on **ADD** on the In Block to add additional inputs to the Rack. Repeat for all input channels assigned in step 5. The inputs in the In Block correspond to the REAKTOR input channels with the same number.



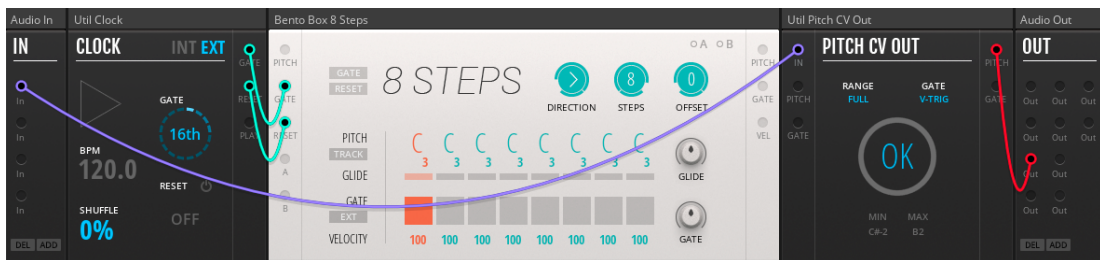
- Add the Pitch CV Out Block to the patch, as well as the Blocks you want to use for controlling your hardware synthesizer. In this example, the Bento Box 8 Steps sequencer is used, clocked from the Util Clock.



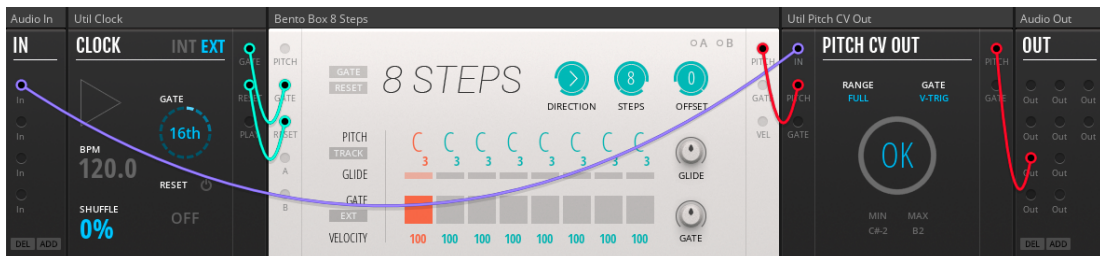
- Route one of the analog inputs of your audio interface to the Pitch CV Out Block by connecting the corresponding **In Port Module** to the Pitch CV Out's **In** input. In this example, the analog input **In 1** is used, which is assigned to input channel **Reaktor 1** (see step 5).



- Connect the Pitch CV Out's **Pitch CV** output to one of the **Out Port Modules** that are assigned to the analog outputs of your DC-coupled audio interface or converter. In this example, the ADAT output **Out 9** is used, which is assigned to output channel **Reaktor 3** (see step 5). ADAT output **Out 9** corresponds to the first analog output on the Expert Sleepers ES-3.



- Connect the **Pitch** output of the Block you want to use for controlling the pitch of your hardware synthesizer to the Pitch CV Out's **Pitch** input. In this example, the Bento 8 Steps sequencer is used.



- Connect the analog output of your DC-coupled audio interface or converter (matching your routing in step 10) to the pitch CV input of the external oscillator on your hardware synthesizer. In this example, the first output of the Expert Sleepers ES-3 (fed from ADAT output **Out 9** of the audio interface) is connected to the 1V/Oct input of an analog oscillator module.



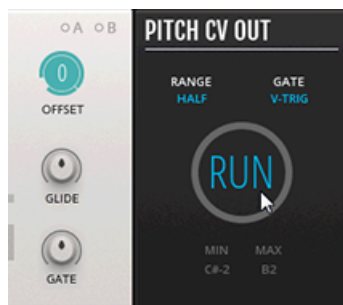
- Connect the output of your hardware synthesizer to the analog input of your audio interface (matching the routing in step 9). The output must carry a constant oscillator signal with a basic waveform (use a sine output for best results). In this example, the sine wave output of the oscillator is connected to the analog input **In 1** of the audio interface.



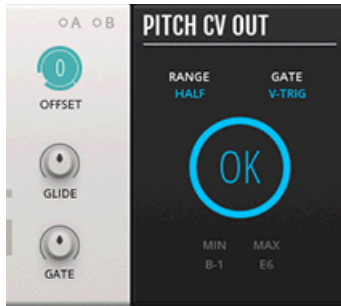
- Set the external oscillator's frequency offset control to a medium frequency, for example somewhere around 261 Hz (MIDI note C3). This can improve the measurement accuracy. You do not have to set a specific frequency, the calibration matches the pitch independently of the external oscillator's base frequency. Only extremely high settings for the oscillator's base frequency can cause the calibration to fail.
- If you are using a DC-coupled converter with a wide voltage range, for example an Expert Sleepers module, set the **RANGE** control on the Pitch CV Out Block to **HALF**. This improves the measurement accuracy. If you are using a standard DC-coupled audio interface, set **RANGE** to **FULL**.



- Click **RUN** on the Pitch CV Out Block to start the calibration.



- When the calibration is finished, the **MIN** and **MAX** displays show the available note range for the external oscillator on your hardware synthesizer.



If the blue circle at the center of the Pitch CV Out Block is not fully lit after the calibration, the measurement has not been successfully completed. This happens if the base frequency of the external oscillator on the hardware synthesizer is set too high. Set to a medium frequency as described above and run the calibration again.

You can now use the Blocks patch to control your hardware synthesizer. In this example, a sequence is played back on the hardware synthesizer using the Bento Box 8 Steps sequencer.



To control an envelope on your hardware synthesizer, connect the gate signal from your Blocks patch to the Pitch CV Out's **Gate** input and route the Pitch CV Out's **Gate CV** output to another analog output of your DC-coupled converter or audio-interface. During calibration the **Gate CV** output goes high, opening any envelope connected to the output of your DC-coupled audio interface or converter. This allows the raw oscillator signal to pass through the VCA on your hardware synthesizer for the measurement.

10. USER BLOCKS

User Blocks, whether you have downloaded them from the [REAKTOR User Library](#) or built your own, [can be loaded in Racks](#) and used alongside officially licensed Blocks from Native Instruments and third-party manufacturers. User Blocks benefit from full Rack functionality including panel patching, view modes, the search box, and saving patches in the host project.



Loading User Blocks in Racks required the full version of REAKTOR 6. Users of REAKTOR 6 Player can load User Blocks into Racks, however the User Blocks will run in Demo mode and be muted after 30 minutes.

User Blocks are stored in the Rack file with a reference to their file path. [Installing User Blocks](#) in the User Blocks folder root directory enables convenient sharing of Racks between users. As long as the User Blocks are located in the same relative path within the User Blocks folder, they will be automatically detected when loading a Rack. Additionally, the User Blocks folder facilitates optimized loading as well as search functionality in REAKTOR.



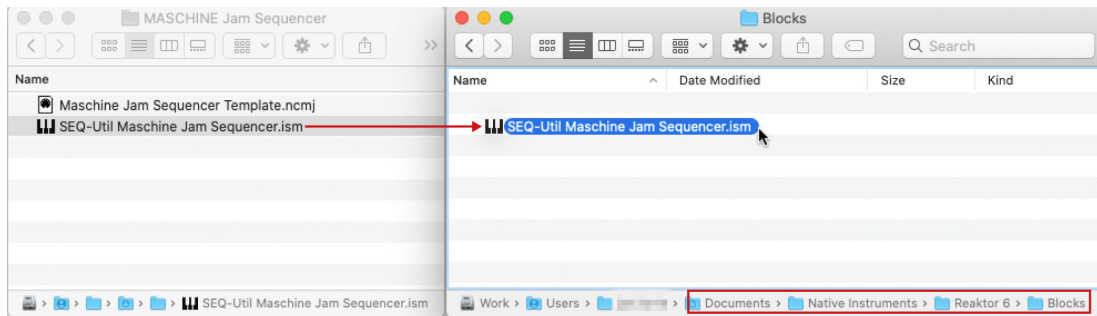
If REAKTOR cannot find all required Blocks when loading a Rack, it will show the Missing Blocks dialog that you can use to relocate them. For more information, see [Blocks Missing Dialog](#).

In addition to being correctly installed, User Blocks have to be based on the Blocks building template and adhere to the specifications detailed in its manual. If a Block does not include Panel Ports, you can use the template to add them. For more information, see [Building Your Own Blocks](#). Once you have made changes to a User Block, you can [update the User Block](#) in the User Blocks folder.

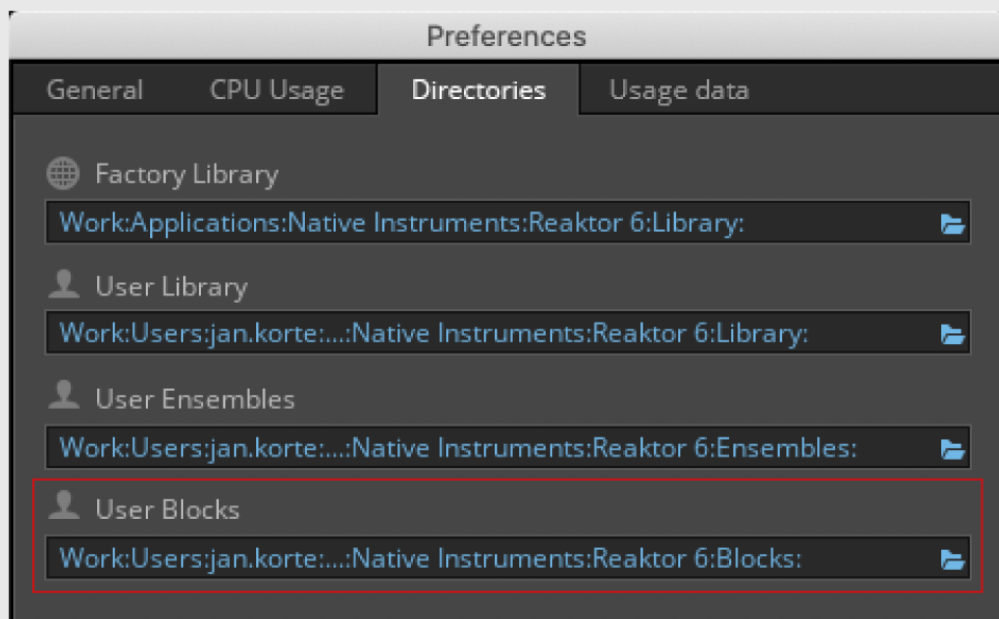
10.1. Installing User Blocks

After building a User Block or downloading one from the [REAKTOR User Library](#), you have to install it in the REAKTOR Library.

- To install the User Block in the REAKTOR Library, move the *.ism* file into the default User Blocks folder on your hard drive:
 - macOS: *Macintosh HD/Users/<your user name>/Documents/Native Instruments/Reaktor 6/Blocks*
 - Windows: *C:\Users\<your user name>\Documents\Native Instruments\Reaktor 6\Blocks*



You can change the location of the User Blocks folder in the **Directories** tab of the REAKTOR Preferences.



10.2. Updating User Blocks

If you have made changes to one of your own User Blocks, or downloaded an updated User Block from the REAKTOR User Library, it is recommended to keep the same file name and replace the existing file in the User Blocks folder. Any Rack referencing the original version of the User Block will then automatically load with the updated one.



Changes made to the User Block can affect the integrity of existing Racks. To avoid this, ensure that the number of Ports is identical, and that the internal structure of the Block has not changed fundamentally. Keep backup versions of your User Blocks in order to be able to restore Racks that rely on them.

10.3. Loading User Blocks in Racks

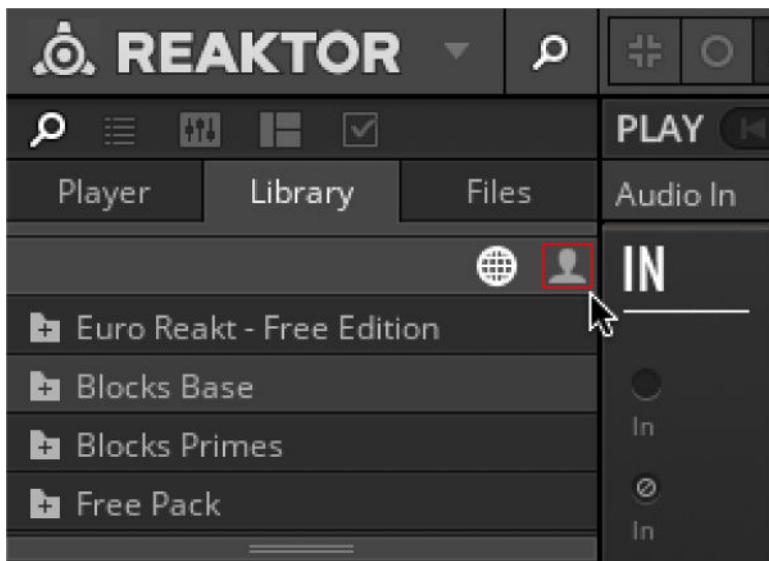
You can add a User Block to your Rack by using drag and drop from the Browser's Library tab:

1. Open an existing Rack in REAKTOR or create a new one.

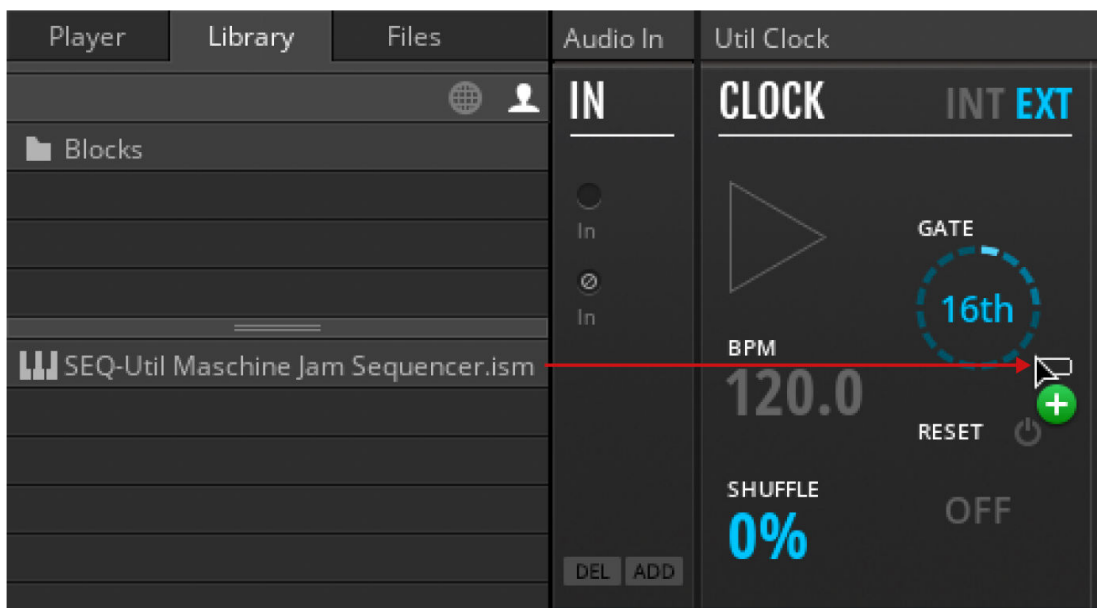


For information about opening and creating Racks, see [Basic Use](#).

2. Click on the User icon in the **Library** tab of the Browser.



3. Drag and drop the User Block from the lower section of the Library tab into the Rack's Panel view.



The User Block is added to the Rack.



You can also use the search box to find and add User Blocks to a Rack. For more information, see [Using the Search Box](#).

10.4. Building Your Own Blocks

Every Block is built on a sophisticated framework that facilitates the user interface, the unified connection scheme, and the underlying signal processing. To support builders who want to contribute new Blocks to the format, an elaborate building template has been created and uploaded to the REAKTOR User Library on our website.

The building template includes all components needed to create fully compatible Blocks based on your own designs, including the Panel Ports and different views required for Racks. It features a comprehensive manual that explains all the specifics of the framework, allowing for a smooth transition into building your own Blocks. Download the building template [here](#).



The Blocks template in REAKTOR

11. BLOCKS REFERENCE

In this chapter you can explore all the Blocks that come with REAKTOR 6 and make yourself familiar with their general functionality as well as their controls, inputs, and outputs.

If you are using REAKTOR 6 PLAYER, you can download the free product Blocks Primes. It includes the Bento Box and Util lines of Blocks. The other lines of Blocks described in this section are included in the product Blocks Primes, which you can purchase separately for use in REAKTOR 6 PLAYER.



The information provided in this section is also available as Info Hints in the REAKTOR 6 application, so you can learn about the Blocks while patching. For more information, see section [Info Hints](#).

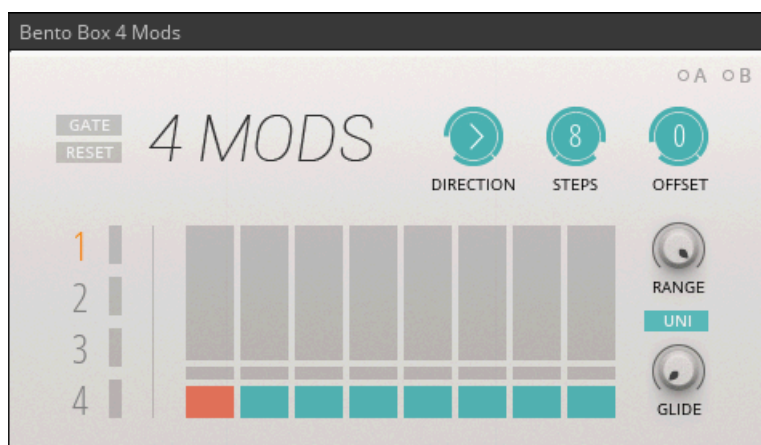
11.1. Bento Box

Bento Box Blocks are an all-purpose set of modules that include the core components of a modular synth setup. Each offers extra twists that open the modules to more advanced applications.



The Bento Box line of Blocks is included in the product Blocks Base, which you can download for free as part of [KOMPLETE START](#).

11.1.1. 4 Mods (SEQ)



The Bento Box 4 Mods

Complementing the [8 Steps \(SEQ\)](#) sequencer, Bento Box 4 Mods provides four sequences for modulation purposes. Each sequence consists of a set of 8 freely adjustable values, with selectable glide for smooth transitions between steps. In addition to using 4 Mods as a modulation source, its outputs can also be sent to the [Quantizer \(PRO\)](#) for creating note sequences. All global parameters (**DIRECTION**, **STEPS**, **OFFSET**) can be modulated, which makes 4 Mods an incredibly powerful tool for creating complex and evolving sequences.

Controls

- **GATE**: Lights up when a gate is received via the **Gate** input, and can be used to manually advance the sequencer one step.
- **RESET**: Clicking this button resets the sequencer to the first step. Additionally, the button indicates when the internal counter is reset when receiving a positive zero crossing at the **Reset** input.
- **DIRECTION**: Allows you to select between five different playback modes:
 - **Forwards**: The sequencer advances from left to right.
 - **Backwards**: The sequencer advances from right to left.
 - **Pendulum**: The sequencer advances forwards until it reaches the final step, at which point it immediately changes direction, advancing backwards until the first step is reached, and so on.
 - **Ping-Pong**: The sequencer advances in the same way as in Pendulum mode, however the first and the last step are repeated when changing direction.
 - **Random**: The sequencer advances in a random fashion. Every time a gate is received, the sequencer randomly chooses one of the active steps to advance to.
- **STEPS**: Allows you to set the length of the sequence in number of steps.
- **OFFSET**: Allows you to shift the playback position, including the first step of the sequence (reset position). This is especially useful when the sequence length is shortened by using the **STEPS** control. If **STEPS** is set to 1, the **OFFSET** parameter provides direct control over the playback position.
- **1 ... 4**: Selects the sequence for editing.
- **Level 1 ... 4**: Displays the current value of the sequence. By clicking on it, the sequence can be muted, indicated by the greyed out Level indicator.
- **Value 1 ... 8**: Allows you to set the value for this step. The output is latched, meaning that the pitch value is only sent to the output if the step is enabled.
- **Glide 1 ... 8**: Allows you to turn glide on or off for this step. When Glide is on, the sequencer smoothly transitions between the previous value and the new value. The duration of that transition is set by the **GLIDE** control.
- **Step Enable 1 ... 8**: Allows you to enable or disable the step. When disabled, this step's value is ignored and the previous value is held.
- **GLIDE**: Allows you to set the glide time between notes. This only has an effect when Glide is enabled for the current step.
- **UNI**: Enables unipolar mode for the sequencer (values are set in a range of [0, 1]). When disabled, the sequencer operates in bipolar mode (values are set in a range of [-0.5, 0.5]).
- **RANGE**: Scales the output value range of the sequencer.

Inputs

- **Gate:** A positive zero crossing at this input causes each sequence to advance one step.
- **Reset:** A positive zero crossing at this input causes each sequence to reset.
- **Mod A:** Modulation bus **A** input
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out1 ... 4:** Sequence output. Range [0, 1] when **UNI** is enabled. Range [-0.5, 0.5] when **UNI** is disabled.

11.1.2. 8 Steps (SEQ)



The Bento Box 8 Steps

8 Steps is a fully-featured step sequencer for controlling rhythm and pitch in a patch. Each of its eight steps has settings for **PITCH**, **GLIDE**, **GATE**, and **VELOCITY**. In addition to the **Pitch** and **Gate** outputs, it features a **Vel** output that provides the velocity values as a separate modulation signal. The **DIRECTION** control allows you change the playback behavior of the sequencer, from playback in different directions to random playback. Furthermore, the **STEPS** and **OFFSET** controls let you define the length and position of the sequence, making it possible to create many variations of the same sequence by only changing two parameters. All global parameters can be modulated, which makes the 8 Steps an incredibly powerful tool for creating complex and evolving sequences.

Controls

- **PITCH 1 ... 8:** Allows you to set the pitch value for this step. The **Pitch** output is latched, meaning that the pitch value is only sent to the output if **GATE** and / or **GLIDE** are turned on for this step.
- **GLIDE 1 ... 8:** Allows you to turn glide on or off for this step. When **GLIDE** is on, the sequencer smoothly transitions between the previous pitch and the new pitch. The duration of that transition is set by the **GLIDE** Time control.

- **GATE 1 ... 8**: Allows you to turn the gate on or off for this step. When **GATE** is on, a gate is sent from the **Gate** output whenever the step is selected. The duration of the gate depends on the **GATE** Time control, as well as the **GATE EXT** switch.
- **VELOCITY 1 ... 8**: Allow you to set the strength of the gate for this step. This is useful for Blocks such as the Bento Box Env and the MONARK Env, which can be configured to respond to gates at varying strength. Additionally, velocity values are also sent from the **Vel** output, and can be used as an additional modulation signal in your patch. Similar to the **Pitch** output, the **Vel** output is latched, meaning that the pitch value is only sent to the output if **GATE** is turned on for this step. Otherwise, the **Vel** output value remains unchanged until the next positive gate event occurs.
- **GATE**: Lights up when a gate is received via the **Gate** input, and can be used to manually advance the sequencer one step.
- **RESET**: Clicking this button resets the sequencer to the first step. Additionally, the button indicates when the internal counter is reset when receiving a positive zero crossing at the **Reset** input.
- **DIRECTION**: Allows you to select between six different playback modes:
 - **Forwards**: The sequencer advances from left to right.
 - **Backwards**: The sequencer advances from right to left.
 - **Pendulum**: The sequencer advances forwards until it reaches the final step, at which point it immediately changes direction, advancing backwards until the first step is reached, and so on.
 - **Ping-Pong**: The sequencer advances in the same way as in Pendulum mode, however the first and the last step are repeated when changing direction.
 - **Brownian**: The sequencer advances in a semi-random fashion. Every time a gate is received, the sequencer makes a coin flip decision whether to go forwards or backwards.
 - **Random**: The sequencer advances in a random fashion. Every time a gate is received, the sequencer randomly chooses one of the active steps to advance to.
- **STEPS**: Allows you to set the length of the sequence in number of steps.
- **OFFSET**: Allows you to shift the playback position, including the first step of the sequence (re-set position). This is especially useful when the sequence length is shortened by using the **STEPS** control. If **STEPS** is set to 1, the **OFFSET** parameter provides direct control over the playback position.
- **TRACK**: Enables sequencer pitch tracking. This means that the values sent from the **Pitch** output are transposed using signals arriving at the **Pitch** input. The transposition amount is based on an initial value of 0.5 (middle C). Values less than 0.5 transpose the sequence down, while values greater than 0.5 transpose the sequence up.
- **GATE EXT**: Toggles between internal and external gate modes. In internal gate mode, the sequencer creates its own gate signal, the length of which is set by the **GATE** Time parameter. The **GATE** Time is a percentage of the time measured between consecutive gates arriving at the **Gate** input. In external mode, the sequencer instead passes through the gates arriving at the **Gate** input, meaning that gates sent from the sequencer's **Gate** output have the same duration as gates arriving at the **Gate** input.
- **GATE** Time: Allows you to control the duration of gates sent from the sequencer's **Gate** output in internal gate mode. The range of this control is determined by the interval between gates arriving at the **Gate** input. When the parameter is set fully clockwise, the output gate length is

100% of the time measured between consecutive gates arriving at the **Gate** input. When the parameter is set fully counter-clockwise, output gate length is 1% of the measured time.

- **GLIDE** Time: Allows you to set the glide time between notes. This only has an effect when **GLIDE** is enabled for the current step.

Inputs

- **Pitch**: Input for transposing the pitch output.
- **Gate**: A positive zero crossing at this input causes each sequence to advance one step.
- **Reset**: A positive zero crossing at this input causes each sequence to reset.
- **Mod A**: Modulation bus **A** input
- **Mod B**: Modulation bus **B** input.

Outputs

- **Pitch**: Sequencer pitch output. Range [0, 1].
- **Gate**: Sequencer gate output. Carries velocity information by varying gate strength. Range [0, 1].
- **Vel**: Sequencer velocity output. Range [0, 1].

11.1.3. ADSR Envelope (MOD)



The Bento Box Envelope

The Bento Box Env is an envelope generator with four stages: **ATTACK**, **DECAY**, **SUSTAIN**, and **RELEASE**. The **ATTACK**, **DECAY** and **RELEASE** stages can be individually switched to linear, logarithmic or exponential shapes by clicking on the icons on the right side of the module. This allows you to customize the envelope for specific needs. Each stage has an additional gate output that goes high when the corresponding stage is active. You can use these gate signals to trigger other events in a patch.

Controls

- **GATE**: Lights up when a gate is received via the **Gate** input, and can be used to manually trigger the envelope.

- **VEL**: Activates velocity sensitivity when using a gate signal that includes velocity information, for example from the Bento Box 8 Steps sequencer.
- **ATTACK**: Sets the attack stage duration.
- **DECAY**: Sets the decay stage duration.
- **SUSTAIN**: Sets the sustain stage level.
- **RELEASE**: Sets the release stage duration.
- **ATTACK Mode**: Selects either a linear, exponential, or logarithmic shape for the **ATTACK** stage.
- **DECAY Mode**: Selects either a linear, exponential, or logarithmic shape for the **DECAY** stage.
- **RELEASE Mode**: Selects either a linear, exponential, or logarithmic shape for the **RELEASE** stage.

Inputs

- **Gate**: A positive zero crossing at this input triggers the envelope.
- **Mod A**: Modulation bus **A** input.
- **Mod B**: Modulation bus **B** input.

Outputs

- **Out**: Main Bento Box Env signal output. Range [0, 1].
- **Gate A**: **ATTACK** stage gate output. Range [0 .. 1], [0, 1] if **VEL** is active.
- **Gate D**: **DECAY** stage gate output. Range [0 .. 1], [0, 1] if **VEL** is active.
- **Gate S**: **SUSTAIN** stage gate output. Range [0 .. 1], [0, 1] if **VEL** is active.
- **Gate R**: **RELEASE** stage gate output. Range [0 .. 1], [0, 1] if **VEL** is active.

11.1.4. CV Processor (PRO)



The Bento Box CV Processor

The CVP, or Control Voltage Processor, combines several functions for processing modulation signals. Its three main controls allow you to attenuate and invert (**LEVEL**), offset (**OFFSET**), and smoothen the input signal (**SLEW**). The additional controls in the middle let you switch between a linear and an exponential mode for the **LEVEL** control, as well as choose from a number of clipping and rectification modes that restrict the signal to either positive or negative values. It can be used to condition modulation signals for specific purposes, or to create additional but related versions of existing modulation signals in a patch. It also offers interesting possibilities for wave shaping when used with audio rate signals (clipping and rectification, ring modulation).

Controls

- **LEVEL**: Controls the amount of attenuation applied to the signal. Unlike the LEVEL control on the VCA, this parameter is bipolar. When the control is at its center position, the signal is completely attenuated. Turning the control clockwise increases the signal amplitude in a positive fashion. Turning the control counter-clockwise increases the signal amplitude in a negative fashion, effectively inverting its polarity.
- **OFFSET**: This parameter allows you to add a constant value to the signal, also known as DC bias. The amplitude of the signal remains unaffected, but its average value is increased or decreased accordingly.
- **SLEW**: Lets you smoothen the signal by means of slew rate limiting. Abrupt changes in the signal are shaped into smooth transitions. The greater the amount of slew rate limiting, the longer these transitions take. For example, this can be used for wave shaping, such as turning a square wave into a triangle wave, or for creating a glide effect in a note sequence.
- **Linear / Exponential**: Selects the behavior for the LEVEL control, and any modulation applied to it. Linear means that the amplitude is attenuated in a linear fashion. Exponential means that the amplitude is attenuated in an exponential fashion.
- **Rectification / Clipping**: Allows you to select between various types of clipping and rectification:
 - **Normal**: No clipping or rectification is applied.
 - **Clip low**: Signal is clipped at zero. Values lower than zero are not passed through.
 - **Rectify low**: Signal is rectified at zero. Values lower than zero are inverted.
 - **Clip high**: Signal is clipped at zero. Values greater than zero are not passed through.
 - **Rectify high**: Signal is rectified at zero. Values greater than zero are inverted.

Inputs

- **In**: CVP signal input.
- **Mod A**: Modulation bus **A** input
- **Mod B**: Modulation bus **B** input.

Outputs

- **Out**: CVP signal output.

11.1.5. LFO (MOD)



The Bento Box LFO

The Bento Box LFO is a fully featured low frequency oscillator for modulation purposes. Its straight-forward interface allows you to quickly find the right settings for any basic modulation task, but it also offers advanced features that can be creatively exploited to create very complex modulation signals, especially when modulated by another LFO. Six basic waveforms can be bent into unusual shapes with the **SHAPE** control. The additional options at the top allow you to switch between bipolar and unipolar operation, manually reset the waveform, and hold the current output value. The Hold parameter can be modulated, which makes it possible to stop and hold the LFO with a gate signal, or slice the waveform up in interesting ways by using another LFO.

Controls

- **Frequency:** Adjusts the base frequency of the LFO.
- **Wave:** Selects the type of waveform produced by the LFO. Choose between sine, triangle, saw, ramp, pulse and random. Each waveform can be further shaped by the **SHAPE** control.
- **SHAPE:** Allows for different kinds of additional wave shaping, depending on which waveform is selected:
 - **Sine:** CCW = Hyper exponential wave shaping, CW = Wave folding
 - **Triangle:** CCW = Threshold wave shaping, CW = Wave folding
 - **Saw:** CCW = Exponential wave shaping, CW = Logarithmic wave shaping
 - **Ramp:** CCW = Exponential wave shaping, CW = Logarithmic wave shaping
 - **Pulse:** CCW = Longer negative portion of pulse, CW = Longer positive portion of pulse
 - **Random:** CCW = Exponential smoothing, CW = Linear smoothing
- **PHASE:** Allows you determine at which point in its cycle the LFO restarts when receiving a positive zero crossing at the **Reset** input.
- **Bipolar / Unipolar:** Sets whether the LFO outputs a bipolar signal (+/- 0.5) or a unipolar signal (0/1).
- **Reset:** Clicking this button restarts the LFO from the position in its cycle as determined by the **PHASE** control. Additionally, the button indicates when the LFO is restarted when receiving a positive zero crossing at the **Reset** input.
- **Hold:** The hold button allows you to temporarily pause the LFO and hold its current value. Once the button is released, the LFO continues its cycle.

Inputs

- **Reset:** A positive zero crossing at this input causes the LFO to restart.
- **Mod A:** Modulation bus **A** input
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out:** Bento Box LFO signal output. Range [-0.5, 0.5] in bipolar mode, [0, 1] in unipolar mode.

11.1.6. Mix (MIX)



The Bento Box Mix

The Bento Box Mix is a straight-forward mixer that can be used for any basic mixing task in a patch, for example mixing multiple oscillators or creating feedback loops. Each channel features a level control and a phase inversion switch. You can mute any channel by clicking on the corresponding channel number. The mixer offers two different modes of operation: In absolute mode (**ABS**), the signals of all channels are simply added, driving the mixer into saturation if the maximum output level is reached. In relative mode (**REL**), the output level is attenuated relative to the level of each individual channel in order to avoid clipping of signals.

Controls

- **Level 1 ... 4:** Controls the level of this channel.
- **Mute 1 ... 4:** Mutes this channel.
- **Invert 1 ... 4:** Inverts the signal polarity for this channel.
- **ABS / REL:** Selects one of two different summing behaviors. In **ABS** (absolute) mode, the four channels are mixed together in a traditional fashion before being passed through a saturator. If the combined level exceeds the maximum threshold, saturation occurs. In this case, a clipping indicator appears on the panel. In **REL** (relative) mode, the mixer ensures that the combined level does not exceed the maximum threshold. As the individual levels are increased, the combined level is decreased, ensuring an optimal output level without changing the levels of the individual channels.

Inputs

- **In 1 ... 4:** Channel 1 ... 4 signal input
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out:** Bento Box Mix signal output.

11.1.7. Oscillator (OSC)



The Bento Box Oscillator

The Bento Box Oscillator is a versatile oscillator that is suitable for a wide range of different synthesis methods. Its features make it particularly useful for FM (frequency modulation) synthesis. The linear thru-zero FM mode (**LIN TZ**) allows for very deep and stable FM sounds. The Bento Box Oscillator's **Sync** input can be freely adjusted from soft sync to hard sync, with the option to accept dedicated OSC Sync signals from other Blocks oscillators (e.g. [Multiwave OSC \(OSC\)](#)) for best audio quality. Syncing oscillators is useful for a number of different applications: adding stability to complex FM patches with multiple operators, patching classic oscillator sync sounds, and creating interesting new waveforms by mixing signals from multiple synced oscillators.

Controls

- **Key Tracking:** When key tracking is enabled, you can control the frequency of the oscillator via the **Pitch** input. Additional tuning in semitones and cents is available via the Coarse and Fine controls. When key Tracking is disabled, the oscillator runs at a fixed rate, which can be adjusted in Hz via the Frequency control.
- **FM:** Controls the amplitude of the signal arriving at the **FM** input, also known as the FM index. When the knob is at its lowest setting, the **FM** input signal is fully attenuated and the oscillator's frequency is not being modulated. As the knob is turned clockwise, the amplitude of the signal present at the **FM** input increases, resulting in a greater amount of frequency modulation.
- **EXP / LIN / LIN TZ:** This switch toggles between the three available types of frequency modulation: exponential (**EXP**), linear (**LIN**) and linear through-zero (**LIN TZ**).

- **PW**: Adjusts the width of the pulse waveform.
- **Frequency**: Adjusts the base frequency of the oscillator.
- **Coarse**: Coarse tuning control in semitones.
- **Fine**: Fine tuning control in cents.
- **Wave**: Morphs smoothly between the 4 classic waveforms: sine, triangle, saw and pulse. The result of this morphing is sent from the oscillator's main output. The same 4 waveforms are also available separately via their dedicated outputs.
- **SYNC**: Adjusts the strength of oscillator sync by a signal arriving at the **Sync** input. When **SYNC** is at its lowest setting, no syncing will occur. By increasing the **SYNC** level, different flavors of sync can be dialed in, from soft sync to hard sync. A maximum setting results in classic hard sync sounds.
- **OSC**: Should be enabled when the **Sync** input is fed with a dedicated OSC Sync signal from another Blocks oscillator (e.g. [Multiwave OSC \(OSC\)](#)). For all other audio sources this option needs to be disabled.
- **PHASE**: Allows you determine at which point in its cycle the oscillator restarts when receiving a positive zero crossing at the **Reset** input. This control only works when **RESET** is enabled.
- **RESET**: When enabled, a positive zero crossing in the signal arriving at the **Reset** input causes the oscillator to restart. The restart position can be adjusted with the **PHASE** control.

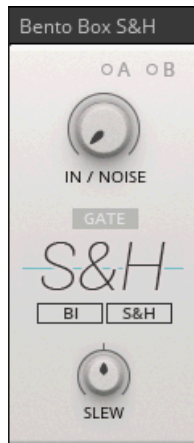
Inputs

- **Pitch**: Input for controlling oscillator pitch. An increase of 0.1 equals a 1 octave increase in pitch (ideal for quantized signals from **Pitch** outputs).
- **FM**: Modulation input for oscillator frequency, amount is set with the **FM** control.
- **Reset**: A positive zero crossing at this input causes the oscillator to restart.
- **Sync**: Input for oscillator synchronization.
- **Mod A**: Modulation bus **A** input
- **Mod B**: Modulation bus **B** input.

Outputs

- **Out**: Main Bento Box signal output.
- **Sine**: Sine wave output.
- **Tri**: Triangle wave output.
- **Saw**: Saw wave output.
- **Pulse**: Pulse wave output.
- **OSC Sync**: Oscillator synchronization signal output.

11.1.8. Sample & Hold (PRO)



The Bento Box Sample & Hold

The Bento Box S&H provides either a sample and hold or a track and hold function. When a gate is received in **S&H** mode, the current value of the input signal is held until the next gate hits. This way, any signal can be converted into a stepped sequence of values that is rhythmically locked to incoming gates. When a gate is received in **T&H** mode, the current value is held for as long as the gate is active. When no gate is active, the input signal is passed through to the output. The top control allows you to blend between the input and an internal noise generator, effectively introducing random values to the signal. When turned fully clockwise, the S&H produces a random sequence of values. The **SLEW** control lets you smoothen the output sequence. Modulating both of these parameters can lead to interesting results.

Controls

- **IN / NOISE**: Cross-fades between the input signal and an internal noise generator. When set fully counter-clockwise, only the input signal is used. Turning the control clockwise introduces random values to the signal. When set fully clockwise, only the internal noise generator is used, resulting in a completely random signal.
- **SLEW**: Lets you smoothen the signal by means of slew rate limiting. Abrupt changes in the signal are shaped into smooth transitions. The greater the amount of slew rate limiting, the longer these transitions take. Turning **SLEW** counter-clockwise creates exponential transitions, while turning **SLEW** clockwise creates linear transitions.
- **GATE**: Lights up when a gate is received via the **Gate** input, and can be used to manually trigger a new sample event.
- **BI / UNI**: Allows you to determine whether the output values are bipolar or unipolar. This affects both the internal noise generator as well as the input signal, which is fully rectified in **UNI** mode.
- **S&H / T&H**: Toggles between sample and hold (**S&H**) and track and hold (**T&H**) mode. In **S&H** mode, a positive gate event at the Gate input causes a new value to be sampled, which is held until another positive gate event occurs. In **T&H** mode, a positive gate event causes a new value to be sampled, however this value is only held for as long as the gate signal remains high. Once the gate signal returns to zero, the input signal is passed through unaffected.

Inputs

- **In**: Bento Box S&H signal input.
- **Gate**: A positive zero crossing at this input causes a new sample event to occur.
- **Mod A**: Modulation bus **A** input
- **Mod B**: Modulation bus **B** input.

Outputs

- **Out**: Bento Box S&H signal output. Range [-1, 1] in bipolar mode, [0, 1] in unipolar mode.

11.1.9. SVF (FLT)



The Bento Box SVF

The SVF is based on the state variable filter design, which has a long history in analog synthesis. Low pass, high pass and band pass filter modes are available straight from the filter core, making the design a popular choice for multi-mode filters. The Bento Box SVF is set up for clean sound and stable operation across the whole **CUTOFF** frequency range, with a controlled **RESONANCE** behavior. Therefore, it is particularly useful if additional coloration of the sound is undesirable. It responds very well to modulation at audio rate, allowing you to explore more extreme sounds by using oscillators or audio feedback to modulate its parameters. The main output's (**Out** port) filter mode can be selected by clicking on the central filter slope icon. All filter modes are also available simultaneously via the **LP**, **HP** and **BP** ports.

Controls

- **CUTOFF**: Sets the cutoff frequency of the filter. Its effect on the sound depends on which filter mode is selected. In low pass mode, frequencies above the cutoff frequency are attenuated. In high pass mode, frequencies below the cutoff frequency are attenuated. In band pass mode, only frequencies around the cutoff frequency are passed through, while all frequencies outside of this range are attenuated.
- **RESONANCE**: Controls the amplitude of the signal at the **CUTOFF** frequency. As resonance increases, the harmonics around the cutoff frequency appear louder and more pronounced than in the original signal. At very high settings, the filter goes into self-oscillation, producing a sine wave on its own. The frequency of this sine wave is set by the **CUTOFF** control.

- **Filter Mode Selector:** Selects the filter mode for the main output (**Out**): low pass, band pass, or high pass.
- **FM:** Controls the amplitude of the signal arriving at the **FM** input. When the knob is at its lowest setting, the **FM** input signal is fully attenuated and the filter's frequency is not being modulated. As the knob is turned clockwise, the amplitude of the signal present at the **FM** input increases, resulting in a greater amount of frequency modulation.
- **Key Tracking:** Allows you to control the cutoff frequency of the filter via the **Pitch** input. This means that you can play the frequency of the filter, just as you can play the frequency of an oscillator. The Key Tracking knob is bipolar, allowing for both positive and negative key tracking. When turned fully clockwise, the filter frequency follows the **Pitch** input accurately. When turned fully counter-clockwise, the filter frequency follows the **Pitch** input inversely.

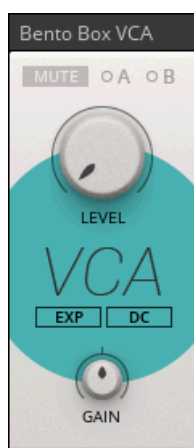
Inputs

- **In:** SVF signal input.
- **Pitch:** Input for filter frequency pitch tracking, amount is set with the Key Tracking control.
- **FM:** Modulation input for filter frequency, amount is set with the **FM** control.
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out:** Main SVF signal output.
- **LP:** Low pass mode signal output.
- **BP:** Band pass mode signal output.
- **HP:** High pass mode signal output.

11.1.10. VCA (AMP)



The Bento Box VCA

The Bento Box VCA is essentially a simple volume control. As one of the basic building blocks in modular synthesis, it is used whenever the level of a signal needs to be controlled dynamically. For example, it is commonly patched as the final stage in a synthesizer patch, modulated by an envelope to shape the amplitude of the sound over time. If used for audio signals, the **AC / DC** switch should be set to **AC**, enabling the DC offset filter. If used for modulation signals, it needs to be set to **DC**, disabling the DC offset filter. Further controls include a switch to select either an exponential (**EXP**) or a linear (**LIN**) response to modulation signals, as well as an additional **GAIN** control that can be used to attenuate or amplify the signal independently of the main level control.

Controls

- **MUTE**: Allows you to completely silence the output of the VCA.
- **LEVEL**: Controls the amount of attenuation applied to the input signal before it is passed on to the output. At maximum setting the signal remains unchanged, while at minimum setting the signal is completely attenuated.
- **LIN / EXP**: Selects the behavior of the **LEVEL** control, and any modulation applied to it. In **LIN** mode, the amplitude is attenuated in a linear fashion. If the level is reduced half way, the signal amplitude is halved. In **EXP** mode, the amplitude is attenuated in an exponential fashion. If the level is reduced half way, the signal amplitude is quartered. In practical terms, **EXP** mode results in a much sharper and snappier response, which is useful for creating drum sounds. Typically, **LIN** mode is used for controlling modulation signals, however it is also useful for audio signals, for example when the VCA is modulated by an exponentially shaped envelope.
- **AC / DC**: Allows you to select whether the VCA is AC or DC coupled. In **DC** mode, any type of signal is passed through, even if DC bias is present. This is especially useful when using the VCA to control modulation signals. In **AC** mode, the VCA eliminates DC bias in the signal, meaning that any signal with DC bias will be resolved back to 0. It is good practice to use **AC** mode with audio signals to avoid potential issues with DC bias in other audio equipment.
- **GAIN**: Allows you to increase or decrease the input signal's volume by up to 12 dB. Excessive gain levels may result in signal clipping. If the signal is clipped, a clipping indicator appears on the panel.

Inputs

- **In**: Bento Box VCA signal input.
- **Mod A**: Modulation bus **A** input.
- **Mod B**: Modulation bus **B** input.

Outputs

- **Out**: Bento Box VCA signal output.

11.1.11. XFade (MIX)



The Bento Box XFade

Bento Box Xfade allows you to create a blend between two input signals with a single **FADE** control. For example, it can be used to smoothly morph from one sound to the other, to combine waveform outputs from oscillators for additional wave shaping, or as a simple dry / wet control when routing effect signals. The Curve control changes the **FADE** response from linear to constant-power, allowing you to compensate for volume drops at the center position.

Controls

- **FADE**: Blends between the signal received at **In 1** and the signal received at **In 2**. Turning the crossfader to the left increases the **In 1** signal level. Turning the crossfader to the right increases the **In 2** signal level.
- **Curve**: Lets you adjust the **FADE** response from linear to constant-power.
- **Gain 1 ... 2**: Adjust the input level for In 1 ... 2, from -12 dB to +12 dB.
- **Invert 1 ... 2**: Inverts the signal polarity for the signal received at In 1 ... 2.

Inputs

- **In 1**: Left Xfade signal input.
- **In 2**: Right Xfade signal input.
- **Mod A**: Modulation bus **A** input.
- **Mod B**: Modulation bus **B** input.

Outputs

- **Out**: XFade signal output.

11.2. Boutique

Boutique Blocks take inspiration from the great synths of the past, bringing the best of the rich history of synthesis to Blocks.



The Boutique line of Blocks is included in the product [Blocks Primes](#), which is included in REAKTOR 6, but can also be purchased separately for use in the free REAKTOR 6 PLAYER.

11.2.1. Dual SKF (FLT)



The Boutique Dual SKF

The SKF is based on the Sallen-Key filter topology, as used in a classic semi-modular monophonic synthesizer from the late seventies. It features a 6 dB/Oct high pass filter and a 12 dB/Oct low pass filter connected in series. By combining the two filters, a variety of different filter responses can be created. This filter strongly colors the sound and features a strong resonance. It is suitable for slightly distorted sounds and electronic percussion when the filters are set to self-oscillation at very high resonance settings.

Controls

- **HP**: Controls the cutoff frequency of the high pass filter. Frequency content below the cutoff frequency is attenuated.
- **LP**: Controls the cutoff frequency of the low pass filter. Frequency content above the cutoff frequency is attenuated.
- **High pass RES**: Sets the resonance amount of the high pass filter. As resonance increases, the harmonics around the high pass cutoff frequency appear louder and more pronounced than in the original signal. At very high settings, the filter goes into self-oscillation, producing a wave on its own. The frequency of this wave is set by the **HP** control.
- **Low pass RES**: Sets the resonance amount of the low pass filter. As resonance increases, the harmonics around the low pass cutoff frequency appear louder and more pronounced than in the original signal. At very high settings, the filter goes into self-oscillation, producing a wave on its own. The frequency of this wave is set by the **LP** control.
- **Key Tracking**: Allows you to control the cutoff frequency of the filter via the **Pitch** input. This means that you can play the frequency of the filter, just as you can play the frequency of an oscillator. The Key Tracking knob is bipolar, allowing for both positive and negative key tracking. When turned fully clockwise, the filter frequency follows the **Pitch** input accurately. When

turned fully counter-clockwise, the filter frequency follows the **Pitch** input inversely. Additional key tracking options are available from the Key Tracking Mode switch below.

- **Key Tracking Mode:** Selects which of the two filters is controlled by the key tracking, as set with the Key Tracking knob. When both arrows are visible, both filters are controlled by signals arriving at the **Pitch** input. If one arrow is visible, only the filter it is pointing at is controlled by signals arriving at the **Pitch** input. Finally, if one of the arrows is hollow, both filters are controlled, but the filter the hollow arrow is pointing at is controlled inversely.
- **FM:** Controls the amplitude of the signal arriving at the **FM** input. When the knob is at its lowest setting, the **FM** input signal is fully attenuated and the filter's frequency is not being modulated. As the knob is turned clockwise, the amplitude of the signal present at the **FM** input increases, resulting in an greater amount of frequency modulation.

Inputs

- **In:** SKF signal input.
- **Pitch:** Input for filter frequency pitch tracking, amount is set with the Key Tracking control.
- **FM:** Modulation input for filter frequency, amount is set with the **FM** control.
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out:** SKF signal output.

11.2.2. Multiwave OSC (OSC)



The Boutique Multiwave OSC

Adapting the oscillator section of a synth classic renowned for its bass sounds, the Multiwave OSC offers four waveforms that can be mixed together by using the faders on the panel: Pulse, Saw, Sub (three different modes), and Noise (white or pink noise). This allows you to create interesting waveform combinations. Additionally, the pulse-width of the Pulse waveform can be modulated to add animation to the sound. The Multiwave OSC is particularly useful for subtractive synthesizer patches where its output is sent to a filter for further sculpting of the sound.

Controls

- **Reset:** When enabled, a positive zero crossing in the signal arriving at the **Reset** input causes the oscillator to restart.
- **Key Tracking:** When key tracking is enabled, you can control the frequency of the oscillator via the **Pitch** input. Additional tuning in semitones and cents is available via the Coarse and Fine controls. When key tracking is disabled, the oscillator runs at a fixed rate, which can be adjusted in Hz via the Frequency control.
- **PW:** Adjusts the width of the pulse waveform.
- **Frequency:** Adjusts the base frequency of the oscillator.
- **Coarse:** Coarse tuning control in semitones.
- **Fine:** Fine tuning control in cents.
- **Pulse Level:** Controls the amplitude of the pulse waveform.
- **Saw Level:** Controls the amplitude of the saw waveform.
- **SUB Level:** Controls the amplitude of the sub waveform.
- **NZ Level:** Controls the amplitude of the noise source.
- **SUB 1 / 2 / 3:** Switches between 3 different modes for the sub oscillator: square wave - 1 octave, square wave -2 octaves, and pulse wave -2 octaves.
- **NZ W / P:** Toggles between white and pink noise

Inputs

- **Pitch:** Input for controlling oscillator pitch. An increase of 0.1 equals a 1 octave increase in pitch (ideal for quantized signals from Pitch outputs).
- **Reset:** A positive zero crossing at this input causes the oscillator to restart.
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out:** Main Multiwave OSC signal output.
- **OSC Sync:** Oscillator synchronization signal output.

11.2.3. OSC 5 (OSC)



The Boutique OSC 5

The OSC 5 is inspired by electronic organs and synthesizers from the seventies. It allows you to blend multiple waveforms at different octaves to create dense sounds with a vintage vibe. Using the rightmost slider (**Level X'**) it is possible to mix in a stacked oscillator voice with additional harmonies, which is useful for creating classic house music chords. Clicking the icons on top of the octave sliders changes the waveform for the corresponding octave. Clicking on the icon for the **Level X'** slider changes the associated chord. The **PW** and **SHAPE** parameters provide additional wave shaping. Modulating these parameters results in very rich and animated sounds.

Controls

- **Key Tracking:** When key tracking is enabled, you can control the frequency of the oscillator via the **Pitch** input. Additional tuning in semitones and cents is available via the Coarse and Fine controls. When key tracking is disabled, the oscillator runs at a fixed rate, which can be adjusted in Hz via the Frequency control.
- **Frequency:** Adjusts the base frequency of the oscillator.
- **Coarse:** Coarse tuning control in semitones.
- **Fine:** Fine tuning control in cents.
- **PW:** Control over the oscillator's pulse width. This parameter affects all 5 of the main oscillators, as well as the stacked oscillator voice. Since the various available waveforms are created using different filtering techniques, the pulse width control always has an effect, regardless of which waveform is selected.
- **SHAPE:** Applies additional wave shaping to the combined waveforms, creating a softer sound and providing further timbral variations. When the shape control is fully clockwise, no shaping is applied.
- **Wave 32' ... 2':** Selects oscillator waveform for (Foot) octave. Available waveforms are square, narrow pulse and ramp.
- **Wave X':** Select chord type for the stacked oscillator voice.
- **Level 32' ... X':** Level control for (Foot) octave, plus the stacked oscillator voice.

Inputs

- **Pitch:** Input for controlling oscillator pitch. An increase of 0.1 equals a 1 octave increase in pitch (ideal for quantized signals from Pitch outputs).
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out:** OSC 5 signal output.

11.3. Digilog

Digilog Blocks are utilities used to create complex rhythms, process notes, and provide structure in patches. They do not generate any sound on their own.



The Digilog line of Blocks is included in the product [Blocks Primes](#), which is included in REAKTOR 6, but can also be purchased separately for use in the free REAKTOR 6 PLAYER.

11.3.1. Clock Divider (PRO)



The Digilog Clock Divider

The Clock Divider is an essential building block for sequencing. It takes any gate signal, for example a clock, and provides six user-definable timing divisions of this signal for further distribution in a patch. This allows you to create a set of related timing signals that are all based on the same tempo, but with a different amount of beats in the same time interval, similar to note divisions. It can be used to run sequencers at different but related speeds, or to set up interesting rhythmical structures within a patch. The ability to modulate the Divisions for each output makes it possible to create very complex polyrhythms.

Controls

- **GATE**: Clicking this button advances the CLK DIV's internal counter by 1. Additionally, the button indicates when the internal counter advances when receiving a positive zero crossing at the **Gate** input.
- **RESET**: Clicking this button causes the CLK DIV's internal counter to reset to 0. Additionally, the button indicates when the internal counter is reset when receiving a positive zero crossing at the **Reset** input.
- **Division 1 ... 6**: Sets the divisor for the associated **Gate** output. This setting determines how many gates arriving at the **Gate** input are counted before a gate is sent from the associated **Gate** output. For example, a Division of 1 means that a gate is sent from the associated **Gate** output for every new gate arriving at the **Gate** input. A Division of 4 means that a gate is sent from the associated **Gate** output for every fourth gate arriving at the **Gate** input.
- **DOWN / UP**: Toggles between two different counting modes, down-counting (**DOWN**) and up-counting (**UP**). In **DOWN** mode, new gates are sent on the first count of the internal counter, while in **UP** mode gates are sent on the last count of the internal counter. For example, in **DOWN** mode a Division of 4 causes new gates to be sent on the count of 1, 5, 9, 13 and so on, while in **UP** mode a Division of 4 causes new gates to be sent on the count of 4, 8, 12, 16, and so on.
- **TRIG / GATE**: Toggles between two different output modes, **TRIG** and **GATE**. In **TRIG** mode, gates sent from the **Gate** outputs have the same duration as the gates arriving at the **Gate** input. In **GATE** mode, the duration of the gates sent from the **Gate** outputs depends on the Division values. The gate remains high for one half of the Division, and low for the other half of the Division. Which half of the Division is high and which half is low depends on the selected **DOWN / UP** counting mode. In **DOWN** mode, the first half of the Division is high, and the second half is low. In **UP** mode, the first half of the Division is low, and the second half is high.

Inputs

- **Gate**: A positive zero crossing at this input advances the CLK DIV's internal counter by 1.
- **Reset**: A positive zero crossing at this input causes the CLK DIV's internal counter to reset to 0.
- **Mod A**: Modulation bus **A** input.
- **Mod B**: Modulation bus **B** input.

Outputs

- **Gate 1**: Division 1 gate output. Range [0 .. 1].
- **Gate 2**: Division 2 gate output. Range [0 .. 1].
- **Gate 3**: Division 3 gate output. Range [0 .. 1].
- **Gate 4**: Division 4 gate output. Range [0 .. 1].
- **Gate 5**: Division 5 gate output. Range [0 .. 1].
- **Gate 6**: Division 6 gate output. Range [0 .. 1].

11.3.2. Quantizer (PRO)



The Digilog Quantizer

The Quantizer is a flexible quantizer that allows you to convert any signal into a stepped sequence of pitches for use with **Pitch** inputs. It automatically maps the input values to a user-definable selection of notes, making it possible to set custom scales. Additionally, **QUANT** sends a gate from the **Gate** output whenever a new value is mapped to one of the selected notes. This can be used to trigger other events in a patch, for example envelopes. Alternatively, **S&H** mode lets you synchronize the quantization to gates arriving at the **Gate** input. This way **QUANT** can be locked to a rhythm that is independent of the input signal.

Controls

- **Notes C ... B**: Selects whether or not the note is active. The QUANT's output only consists of active notes. A signal arriving at the **In** input is transposed to the closest active note.
- **GATE**: Lights up when a gate is received via the **Gate** input, and can be used to manually trigger a new sample event. The **GATE** button has no effect when S&H is disabled.
- **INV**: Inverts the polarity of the signal arriving at the **In** input.
- **S&H**: Enables the sample and hold mode. When S&H is enabled, QUANT only reads the value arriving at the **In** input when a gate is received at the **Gate** input. When S&H is disabled, QUANT continuously reads the value at the **In** input.
- **LEVEL**: Allows you to control the amplitude of the signal arriving at the **In** input. At maximum setting the signal remains unchanged, while at minimum setting the signal is completely attenuated. A stronger signal produces a wider range of values at the QUANT's output.
- **OFFSET**: Introduces an offset to the input signal by adding a constant value before the quantization. The amplitude of the signal remains unaffected, but its average value is increased or decreased accordingly.
- **TRANS**: Introduces an offset to the signal after the quantization. It allows you to shift the output up or down in semitone steps.

Inputs

- **In**: QUANT signal input.
- **Gate**: A positive zero crossing at this input causes a new sample event to occur (**S&H** mode only).

- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out:** Main QUANT signal output. Range [-1, 1].
- **Gate:** Gate output, a trigger is sent for every new quantization value. Range [0 .. 1].

11.4. DRIVER

Create your own custom synth using the finest NI components. The DRIVER Block is a single rack-style module from one of Native Instruments' most renowned effects.



The DRIVER line of Blocks is included in the product [Blocks Primes](#), which is included in REAKTOR 6, but can also be purchased separately for use in the free REAKTOR 6 PLAYER.

11.4.1. DRIVER (EFX)



Driver

DRIVER combines a smooth 12 dB/Oct state-variable filter with a powerful distortion unit and interesting modulation capabilities. It is suitable for a wide range of applications, from light saturation to very animated distortion. The top half of the interface consists of the basic filter and distortion controls. The lower half contains parameters related to modulation, including an envelope follower and an audio modulator. The audio modulator processes signals for modulation of parameters at audio rate, which leads to very aggressive sounds. Both the envelope follower and the audio modulator can be fed by the input signal itself, or by another signal connected to the sidechain inputs (**SC L**, **SC R**).

Controls

- **Input:** Adjusts the amount of amplification or attenuation applied to the incoming audio signal.

- **RES:** Controls the amplitude of the signal at the CUTOFF frequency. As resonance increases, the harmonics around the cutoff frequency appear louder and more pronounced than in the original signal.
- **FREQ:** Sets the cutoff frequency of the low pass filter. Frequency content above the cutoff frequency is attenuated. In NOTCH mode, FREQ sets the center frequency of the notch filter. In this case, all frequencies except those in a narrow band around the cutoff frequency are passed through.
- **DISTO:** Controls the amount of distortion in the output signal.
- **COLOR:** Allows for drastic changes to the distortion tone: with the knob to the left the tone is duller; to the right the tone is brighter.
- **LPF / NOTCH:** Selects one of two available filter modes. In LPF mode, the filter operates as a 12 dB/Oct low pass filter. In this case, frequency content above the cutoff frequency is attenuated. In NOTCH mode, the filter operates as a notch filter that passes through all frequencies except those in a narrow band around the center frequency.
- **Output:** Adjusts the amount of amplification or attenuation applied to the processed signal.
- **Env IN / SC:** Selects the audio input that is processed by the envelope follower. This can be either IN (input, In L and In R ports) or SC (side-chain, SC L and SC R ports).
- **Smooth:** Sets the responsiveness of the envelope. Similar to the attack stage of an envelope, turn the knob left for a smoothed envelope or right for a sharpened envelope.
- **Release:** Sets the release time of the envelope follower.
- **FREQ envelope follower polarity:** Inverts the polarity of the envelope applied to FREQ.
- **FREQ envelope follower slider:** Sets the amount of modulation applied to FREQ from the envelope follower.
- **FREQ audio modulation slider:** Sets the amount of modulation applied to FREQ from the audio modulator.
- **DISTO envelope follower polarity:** Inverts the polarity of the envelope applied to DISTO.
- **DISTO envelope follower slider:** Sets the amount of modulation applied to DISTO from the envelope follower.
- **DISTO audio modulation slider:** Sets the amount of modulation applied to DISTO from the audio modulator.
- **AM IN / SC:** Selects the audio input that is processed by the AM (audio modulator) section. This can be either IN (input, In L and In R ports) or SC (side-chain, SC L and SC R ports).
- **Clip Cut:** Increases the amount of clipping applied to the modulation signal. This can be used to create a more aggressive sound.
- **Lag:** Limits the rate of change of the modulation signal. This can be used to smoothen the edges of the modulation signal.

Inputs

- **In L:** Driver left signal input.
- **In R:** Driver right signal input.
- **SC L:** Driver left side-chain input.
- **SC R:** Driver right side-chain input.
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out L:** Driver left signal output.
- **Out R:** Driver right signal output.

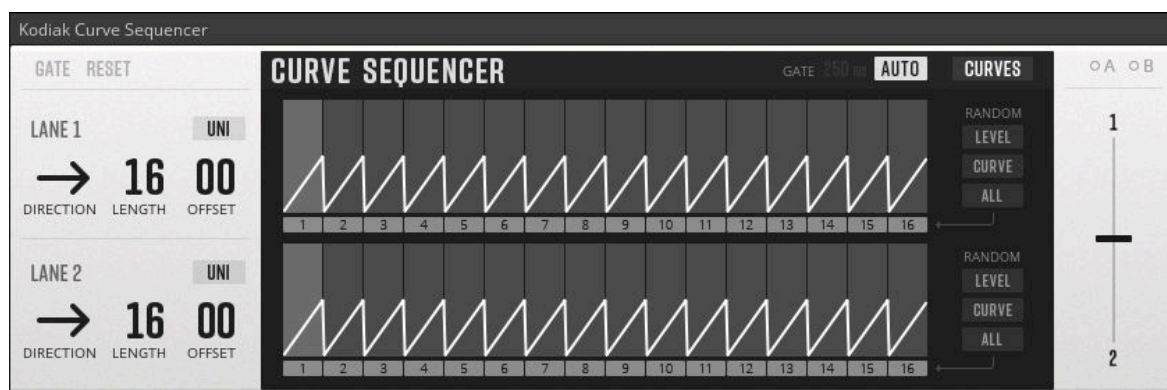
11.5. Kodiak

Kodiak Blocks are forward-thinking modules that combine advanced software concepts with the experience of patching a modular synthesizer.



The Kodiak line of Blocks is included in the product [Blocks Primes](#), which is included in REAKTOR 6, but can also be purchased separately for use in the free REAKTOR 6 PLAYER.

11.5.1. Curve Sequencer (SEQ)



The Kodiak Curve Sequencer

Inspired by MASSIVE's Performer modulation source, the Curve Sequencer takes the idea of sequencing complex envelope shapes to the next level. Two sequences of envelope shapes, or curves, are available as **LANE 1** and **LANE 2**. For each step of the sequence you can choose from a number of curve types and set the level at which the curve is played back. Both **LANE 1** and **LANE 2** have independent controls over **DIRECTION**, **LENGTH**, and **OFFSET**. This allows you to break up the relationship between the two sequences and generate shifting and evolving modulation signals. The two sequences produce either unipolar or bipolar signals, which are sent individually from **Out 1** and **Out 2**, as well as mixed with the 1 / 2 Crossfade control and sent from **Out Mix**. To quickly create variations of a sequence, or entirely new sequences, a comprehensive set of **RANDOM** controls is available, allowing you to randomize the level, the curve type, or both for a selectable number of steps. By default, each step's curve length is automatically set to remain synchronous with the input clock signal. However, it is also possible to set the curve length manually, resulting in unusual envelope shapes. This can also be useful when the sequencer is not controlled by a regular clock signal, but an arbitrary sequence of gates. The Curve Sequencer's panel layout matches the Shift Sequencer, which allows for sequence alignment when the two Blocks are placed on top of each other in Panel view.

Controls

- **GATE:** Lights up when a gate is received via the Gate input, and can be used to manually advance the sequencer one step.
- **RESET:** Clicking this button resets the sequencer to the first step. Additionally, the button indicates when the internal counter is reset when receiving a positive zero crossing at the Reset input.
- **BI / UNI 1 ... 2:** Sets whether this LANE outputs a bipolar signal [-0.5, 0.5] or a unipolar signal [0, 1].
- **DIRECTION 1 ... 2:** Allows you to select between six different playback modes for this LANE:
 - **Forwards:** The sequencer advances from left to right.
 - **Backwards:** The sequencer advances from right to left.
 - **Pendulum:** The sequencer advances forwards until it reaches the final step, at which point it immediately changes direction, advancing backwards until the first step is reached, and so on.
 - **Ping-Pong:** The sequencer advances in the same way as in Pendulum mode, however the first and the last step are repeated when changing direction.
 - **Brownian:** The sequencer advances in a semi-random fashion. Every time a gate is received, the sequencer makes a coin flip decision whether to go forwards or backwards.
 - **Random:** The sequencer advances in a random fashion. Every time a gate is received, the sequencer randomly chooses one of the active steps to advance to.
- **LENGTH 1 ... 2:** Allows you to set this LANE's sequence length in number of steps.
- **OFFSET 1 ... 2:** Allows you to shift this LANE's playback position, including the first step of the sequence (reset position). This is especially useful when the sequence length is shortened by using the LENGTH control. If LENGTH is set to 1, the OFFSET parameter provides direct control over the playback position.
- **Editing Area 1 ... 2:** Allows you to set the level and curve type for each step in this LANE. When the CURVES button is disabled, clicking and dragging with the mouse changes the level. When the CURVES button is enabled, the currently selected curve type is loaded in addition to changing the level.
- **Curve Selector:** Allows you to select a curve type that will be loaded into the LANE's steps when clicking (and dragging) with the mouse in the Editing Area.
- **Random Select:** When enabled, clicking any one of the 3 RANDOM buttons creates new random values for this step.
- **RANDOM LEVEL:** Clicking this button randomizes the level of all steps that have Random Select enabled.
- **RANDOM CURVE:** Clicking this button randomizes the curve type of all steps that have Random Select enabled.
- **RANDOM ALL:** Clicking this button randomizes the level and curve type of all steps that have Random Select enabled.
- **CURVES:** Toggles curve type selection mode for the Editing Area. When enabled, the currently selected curve type is loaded in addition to changing the level of the steps.
- **1 / 2 Crossfade:** Blends between the signal created by LANE 1 (move crossfader up to increase LANE 1 signal level) and the signal created by LANE 2 (move crossfader down to increase LANE 2 signal level). The resulting signal is sent from the Out Mix output.

- **AUTO**: Toggles between automatic and manual curve length modes. When enabled, the curve length matches the time measured between consecutive gates arriving at the Gate input. When disabled, the curve length can be manually set with the GATE Length control.
- **GATE Length**: Sets the curve length in milliseconds when AUTO is disabled.

Inputs

- **Gate**: A positive zero crossing at this input causes the sequencer to advance one step.
- **Reset**: A positive zero crossing at this input causes the sequencer to reset.
- **Mod A**: Modulation bus **A** input.
- **Mod B**: Modulation bus **B** input.

Outputs

- **Out 1**: Sequence 1 signal output. Range [0, 1] when UNI is enabled. Range [-0.5, 0.5] when UNI is disabled.
- **Out 2**: Sequence 2 signal output. Range [0, 1] when UNI is enabled. Range [-0.5, 0.5] when UNI is disabled.
- **Out Mix**: Crossfader signal output. Range [0, 1] when UNI is enabled for both sequences. Range [-0.5, 0.5] when UNI is disabled for either sequence.

11.5.2. Duality Osc (OSC)



The Kodiak Duality Osc

The Duality Osc is an unusual dual oscillator that is particularly well suited for growling and creaking bass sounds, however it is also a good choice for vibrant lead and percussion sounds. Its two internal oscillators, the **MAIN OSC** and the **MOD OSC**, are combined by means of phase modulation (**PM**), frequency modulation (**FM**), and amplitude modulation (**AM**). The **TIMBRE** parameter adds harmonics to the Main output signal by folding the the waveform into itself, which can also be modulated from the **MOD OSC** by increasing the amount of timbre modulation (**TM**). The main tuning controls apply to both oscillators, however the **MAIN OSC**'s frequency can be offset at a ratio set with the **MULT** parameter. This allows you to shape the harmonic structure of your sound when applying modulation with the **PM**, **FM**, **AM**, and **TM** parameters. Adding **SYNC** between the two oscillators further adds to the range of available sounds, either by providing stability (**SYNC >**), or introducing erratic behaviour (**< SYNC**, **X-SYNC**). All parameters are highly complementary and respond well to elaborate modulation routings via the **A** and **B** modulation buses. This makes it easy to find sweet spots and add animation and texture to your sounds. The central display generates a waveform that corresponds to your settings, serving as a visual guide while exploring the Duality Osc's complex wave shaping abilities.

Controls

- **Frequency**: Adjusts the base frequency of the oscillator.
- **COARSE**: Coarse tuning control in semitones.
- **FINE**: Fine tuning control in cents.
- **SHAPE**: Morphs smoothly between the four classic waveforms for the MOD OSC oscillator: sine, triangle, saw and pulse. The result is sent from the Out Mod output, as well as being used for the internal modulation buses.
- **MULT**: Adjusts the frequency of the MAIN OSC as a ratio of the frequency set with the main tuning controls.
- **MULT Quantize**: Quantizes the MULT control to 0.5 sized steps.
- **SYNC**: Sets the behaviour for the oscillator sync between the MAIN OSC and the MOD OSC. **SYNC >**: MAIN OSC is synced to MOD OSC. **< SYNC**: MOD OSC is synced to MAIN OSC. **X-SYNC**: The MAIN OSC and the MOD OSC are cross-synced to each other.
- **PM**: Amount of the internal phase modulation. Controls the amplitude and polarity of the signal arriving internally from the MOD OSC. When the knob is in the center position, the signal is fully attenuated and the MAIN OSC's phase is not being modulated by the MOD OSC. Turning the control clockwise increases the amplitude of the signal in a positive fashion. Turning the control counter-clockwise increases the amplitude of the signal in a negative fashion, effectively inverting its polarity. Increasing the amplitude of the signal results in a greater amount of modulation being applied to the MAIN OSC's phase.
- **FM**: Amount of the internal frequency modulation. Controls the amplitude and polarity of the signal arriving internally from the MOD OSC. When the knob is in the center position, the signal is fully attenuated and the MAIN OSC's frequency is not being modulated by the MOD OSC. Turning the control clockwise increases the amplitude of the signal in a positive fashion. Turning the control counter-clockwise increases the amplitude of the signal in a negative fashion, effectively inverting its polarity. Increasing the amplitude of the signal results in a greater amount of modulation being applied to the MAIN OSC's frequency.
- **AM**: Amount of the internal amplitude modulation. Controls the amplitude and polarity of the signal arriving internally from the MOD OSC. When the knob is in the center position, the signal is fully attenuated and the MAIN OSC's amplitude is not being modulated by the MOD OSC.

Turning the control clockwise increases the amplitude of the signal in a positive fashion. Turning the control counter-clockwise increases the amplitude of the signal in a negative fashion, effectively inverting its polarity. Increasing the amplitude of the signal results in a greater amount of modulation being applied to the MAIN OSC's amplitude.

- **OFFSET:** Applies an offset to the MAIN OSC wavfolder's input signal, introducing asymmetry into the waveform and altering the balance between odd and even harmonics.
- **TIMB MOD:** Amount of the internal TIMBRE modulation. Controls the amplitude and polarity of the signal arriving internally from the MOD OSC. When the knob is in the center position, the signal is fully attenuated and the TIMBRE parameter is not being modulated by the MOD OSC. Turning the control clockwise increases the amplitude of the signal in a positive fashion. Turning the control counter-clockwise increases the amplitude of the signal in a negative fashion, effectively inverting its polarity. Increasing the amplitude of the signal results in a greater amount of modulation being applied to the TIMBRE parameter.
- **TIMBRE:** Increases the amplitude of the MAIN OSC wavfolder's input signal. As the amplitude increases, the wave is repeatedly folded back into itself, resulting in an greater degree of harmonic content and complexity.
- **SUB BOOST:** Mixes a sine wave at the MAIN OSC's frequency with the output of the wavfolder. This can be used to reinforce the fundamental frequency of harmonically rich sounds.

Inputs

- **Pitch:** Input for controlling oscillator pitch. An increase of 0.1 equals a 1 octave increase in pitch (ideal for quantized signals from Pitch outputs).
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out Main:** Main oscillator output.
- **Out Mod:** Modulation oscillator output.

11.5.3. Flip Gen (OSC)



The Kodiak Flip Gen

Flip Gen is a peculiar tone generator and noise source. It can produce bleepy sequences, morse signals, pitched noise, crackly textures, and anything in between. Furthermore, it can serve as a modulation source, allowing you to introduce intricate, chaotic, or even random movement to any parameter in your patch. At audio rates, this will add noisy artifacts to your sound, however you can also create slow modulation signals by enabling **LFO** mode. At its core, the Flip Gen consists of a sine wave oscillator that repeatedly switches between two frequencies **FREQ A** and **FREQ B**. With the **MIX** control it is possible to set the level of the output signal depending on which frequency is selected. The rate of switching is controlled by the **SPEED** control, with the option to introduce **INSTABILITY** for random fluctuations of the rate. By turning the **SPEED** control clockwise, you can gradually morph the Flip Gen's basic bleep and morse sounds into colorful noises. The switching can either introduce clicks and pops to the signal, or be rounded off by turning up the **SMOOTH** control. To further steer the Flip Gen's chaotic behaviour, the **CHANCE** control allows you to set the probability of selecting either **FREQ A** or **FREQ B** when switching occurs.

Controls

- **LFO**: Enables LFO mode for the internal sine wave oscillator.
- **FREQ A**: Adjusts base frequency FREQ A of the internal sine wave oscillator.
- **FREQ B**: Adjusts base frequency FREQ B of the internal sine wave oscillator.
- **MIX**: Adjusts the relative level of the internal sine wave oscillator when switching between frequencies FREQ A and FREQ B. Moving the MIX control to the left increases the level at FREQ A, and decreases the level of the at FREQ B. Moving the MIX control to the right increases the level at FREQ B, and decreases the level at FREQ A. When the MIX control is in its middle position, the level remains the same at both FREQ A and FREQ B.
- **CHANCE**: Adjusts the probability of selecting either FREQ A or FREQ B when switching occurs. Moving the CHANCE control to the left increases the probability of selecting FREQ A, and decreases the probability of selecting FREQ B. Moving the CHANCE control to the right increases the probability of selecting FREQ B, and decreases the probability of selecting FREQ A.

When the CHANCE control in its middle position, FREQ A and FREQ B are selected at equal probability.

- **SPEED:** Adjusts the rate of switching between FREQ A and FREQ B.
- **INSTABILITY:** Destabilizes the rate of switching. When INSTABILITY is at its lowest setting, switching occurs at a stable rate set by the SPEED control. As the knob is turned clockwise, the rate begins to fluctuate with an increasing amount of deviation from the SPEED value.
- **SMOOTH:** Allows you to smoothen the switching between FREQ A and FREQ B by means of slew rate limiting. Abrupt changes of frequency and level are shaped into smooth transitions. The greater the amount of slew rate limiting, the longer these transitions take.

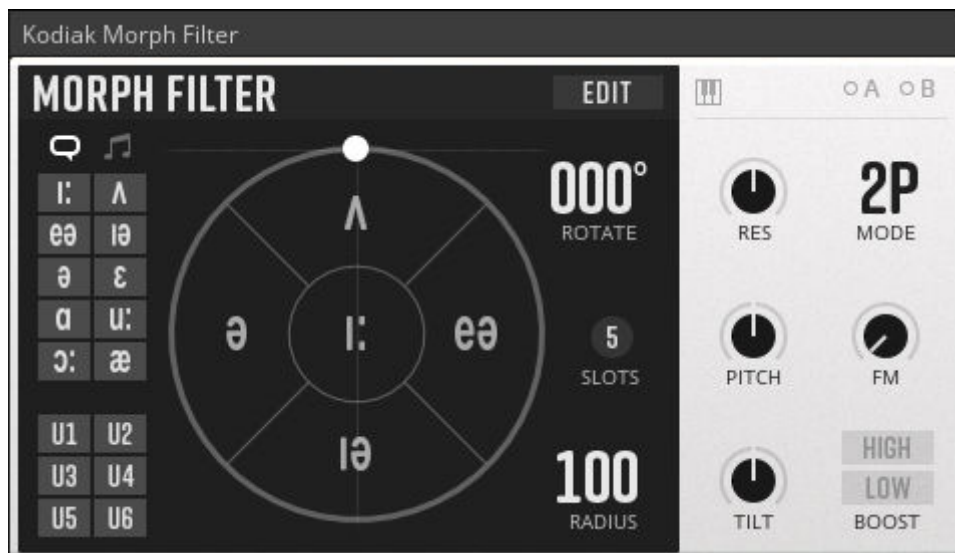
Inputs

- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out:** Main Flip Gen signal output.
- **Pitch:** Flip Gen pitch output. Outputs the current frequency FREQ A or FREQ B as a pitch signal. Range [0, 1].
- **Amp:** Flip Gen amplitude output. Outputs a value representing the relative level of the internal sine wave oscillator at frequencies FREQ A and FREQ B. Range [0, 1].
- **Gate:** Flip Gen gate output. Sends a gate when the frequency FREQ A is active. Range [0 .. 1].

11.5.4. Morph Filter (FLT)



The Kodiak Morph Filter

The Morph Filter combines a pristine sounding filter bank with a sophisticated interface for morphing between filter settings. Its preset system allows you to quickly load filter settings that space the three available filter bands at specific intervals, forming vowels or chords. The Morphing Area can be used to smoothly blend between these filter settings, giving you access to a huge pallet of sound with only two controls, **ROTATE** and **RADIUS**. In **EDIT** view, you can create and customize filter settings, with the possibility to save them as your own presets. The **MODE** switch gives you access to three different filter responses (**2P**, **4P**, **8P**), which drastically change the Morph filter's character. A global **PITCH** control and Key Tracking let you play the filters in a harmonically related manner. It's characteristic sound and unique resonance behaviour make the Morph Filter an exceptional tool for sculpting and animating harmonically rich oscillator signals as well as noise. Taking this to the extreme, you can for example turn any noise signal into a lush pad sound by increasing the resonance (**RES** control) in **8P** mode and loading chord presets to the Morphing Area. Beyond that, the Morph Filter can also become a sound generator in its own right by sending hard-edged signals (e.g. triggers and gates) to its inputs and turning up the **RES** control, setting the resonance at the brink of self-oscillation. This so-called 'pinging' produces smoothly decaying sine wave signals that are a great source for tonal percussion sounds.

Controls

- **Vowel / Chord**: Switches the Preset Bank between vowels and chords.
- **Factory Preset Bank**: Allows you to choose from a number of presets containing filter settings that can be loaded to a slot in the Morphing Area by dragging and dropping with the mouse. You also also drag and drop them to the Filter Plot in EDIT view.
- **User Preset Bank**: Allows you to choose from a number of presets containing filter settings that can be loaded to one or more slots in the Morphing Area by dragging and dropping with the mouse. You can also drag and drop them onto the Filter Plot in EDIT view. Presets can be saved to the User Preset Bank by using the SAVE button in EDIT view.
- **Morphing Area**: Every slot in the Morphing Area holds a set of filter settings. You can morph between them by using the ROTATE and RADIUS controls, or by clicking and dragging with the mouse in the Morphing Area. The number of slots in the Morphing Area is set by the SLOTS control. Presets containing filter settings can be loaded from the Factory Preset Bank and the User Preset Bank by simply dragging and dropping a preset directly onto a slot in the Morphing Area. You can create or customize filter settings in EDIT view by double-clicking on any slot, or by clicking the EDIT button.
- **ROTATE**: Controls the position in the Morphing Area in a circular motion.
- **RADIUS**: Controls the position in the Morphing Area relative to its center.
- **SLOTS**: Sets the number of slots available in the Morphing Area.
- **Key Tracking**: Allows you to control the cutoff frequencies of the filter bands via the Pitch input. This means that you can play the frequencies of the filters, just as you can play the frequency of an oscillator.
- **RES**: Global resonance control. As resonance increases, the harmonics around the cutoff frequencies appear louder and more pronounced than in the original signal. At high settings, when the filters are on the brink of self-oscillation, they can be 'pinged' by sending hard-edged signals (e.g. triggers and gates) to the Morph Filter's inputs. This produces smoothly decaying sine wave signals that are a great source for percussion sounds.
- **PITCH**: Global pitch control. Transposes the frequency of all filter bands by up to 12 semitones up or down.

- **TILT:** Controls the relative levels of all three filter bands. Turning the knob counter-clockwise increases the level of filter band B1 while reducing the level of filter bands B2 and B3. Turning the knob clockwise increases the level of filter band B3, while reducing the level of filter bands B1 and B2. When the knob is in its middle position, the 3 filter bands are at the same level.
- **MODE:** This switch allows you to select one of three different filter responses, allowing you drastically change the Morph Filter's character. The filter response determines the amount of attenuation applied to frequencies above and below the cutoff frequency. The following responses are available:
 - **2P:** State variable filter with a relatively soft response, attenuating frequencies above and below the cutoff frequency by 6 dB/Oct.
 - **4P:** Two cascaded state variable filters, attenuating frequencies above and below the cutoff frequency by 12 dB/Oct.
 - **8P:** Butterworth filter with an aggressive response, attenuating frequencies above and below the cutoff frequency by 24 dB/Oct.
- **FM:** Controls the amplitude of the signal arriving at the FM input. When the knob is at its lowest setting, the FM input signal is fully attenuated and the filter frequencies is not being modulated. As the knob is turned clockwise, the amplitude of the signal present at the FM input increases, resulting in an greater amount of frequency modulation.
- **HIGH BOOST:** Switches filter band B3 to high pass mode, emphasizing high frequencies.
- **LOW BOOST:** Switches filter band B1 to low pass mode, emphasizing low frequencies.
- **EDIT:** Enables EDIT view, which allows you to create and customize filter settings for any of the slots in the Morphing Area, as well as load and save presets. When EDIT view is enabled, morphing is temporarily disabled, allowing you to monitor the exact filter settings for the selected slot.
- **Freq / Pitch:** Toggles value entry for the cutoff frequencies of the three filter bands between frequency and pitch. This option can be set per slot.
- **B1:** Sets the cutoff frequency of filter band B1.
- **B2:** Sets the cutoff frequency of filter band B2.
- **B3:** Sets the cutoff frequency of filter band B3.
- **Filter Plot:** Displays the filter settings for the selected slot. Double-clicking the Filter Plot disables EDIT view.
- **SLOT:** Selects which slot in the Morphing Area is currently selected for editing.
- **SAVE:** Settings made in EDIT view can be saved to any one of the six presets in the User Preset Bank. To do so, first click the SAVE button. The User Preset Bank becomes highlighted. Next, click the preset in which you would like to save your filter settings. After saving, your preset is ready to be loaded to one or more slots in the Morphing area at any time. You can also load it in EDIT view by dragging and dropping onto the Filter Plot for further adjustments.

Inputs

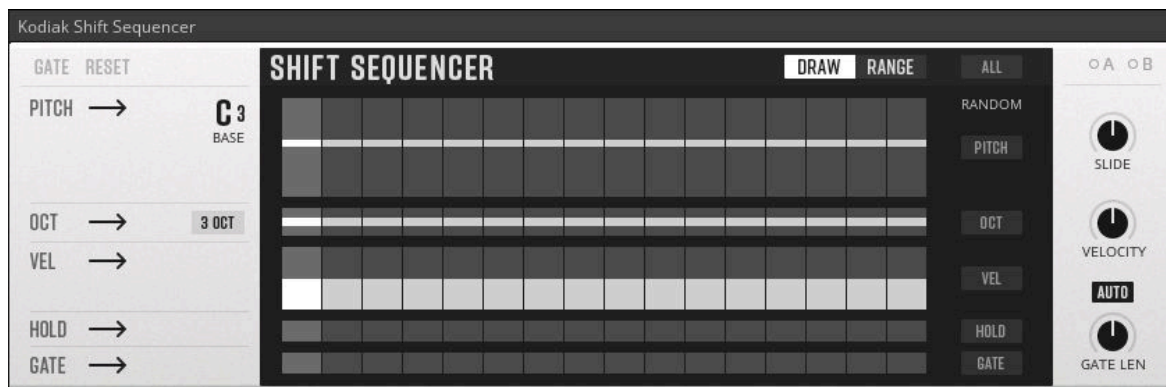
- **In 1:** Filter band 1 signal input. Normalled to In 2 and In 3. This way a signal arriving at In 1 is also sent to In 2 and In 3 when no connection is made to those inputs.
- **In 2:** Filter band 2 signal input. Normalled to In 1. Making a connection to this input break the normalization from In 1.
- **In 3:** Filter band 2 signal input. Normalled to In 1. Making a connection to this input break the normalization from In 1.

- **Pitch**: Input for global filter frequency pitch tracking when the Key Tracking button is enabled.
- **FM**: Modulation input for global filter frequency, amount is set with the FM control.
- **Mod A**: Modulation bus **A** input.
- **Mod B**: Modulation bus **B** input.

Outputs

- **Out Mix**: Combined signal output for filter bands B1, B2, and B3.
- **Out 1**: Filter band B1 signal output.
- **Out 2**: Filter band B2 signal output.
- **Out 3**: Filter band B3 signal output.

11.5.5. Shift Sequencer (SEQ)



The Kodiak Shift Sequencer

Bringing an entirely new take on step sequencing to Blocks, the Shift Sequencer allows you to sequence different attributes of a note independently from another. This can be used to create poly-rhythms within notes by letting the step values of each attribute shift against each other. Separate lanes for **PITCH**, octave (**OCT**), velocity (**VEL**), **HOLD**, and **GATE** each have their own controls for playback direction, as well as sequence length and position offset in **RANGE** view. These controls can be used to explore a multitude of subtly evolving or drastically transformed sequences based on the same set of step values. Value entry in steps can be quickly achieved by clicking and dragging across the lanes, and a mouse-over piano roll display in the **PITCH** lane makes it easy to enter specific notes. As an alternative to manually entering step values for each lane in **DRAW** view, a set of **RANDOM** functions is available. This way you can randomize the values for all active steps in a sequence, not only for all lanes at once but also separately for each lane. Additional functions include variable glide (**SLIDE**) for **HOLD** events, velocity output scaling, and a gate length control (**GATE LEN**) with optional **AUTO** mode. The **Play** input allows you to control sequencer playback externally. This can be used in conjunction with the Util Clock's **Play** output to prevent hanging notes when stopping the clock, or exploited creatively, for example by starting and stopping playback from a keyboard. The Shift Sequencer's panel layout matches the Curve Sequencer, which allows for sequence alignment when the two Blocks are placed on top of each other in Panel view.

Controls

- **GATE:** Lights up when a gate is received via the Gate input, and can be used to manually advance the sequencer one step.
- **RESET:** Clicking this button resets the sequencer to the first step. Additionally, the button indicates when the internal counter is reset when receiving a positive zero crossing at the Reset input.
- **Direction PITCH / OCT / VEL / HOLD / GATE:** Allows you to select between six different playback modes for this lane:
 - **Forwards:** The sequencer advances from left to right.
 - **Backwards:** The sequencer advances from right to left.
 - **Pendulum:** The sequencer advances forwards until it reaches the final step, at which point it immediately changes direction, advancing backwards until the first step is reached, and so on.
 - **Ping-Pong:** The sequencer advances in the same way as in Pendulum mode, however the first and the last step are repeated when changing direction.
 - **Brownian:** The sequencer advances in a semi-random fashion. Every time a gate is received, the sequencer makes a coin flip decision whether to go forwards or backwards.
 - **Random:** The sequencer advances in a random fashion. Every time a gate is received, the sequencer randomly chooses one of the active steps to advance to.
- **BASE:** Sets the base key for the mouse-over piano roll in the PITCH lane and transposes the Pitch output accordingly. If a signal is connected to the Pitch input, the BASE control is greyed out. In this case, the values sent from the Pitch output are transposed using signals arriving at the Pitch input. The transposition amount is based on an initial value of 0.5 (middle C). Values less than 0.5 transpose the sequence down, while values greater than 0.5 transpose the sequence up.
- **Octave Range:** Allows you to set the number of octaves available on the OCT lane.
- **DRAW / RANGE:** Toggles between DRAW view and RANGE view. DRAW view allows you to create and edit values for all lanes. RANGE view allows you to set the sequence length and position for all lanes, enabling and disabling steps accordingly.
- **PITCH Lane:** In DRAW view, the PITCH lane allows you to set the pitch value for each step. The pitch value is combined with the octave value set in the OCT lane. The Pitch output is latched, meaning that the pitch value is only sent to the output if a GATE and / or HOLD event occurs. In RANGE view, you can set the sequence length and position for the PITCH lane, enabling and disabling steps accordingly. The highlighted area represents the enabled steps of the sequence. Clicking and dragging the center of this area (< > symbol is shown) sets the position. Clicking and dragging the left border (< symbol is shown) sets the first step of the sequence. Clicking and dragging the right border (> symbol is shown) sets the last step of the sequence.
- **OCT Lane:** In DRAW view, the OCT lane allows you to set the octave value for each step. The octave value is combined with the pitch value set in the PITCH lane. The Pitch output is latched, meaning that the pitch value is only sent to the output if a GATE and / or HOLD event occurs. In RANGE view, you can set the sequence length and position for the OCT lane, enabling and disabling steps accordingly. The highlighted area represents the enabled steps of the sequence. Clicking and dragging the center of this area (< > symbol is shown) sets the position.

Clicking and dragging the left border (< symbol is shown) sets the first step of the sequence. Clicking and dragging the right border (< symbol is shown) sets the last step of the sequence.

- **VEL Lane:** In DRAW view, the VEL lane allows you to set the strength of the gate for each step. Additionally, velocity values are also sent from the Vel output, and can be used as an additional modulation signal in your patch. Similar to the Pitch output, the Vel output is latched, meaning that the velocity value is only sent to the output if a GATE event occurs. Otherwise, the Vel output value remains unchanged until the next positive gate event occurs. In RANGE view, you can set the sequence length and position for the VEL lane, enabling and disabling steps accordingly. The highlighted area represents the enabled steps of the sequence. Clicking and dragging the center of this area (< > symbol is shown) sets the position. Clicking and dragging the left border (< symbol is shown) sets the first step of the sequence. Clicking and dragging the right border (< symbol is shown) sets the last step of the sequence.
- **HOLD Lane:** In DRAW view, the HOLD lane allows you to turn hold and / or glide on or off for each sequencer step. A HOLD event ties successive gates, meaning the gate output remains high from one step to the next. Additionally, the sequencer smoothly transitions to the next step's pitch. The duration of that transition is set by the GLIDE Time control. In RANGE view, you can set the sequence length and position for the HOLD lane, enabling and disabling steps accordingly. The highlighted area represents the enabled steps of the sequence. Clicking and dragging the center of this area (< > symbol is shown) sets the position. Clicking and dragging the left border (< symbol is shown) sets the first step of the sequence. Clicking and dragging the right border (< symbol is shown) sets the last step of the sequence.
- **GATE Lane:** In DRAW view, the GATE lane allows you to turn the gate on or off for each sequencer step. When GATE is on, a gate is sent from the Gate output whenever the step is selected. The duration of the gate depends on the GATE Time control. In RANGE view, you can set the sequence length and position for the GATE lane, enabling and disabling steps accordingly. The highlighted area represents the enabled steps of the sequence. Clicking and dragging the center of this area (< > symbol is shown) sets the position. Clicking and dragging the left border (< symbol is shown) sets the first step of the sequence. Clicking and dragging the right border (< symbol is shown) sets the last step of the sequence.
- **RANDOM ALL:** Clicking this button randomizes the values of all enabled steps in all lanes.
- **RANDOM PITCH:** Clicking this button randomizes the values of all enabled steps in the PITCH lane.
- **RANDOM OCT:** Clicking this button randomizes the values of all enabled steps in the OCT lane.
- **RANDOM VEL:** Clicking this button randomizes the values of all enabled steps in the VEL lane.
- **RANDOM HOLD:** Clicking this button randomizes the values of all enabled steps in the HOLD lane.
- **RANDOM GATE:** Clicking this button randomizes the values of all enabled steps in the GATE lane.
- **SLIDE:** Allows you to set the glide time between pitches. This only has an effect when a HOLD event occurs.
- **VELOCITY:** Allows you to scale the values sent from the VEL lane. This affects both the strength of the Gate output, as well as the values sent from the Vel output.
- **GATE LEN:** Allows you to control the length of gates sent from the Gate output. When AUTO is enabled, the range of this control is determined by the interval between gates arriving at the

Gate input. When the parameter is set fully clockwise, the output gate length is 400% of the time measured between consecutive gates arriving at the Gate input. When the parameter is set fully counter-clockwise, output gate length is 1% of the measured time. When AUTO is disabled, the output gate length can be set manually from 1 ms to 1 s.

- **AUTO:** Toggles between automatic and manual gate length modes. When enabled, the gate length can be set to a duration based on the interval between consecutive gates arriving at the Gate input. When disabled, the curve length can be set in ms manually via the GATE LEN control.

Inputs

- **Pitch:** Input for transposing the pitch output.
- **Gate:** A positive zero crossing at this input causes the sequencer to advance one step.
- **Reset:** A positive zero crossing at this input causes the sequencer to reset. Additionally, it resets any active HOLD event.
- **Play:** A constant gate at this input enables sequencer playback, allowing it to advance one step every time a gate is received at the Gate input. When the gate at the Play input drops to 0 or below, playback is disabled and gate events at the Gate input are ignored. If the sequencer stops with an active HOLD event, it gets terminated. Used in conjunction with the Util Clock's Play output, this prevents hanging notes when stopping the clock. Playback is constantly enabled when no connection is made to the Play input.
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Pitch:** Sequencer pitch output. Range [0, 1].
- **Gate:** Sequencer gate output. Carries velocity information by varying gate strength. Range [0, 1].
- **Vel:** Sequencer velocity output. Range [0, 1].
- **Hold:** Hold gate output. Sends a gate when the sequencer advances to a step with a HOLD event. Range [0 .. 1].

11.6. Modern

Modern Blocks are state-of-the-art-modules with a contemporary twist. They represent the latest developments in the modular world, showcasing specific functions that are out of the ordinary.



The Modern line of Blocks is included in the product [Blocks Primes](#), which is included in REAKTOR 6, but can also be purchased separately for use in the free REAKTOR 6 PLAYER.

11.6.1. Comb Filter (FLT)



The Modern Comb Filter

The Comb Filter delays the input signal and feeds it back onto itself, similar to an echo effect. However, the Comb Filter is optimized for very short delay times, causing interferences in the audible range. It creates regularly spaced peaks and troughs in the frequency response, resembling the appearance of a comb. The effect gets more pronounced as the feedback is increased. Apart from filtering oscillator signals, the Comb Filter can also be used to create effects like flanging and chorus, or serve as the basis for physical modelling sounds. If Key Tracking is turned up, it can be controlled with a pitch signal arriving at the **Pitch** input, making it possible to play the Comb Filter's frequency according to a musical scale.

Controls

- **FM:** Controls the amplitude of the signal arriving at the FM input. When the knob is at its lowest setting, the FM input signal is fully attenuated and the filter's frequency is not being modulated. As the knob is turned clockwise, the amplitude of the signal present at the FM input increases, resulting in a greater amount of frequency modulation.
- **Key Tracking:** Allows you to control the frequency of the filter via the Pitch input. This means that you can play the frequency of the filter, just as you can play the frequency of an oscillator. The Key Tracking knob is bipolar, allowing for both positive and negative key tracking. When turned fully clockwise, the filter frequency follows the Pitch input accurately. When turned fully counter-clockwise, the filter frequency follows the Pitch input inversely.
- **FREQUENCY:** Sets the size of the delay buffer inside of the Comb. A larger delay buffer causes a longer delay time, which results in a lower filter frequency. A smaller delay buffer causes a shorter delay time, which results in a higher filter frequency.
- **DAMP:** Controls the cutoff frequency of a low pass filter in the internal feedback loop of the Comb. Increasing the DAMP value attenuates the high frequency content of the feedback signal, resulting in a softened sound at the output.
- **FEEDBACK:** Sets the level of the internal feedback loop of the Comb. Higher settings cause a stronger feedback signal, resulting in a longer resonance of the filter. This control is bipolar, which allows you to set both positive and negative FEEDBACK levels. Negative settings invert the polarity of the feedback signal, resulting in inharmonic content.

Inputs

- **In**: Comb signal input.
- **Pitch**: Input for filter frequency pitch tracking, amount is set with the Key Tracking control.
- **FM**: Modulation input for filter frequency, amount is set with the FM control.
- **Mod A**: Modulation bus **A** input.
- **Mod B**: Modulation bus **B** input.

Outputs

- **Out**: Comb signal output.

11.6.2. Paul Filter (FLT)



The Modern Paul Filter

The Paul Filter is a smooth sounding low pass filter, adapting the filter design of early polyphonic analog synthesizers. Its resonance behavior has been optimized to provide a warm and full sound: as you turn up the **RESONANCE**, the bass is boosted. You can click on the F-clef icon in the bottom-right corner to deactivate the bass boost. This gives you an even bass response that is independent of the **RESONANCE** setting. Four filter modes allow you to change the overall sound of the filter, from bright (**LP1**) to boomy (**LP4**). This filter is particularly useful for all kinds of bass sounds, from rumbling sub-bass sounds to liquid sounding acid basslines.

Controls

- **CUTOFF**: Sets the cutoff frequency of the low pass filter. Frequency content above the cutoff frequency is attenuated. The degree of attenuation is determined by the setting of the Mode switch.
- **RESONANCE**: Controls the amplitude of the signal at the CUTOFF frequency. As resonance increases, the harmonics around the cutoff frequency appear louder and more pronounced than in the original signal. At very high settings, the filter goes into self-oscillation, producing a sine wave on its own. The frequency of this sine wave is set by the CUTOFF control.
- **Key Tracking**: Allows you to control the cutoff frequency of the filter via the Pitch input. This means that you can play the frequency of the filter, just as you can play the frequency of an

oscillator. The Key Tracking knob is bipolar, allowing for both positive and negative key tracking. When turned fully clockwise, the filter frequency follows the Pitch input accurately. When turned fully counter-clockwise, the filter frequency follows the Pitch input inversely.

- **FM:** Controls the amplitude of the signal arriving at the FM input. When the knob is at its lowest setting, the FM input signal is fully attenuated and the filter's frequency is not being modulated. As the knob is turned clockwise, the amplitude of the signal present at the FM input increases, resulting in a greater amount of frequency modulation.
- **Mode:** This switch allows you to select different filter responses. The filter response determines the amount of attenuation applied to frequencies above the cutoff frequency, measured in decibels per octave (dB/Oct). The following responses are available:
 - **LP1:** The softest response, content above the cutoff frequency is attenuated by 6 dB/Oct. This means a frequency one octave greater than the cutoff frequency is attenuated by 6 dB, a frequency two octaves greater than the cutoff frequency is attenuated by 12 dB, and so on.
 - **LP2:** Slightly steeper response, content above the cutoff frequency is attenuated by 12 dB/Oct.
 - **LP3:** Even steeper response, with a response of 18 dB/Oct. Finally, LP4 is the most aggressive response, with frequencies above the cutoff frequency being attenuated by 24 dB/Oct.
- **Bass Boost:** At high resonance settings, low pass filters can cause a reduction of low frequency content. Often this is a desirable result, but on other occasions you might wish to preserve the low frequency content of your signal. When Bass Boost is switched on, additional processing is activated within the filter to ensure that low frequency content is preserved, even at highest resonance settings.

Inputs

- **In:** Paul signal input.
- **Pitch:** Input for filter frequency pitch tracking, amount is set with the Key Tracking control.
- **FM:** Modulation input for filter frequency, amount is set with the FM control.
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out:** Paul signal output.

11.7. MONARK

Create your own custom synth using the finest NI components. MONARK Blocks are single rack-style modules from one of Native Instruments' most renowned synths.



The MONARK line of Blocks is included in the product [Blocks Primes](#), which is included in REAKTOR 6, but can also be purchased separately for use in the free REAKTOR 6 PLAYER.

11.7.1. MONARK ADS ENV (MOD)



The Monark ADS ENV

Modelled after the envelope section of an iconic monophonic synthesizer from the seventies, the MONARK ADS Env captures every nuance of the original circuit, including the unique behavior of its switchable release stage. Additionally, the **MODE** switch allows you to choose between the amplitude or filter envelope characteristics of the original instrument. Renowned for its punchy response, this envelope is a good choice for strong bass and lead sounds as well as snappy percussion.

Controls

- **ATTACK**: Sets the duration of the attack stage of the envelope.
- **DECAY**: Sets the duration of the decay stage of the envelope.
- **SUSTAIN**: Sets the level of the sustain stage of the envelope.
- **MODE**: Selects one of two different envelope modes. Mode A is modelled after the amplitude envelope of the original instrument this envelope is based on, mode F is modelled after the filter envelope.
- **REL**: Sets the duration of the release stage of the envelope. It can either be a fraction of the decay time (off) or equal to the decay time (on).

Inputs

- **Gate**: A positive zero crossing at this input triggers the envelope.
- **Mod A**: Modulation bus **A** input.
- **Mod B**: Modulation bus **B** input.

Outputs

- **Out**: MONARK Env signal output.

11.7.2. MONARK Filter (FLT)



The Monark Filter

Modelled after the classic 4-pole ladder low pass filter of an iconic monophonic synthesizer from the seventies, the MONARK Filter captures every nuance of the original circuit, including its rich saturation behavior. This can be explored even more with the additional **LOAD** and **FEEDBACK** parameters, allowing you to overdrive the input and add nonlinear behavior to the filter. The MONARK Filter offers four different filter modes, including three low pass modes and one band pass mode, all derived from the classic ladder structure. This filter is an excellent choice for huge sounding synth bass and lead sounds.

Controls

- **FM:** Controls the amplitude of the signal arriving at the FM input. When the knob is at its lowest setting, the FM input signal is fully attenuated and the filter's frequency is not being modulated. As the knob is turned clockwise, the amplitude of the signal present at the FM input increases, resulting in a greater amount of frequency modulation.
- **CUTOFF:** Sets the cutoff frequency of the filter. Its effect on the sound depends on which filter mode is selected. In low pass mode, frequencies above the cutoff frequency are attenuated. In band pass mode, only frequencies around the cutoff frequency are passed through, while all frequencies outside of this range are attenuated.
- **FEEDBACK:** Sets the amount of additional feedback being applied to the filter. The type of feedback depends on the setting of the A / B Feedback Type switch.
- **A / B Feedback Type:** Selects one of two different feedback types. Type A is an authentic reproduction of one of the feedback paths available on the original instrument this filter is modeled after. It can totally distort the signal and is suitable for very aggressive sounds. Type B is more subtle and can be used to achieve a range of warm timbres.
- **RESONANCE:** Controls the amplitude of the signal at the CUTOFF frequency. As resonance increases, the harmonics around the cutoff frequency appear louder and more pronounced than in the original signal. At very high settings, the filter goes into self-oscillation, producing a wave on its own. The frequency of this wave is set by the CUTOFF control.
- **LOAD:** Increases the input signal level. Turn the knob fully counter-clockwise for an authentic sound. When the LOAD knob is turned clockwise, more overdrive is introduced to the sound.

- **KT1:** Allows you to control the cutoff frequency of the filter via the Pitch input. This means that you can play the frequency of the filter, just as you can play the frequency of an oscillator. The first KT switch adds 1/3 of key tracking to filter cutoff frequency.
- **KT2:** Allows you to control the cutoff frequency of the filter via the Pitch input. This means that you can play the frequency of the filter, just as you can play the frequency of an oscillator. The second KT switch adds 2/3 of key tracking to filter cutoff frequency.
- **Mode:** Selects one of four different filter modes: MM (24 dB/Oct low pass), LP2 (12 dB/Oct low pass), LP1 (6 dB/Oct low pass), BP (12 dB/Oct band pass).

Inputs

- **In:** MONARK Filter signal input.
- **Pitch:** Input for filter frequency pitch tracking, amount is set with the KT switches.
- **FM:** Modulation input for filter frequency, amount is set with the FM control.
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out:** MONARK Filter signal output.

11.7.3. MONARK Oscillator (OSC)



The Monark OSC

Modelled after the oscillators of an iconic monophonic synthesizer from the seventies, the MONARK OSC captures every nuance of the original circuit, including all the inaccuracies that contribute to its distinctive sound. All three oscillators from the Native Instruments MONARK synthesizer are available and can be selected by clicking on the **OSC 1** label at the top of the panel. This way the unique characteristics of all three oscillators can be used to create very rich and lively sounds, especially when mixing together multiple instances of the MONARK Oscillator.

Controls

- **OSC 1 / 2 / 3:** There are three different oscillator models available in the Monark OSC, each with its own unique character and behaviour. This can prove especially useful when using mul-

tiple instances of the Monark OSC within one patch. When combined, these subtle differences help to produce some incredibly rich and lively sounds.

- **KT:** When key tracking is enabled, you can control the frequency of the oscillator via the Pitch input. Additional tuning is available at the FREQUENCY and RANGE controls.
- **FREQUENCY:** Changes the pitch of the oscillator across a range of -7.5 to +7.5 semitones. When KT is disabled and OSC 3 is active, the FREQUENCY knob's range is increased.
- **RANGE:** Sets the pitch range of the oscillator. The settings of 32', 16', 8', 4' and 2' are equivalent to octave standards based on organ stops. The LO setting puts the oscillator into sub-audio range.
- **FM:** Controls the amplitude of the signal arriving at the FM input, also known as the FM index. When the knob is at its lowest setting, the FM input signal is fully attenuated and the oscillator's frequency is not being modulated. As the knob is turned clockwise, the amplitude of the signal present at the FM input increases, resulting in a greater amount of frequency modulation.
- **WAVEFORM:** Selects the oscillator waveform. Available waveforms are triangle, sawtooth, sawtooth triangular (or reverse sawtooth when OSC 3 is active) square, pulse, narrow pulse and a very narrow pulse.

Inputs

- **Pitch:** Input for controlling oscillator pitch. An increase of 0.1 equals a 1 octave increase in pitch (ideal for quantized signals from Pitch outputs).
- **FM:** Modulation input for oscillator frequency, amount is set with the FM control.
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out:** MONARK OSC signal output.

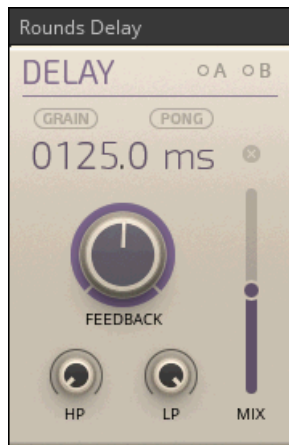
11.8. ROUNDS

Create your own custom synth using the finest NI components. ROUNDS Blocks are single rack-style modules from one of Native Instruments' most renowned synths.



The ROUNDS line of Blocks is included in the product [Blocks Primes](#), which is included in REAKTOR 6, but can also be purchased separately for use in the free REAKTOR 6 PLAYER.

11.8.1. ROUNDS Delay (EFX)



The ROUNDS Delay

The ROUNDS Delay can produce a wide range of different echo, flanging and chorus effects. It offers two basic modes of operation, normal and **GRAIN**. In normal mode, changing the delay time momentarily alters the pitch of the signal, similar to tape echo effects. In **GRAIN** mode, the pitch is not altered. The **PONG** option lets the delay repetitions alternate between the left and right channel for a wide stereo effect. By clicking on the **ms** unit label next to the Delay Time, you can synchronize the ROUNDS Delay externally to REAKTOR's master tempo or internally to a sequence of gates from within the Blocks patch.

Controls

- **GRAIN**: Enables GRAIN mode. In normal mode, changing the delay time momentarily alters the pitch of the signal, similar to tape echo effects. In GRAIN mode, the pitch is not altered.
- **PONG**: Lets the delay repetitions alternate between the left and right channel for a wide stereo effect. When PONG is not active, the left and right channels are processed independently.
- **Delay Time**: Sets the size of the delay buffer. A larger delay buffer causes a longer delay time, a smaller delay buffer causes a shorter delay time. The delay time can be adjusted in ms or in note divisions, depending on the setting of the ms / th switch.
- **ms / th**: Selects between time mode (ms) and synced mode (th) for the delay. In time mode, the Delay Time is adjusted in milliseconds. In synced mode (th), the delay is synchronized to the tempo and the Delay Time is adjusted in note divisions (e.g. 1/04, 1/08, 3/16). The tempo reference can either be REAKTOR's master tempo or a sequence of gates received at the Gate input, depending on the setting of the SYNC INT / SYNC EXT switch.
- **SYNC INT / SYNC EXT**: Sets the synchronization reference for synced mode (th). When SYNC INT is selected, the delay can be synchronized to a sequence of gates received at the Gate input. When SYNC EXT is selected, the delay is synchronized to REAKTOR's master tempo (or the tempo of the host sequencer when using REAKTOR as a plug-in).
- **FEEDBACK**: Changes the level of the delay feedback. Higher feedback levels result in a longer decay of the delay repetitions.
- **Mute**: Switches muting on or off for the input of the delay effect.
- **MIX**: Adjusts the amount of delay signal added to the input signal.
- **HP**: Adjusts the damping of high frequencies in the delay signal (high pass filter).

- **LP:** Controls the low frequency content in the delay signal (low pass filter).

Inputs

- **In L:** ROUNDS Delay left signal input.
- **In R:** ROUNDS Delay right signal input.
- **Gate:** Positive zero crossings arriving at this input are timed, providing a tempo which delay times can be synchronized to.
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out L:** ROUNDS Delay left signal output.
- **Out R:** ROUNDS Delay right signal output.

11.8.2. ROUNDS LFO (MOD)



The ROUNDS LFO

The ROUNDS LFO brings the LFO (Low Frequency Oscillator) from the Modulation section of the ROUNDS synthesizer to Blocks. Just like the [ROUNDS Delay \(EFX\)](#), it features a tempo-sync option. This lets you synchronize the LFO externally to REAKTOR's master tempo or internally to a sequence of gates from within the Blocks patch, making it possible to create modulation signals that are always locked to the beat of your music.

Controls

- **Wave:** Selects the type of waveform produced by the LFO. Choose between triangle, saw, square, and random.
- **BI / UNI:** Sets whether the LFO outputs a bipolar signal (+/- 0.5) or a unipolar signal (0/1).
- **RESET:** When enabled, a positive zero crossing at the Reset input causes the LFO to restart. When disabled, a signal at the Reset input is ignored.
- **Rate:** Adjusts the rate of the LFO. Depending on the mode selected with the FREE / GATE / HOST button, you can set a frequency in Hz (FREE) or note divisions (GATE, HOST).

- **FREE / GATE / HOST:** Selects between three different synchronization modes for the LFO.
 - **FREE:** The LFO is not externally synced. The rate can be freely adjusted in Hz with the Rate control.
 - **GATE:** The LFO is synchronized to the tempo derived from a sequence of gates received at the Gate input. The rate is adjusted in note divisions (e.g. 1/04, 1/08, 3/16).
 - **HOST:** The LFO is synchronized to REAKTOR's master tempo. The rate is adjusted in note divisions (e.g. 1/04, 1/08, 3/16).

Inputs

- **Gate:** Positive zero crossings arriving at this input are timed, providing a tempo which the LFO can be synchronized to (in GATE synchronization mode).
- **Reset:** A positive zero crossing at this input causes the LFO to restart (if RESET is enabled).
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out:** ROUNDS LFO signal output. Range [-0.5, 0.5] in BI mode, [0, 1] in UNI mode.

11.8.3. ROUNDS Reverb (EFX)



The Rounds Reverb

The ROUNDS Reverb is based on a feedback delay network with eight delay taps. It can produce a wide range of effects: from big, deep and cloudy halls to small, resonant rooms or grainy delays. All parameters have been optimized for modulation via the **A** and **B** modulation busses, so you can morph from one space into the other, create very animated textures, or add interesting rhythmic effects to the reverb sound. This makes the ROUNDS Reverb suitable not only for adding space to a sound, but also for inserting it at any point in your patch as part of the sound generation.

Controls

- **DIFFUSION:** Adjusts the density of reverb reflections.

- **SIZE:** Adjusts the size of the reverb room.
- **SPREAD:** Spreads the delay times between the eight delay taps. This is good for resonant and metallic sounds when SIZE and SPREAD are at low levels.
- **FEEDBACK:** Controls the decay time of the reverb reflections.
- **Mute:** Switches muting on or off for the input of the reverb effect.
- **MIX:** Adjusts the amount of reverb signal added to the input signal.
- **CENTER:** Controls the delay time of all eight delay taps at once. This adds density and depth to the reverb sound.
- **HP:** Adjusts the damping of high frequencies in the reverberated signal (high pass filter).
- **LP:** Controls the low frequency content in the reverberated signal (low pass filter).

Inputs

- **In L:** ROUNDS Reverb left signal input.
- **In R:** ROUNDS Reverb right signal input.
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out L:** ROUNDS Reverb left signal output.
- **Out R:** ROUNDS Reverb right signal output.

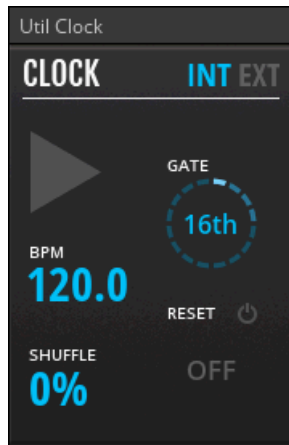
11.9. Util

Util – short for Utility – Blocks are the modules that help make larger patches possible. Like Digilog Blocks, they don't generate sound on their own, but make the entire world of Blocks possible.



The Util line of Blocks is included in the product Blocks Base, which you can download for free as part of [KOMPLETE START](#).

11.9.1. Clock (INT)



The Util Clock

Clock provides a clock signal as a sequence of gates via its **Gate** output. It allows you to run Blocks that rely on external timing information, for example sequencers. For this, the Clock's **Gate** output is connected to the **Gate** input of a sequencer. The tempo is set in beats per minute (**BPM**), with a definable note division for the sequence of gates (**4th**, **8th**, and so on). Additionally, you can add some groove by turning up **SHUFFLE**. The **Reset** output sends a gate every time Clock is activated. When the **RESET** function is enabled, reset gates are also sent while the clock is running, at an interval set by the Reset Interval control on the panel. Reset gates are useful for synchronizing multiple sequencers in a patch.

Controls

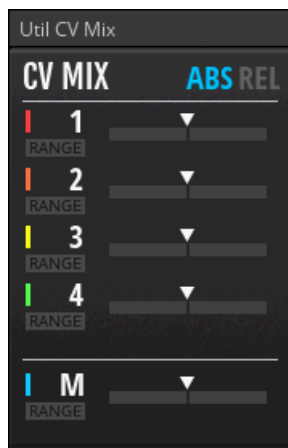
- **INT EXT**: Toggles between the internal clock and the external clock, for example from a host sequencer. In INT mode, start / stop and the BPM can be set with the controls on the panel. In EXT mode, start / stop and the BPM are controlled by the host sequencer.
- **Play**: Activates Clock and causes it to start sending a sequence of gates from its Gate output. The tempo can be adjusted with the BPM and GATE controls. Additionally, Clock sends a gate from the Reset output every time it is activated. This can be used to reset sequencers, ensuring that they start from the correct step.
- **BPM**: Sets the tempo for the sequence of gates created by Clock, measured in beats per minute. In EXT mode, the BPM control becomes a display, showing the tempo it receives from the host sequencer.
- **GATE**: Allows you to set the number of gates sent for every beat. This parameter is available in both INT and EXT mode. Typically, 16th notes provide the best results when working in both internal and external modes. However, different divisions can for example allow you to run sequencers at half or double the tempo of the host sequencer.
- **RESET**: Sends out additional gates from the Reset output when enabled. The interval between these reset gates is determined by the Reset Interval parameter.
- **Reset Interval**: Allows you to set the interval between gates sent from the Reset output (when RESET is enabled). Aside from synchronizing sequencers, this is useful for creating unusual time signatures.
- **SHUFFLE**: Allows you to add shuffle, or swing, to the sequence of gates sent out from the Gate output. Shuffle is achieved by creating rhythmical unevenness between odd and even

numbered gates. The shuffle control is bipolar, meaning that both positive and negative shuffle can be added. Positive shuffle means that even numbered gates occur later than normal, while negative shuffle means that even numbered gates occur earlier than normal.

Outputs

- **Gate:** Main Clock gate output. Range [0 .. 1].
- **Reset:** Reset gate output. Sends a gate when Clock is activated. If RESET is enabled, additional gates are sent at an interval specified by the Reset Interval control.

11.9.2. CV Mix (MIX)



The Util CV Mix

CV Mix is a specialized mixer for processing modulation signals. It offers four channels and a master section, which all feature a versatile control slider that can be configured as a standard bipolar level control or a special range control. This makes it easy to sculpt modulation signals in new and interesting ways. The CV Mix offers two different modes of operation: In absolute mode (**ABS**), the signals of all channels are simply added. In relative mode (**REL**), the output level is attenuated relative to the level of each individual channel in order to keep signals in the range from -1 to 1. The color selectors let you assign distinct colors to different parts or signal chains in a patch. The CV Mix can also be used to mix audio signals, for example when feeding it multiple waveforms from the same oscillator for external wave shaping.

Controls

- **ABS REL:** Selects one of two different summing behaviors. In ABS (absolute) mode, the four channels are simply added together. The output is not clipped, meaning the combined level can exceed the range from -1 to 1. In REL (relative) mode, the mixer ensures that the combined level does not exceed the range from -1 to 1. As the individual levels are increased, the combined level is decreased, ensuring an optimal output level without changing the levels of the individual channels.
- **RANGE:** Activates range mode for this level control. In this mode, the level control turns into a dual control that allows you to set the minimum value (bottom control) and maximum value (top control) for the input signal across the negative and positive range. By clicking and dragging the meter bar both controls can be moved at the same time.

- **Level 1 ... 4:** Allows you to adjust the level of this channel, either non-inverted (slider is moved to the right) and inverted (slider is moved to the left).
- **Min Value 1 ... 4 / M:** Sets the minimum value for this channel / the main output.
- **Max Value 1 ... 4 / M:** Sets the maximum value for this channel / the main output.
- **Shift 1 ... 4 / M:** Moves both the Min Value and Max Value simultaneously, without changing their relation to each other.
- **Mute 1 ... 4:** Mutes this channel.
- **Mute M:** Mutes the main output.
- **Color 1 ... 4 / M:** Sets the color for this channel meter / the main level meter.

Inputs

- **In 1:** Channel 1 signal input.
- **In 2:** Channel 2 signal input.
- **In 3:** Channel 3 signal input.
- **In 4:** Channel 4 signal input.

Outputs

- **Out:** CV Mix signal output.

11.9.3. Gates & Trigs (INT)



The Util Gates & Trigs

Gates & Trigs allows you to create or modify gate and trigger signals. Apart from sending these signals to Gate inputs in your Blocks patch, you can use them to control external hardware synthesizers with analog gate CV (Control Voltage) inputs via DC-coupled audio interfaces or D/A converters. The outputs are optimized to facilitate this use case with improved compatibility, allowing you to directly route the signals to your DC-coupled audio interface or D/A converter. Gates & Trigs accepts any signal and derives gates and triggers from it based on the settings made with the Threshold sliders. This way you can condition gate and trigger signals for a specific use, or create interesting sequences of gates and triggers based on arbitrary signals.

Controls

- **Mute 1 ... 4:** Mutes this channel.
- **Color 1 ... 4:** Sets the color for this channel. The colors help to visually organize patches.
- **Gate High Threshold 1 ... 4:** Sets the gate high threshold for this channel. When the input signal rises above this value, the **Gate** output goes high.
- **Gate Low Threshold 1 ... 4:** Sets the gate low threshold for this channel. When the input signal falls below this value, the **Gate** output goes low.
- **Shift 1 ... 4:** Moves both the Gate Low Threshold and the Gate High Threshold simultaneously, without changing their relation to each other.
- **Trigger Threshold 1 ... 4:** Sets the trigger threshold for this channel. When the input signal rises above this value, a trigger is sent from the **Gate** output.
- **TRIG 1 ... 4:** Enables trigger mode for this channel. In trigger mode, the **Gate** output goes high for a duration set with the Trigger Length control whenever the input signal rises above the Trigger Threshold.
- **Trigger Length 1 ... 4:** Sets the duration of the trigger sent from the **Gate** output in trigger mode.
- **INV:** Inverts the signal sent from the **Gate** output. This is useful for sending gates or triggers to external synthesizers with an S-Trig input.

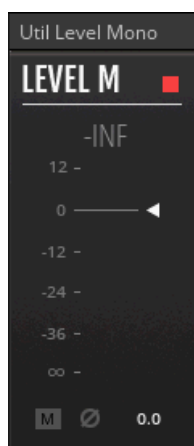
Inputs

- **In 1 ... 4:** Channel 1 ... 4 signal input.

Outputs

- **Out 1 ... 4:** Channel 1 ... 4 gate output. Range [0 .. 1].

11.9.4. Level Mono (MIX)



The Util Level Mono

Level M is a mono volume control including level metering, with additional switches for mute and phase inversion. It is intended to be used as the first Block after an **In Port** Module or the last Block before an **Out Port** module, however it may also be used to monitor and control signals at any point in a patch. The color selector in the upper right corner of the panel lets you assign distinct colors to different parts or signal chains in a patch.

Controls

- **Level:** Controls the output level.
- **Peak Level:** Displays the peak level.
- **Level Meter:** Displays the current signal level on a scale from +12 dBfs to $-\infty$ dBfs.
- **Level Value:** Displays the value of the Level control.
- **Mute:** Mutes the input.
- **Invert:** Inverts the signal polarity.
- **Color:** Sets the color for this Level M instance.

Inputs

- **In:** Level M signal input.

Outputs

- **Out:** Level M signal output.

11.9.5. Level Stereo (MIX)



The Util Level Stereo

LEVEL S is a stereo volume control including level metering, with additional switches for mute, mono summing and phase inversion. It is intended to be used as the first Block after an **In Port** module or the last Block before an **Out Port** module, however it may also be used to monitor and control signals at any point in a patch. The color selector in the upper right corner of the panel lets you assign distinct colors to different parts or signal chains in a patch.

Controls

- **Level:** Controls the output level.
- **Peak Level:** Displays the peak level.
- **Level Meter:** Displays the current signal level on a scale from +12 dBfs to $-\infty$ dBfs.
- **Level Value:** Displays the value of the Level control.
- **Mute:** Mutes the input.
- **Mono Sum:** Sums the stereo input to mono.
- **Invert:** Inverts the signal polarity.
- **Color:** Sets the color for this Level S instance.

Inputs

- **In L:** Level S left signal input.
- **In R:** Level S right signal input.

Outputs

- **Out L:** Level S left signal output.
- **Out R:** Level S right signal output.

11.9.6. Macro Knobs (INT)



The Util Macro Knobs

Macro Knobs provides four knobs that can be connected to any modulation input in a patch, allowing for central control of multiple parameters. This is useful for creating controller mappings, distributing host automation data, as well as managing modulation signals in a patch. The knobs can be switched from unipolar operation (sending out only positive values) to bipolar operation (sending out negative and positive values).

Controls

- **UNI / BI 1:** Sets whether the Macro knob outputs a unipolar signal [0, 1] or a bipolar signal [-0.5, 0.5].

- **UNI / BI 2:** Sets whether the Macro knob outputs a unipolar signal [0, 1] or a bipolar signal [-0.5, 0.5].
- **UNI / BI 3:** Sets whether the Macro knob outputs a unipolar signal [0, 1] or a bipolar signal [-0.5, 0.5].
- **UNI / BI 4:** Sets whether the Macro knob outputs a unipolar signal [0, 1] or a bipolar signal [-0.5, 0.5].
- **Macro Knob 1:** Sets the value sent from the Out 1 output.
- **Macro Knob 2:** Sets the value sent from the Out 2 output.
- **Macro Knob 3:** Sets the value sent from the Out 3 output.
- **Macro Knob 4:** Sets the value sent from the Out 4 output.

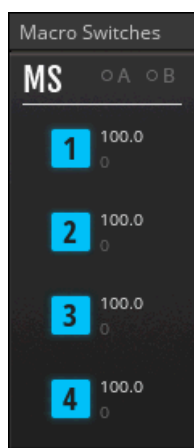
Inputs

- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out 1:** Macro knob 1 signal output. Range [0, 1] in UNI mode, [-0.5, 0.5] in BI mode.
- **Out 2:** Macro knob 2 signal output. Range [0, 1] in UNI mode, [-0.5, 0.5] in BI mode.
- **Out 3:** Macro knob 3 signal output. Range [0, 1] in UNI mode, [-0.5, 0.5] in BI mode.
- **Out 4:** Macro knob 4 signal output. Range [0, 1] in UNI mode, [-0.5, 0.5] in BI mode.

11.9.7. Macro Switches (INT)



The Util Macro Switches

Macro Switches provides four buttons that can be connected to any modulation input in a patch, allowing for central control of multiple parameters. This is useful for creating controller mappings, distributing host automation data, as well as managing modulation signals in a patch. The values that are sent out in the off state as well as the on state can be freely adjusted.

Controls

- **1:** Switches the signal sent from the Out 1 output between the values set by the On Value 1 and Off Value 1 parameters.
- **2:** Switches the signal sent from the Out 2 output between the values set by the On Value 2 and Off Value 2 parameters.
- **3:** Switches the signal sent from the Out 3 output between the values set by the On Value 3 and Off Value 3 parameters.
- **4:** Switches the signal sent from the Out 4 output between the values set by the On Value 4 and Off Value 4 parameters.
- **On Value 1:** Sets the value sent from the Out 1 output when Switch 1 is on.
- **On Value 2:** Sets the value sent from the Out 2 output when Switch 2 is on.
- **On Value 3:** Sets the value sent from the Out 3 output when Switch 3 is on.
- **On Value 4:** Sets the value sent from the Out 4 output when Switch 4 is on.
- **Off Value 1:** Sets the value sent from the Out 1 output when Switch 1 is off.
- **Off Value 2:** Sets the value sent from the Out 2 output when Switch 2 is off.
- **Off Value 3:** Sets the value sent from the Out 3 output when Switch 3 is off.
- **Off Value 4:** Sets the value sent from the Out 4 output when Switch 4 is off.

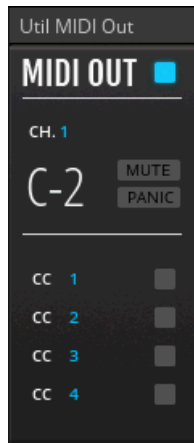
Inputs

- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out 1:** Macro switch 1 signal output. Range as set with the On Value 1 and Off Value 1 parameters.
- **Out 2:** Macro switch 2 signal output. Range as set with the On Value 2 and Off Value 2 parameters.
- **Out 3:** Macro switch 3 signal output. Range as set with the On Value 3 and Off Value 3 parameters.
- **Out 4:** Macro switch 4 signal output. Range as set with the On Value 4 and Off Value 4 parameters.

11.9.8. MIDI Out (INT)



The Util MIDI Out

The Midi Out allows you to use Blocks for controlling MIDI software or hardware from your Blocks patch. It converts pitch and gate signals to MIDI notes and offers four channels of MIDI CC output. This way any modulation signal from within your Blocks patch can be sent to your MIDI destination as Control Change messages.

To choose an external destination for the MIDI output (for example another software application, a MIDI interface, or a USB-MIDI synthesizer):

- Select the Block and go to the Side Pane > Properties > **Connect** > **MIDI OUT** > **External**.

To choose an internal destination for the MIDI output (for example a REAKTOR Instrument):

- Select the Block and go to the Side Pane > Properties > **Connect** > **MIDI OUT** > **Internal**.

Controls

- **Color**: The color selector helps to visually organize patches, for example when using multiple MIDI Outs for different MIDI destinations.
- **CH.:** Selects the MIDI channel for the MIDI output.
- **Note Display**: Shows the current MIDI note output.
- **MUTE**: Mutes the MIDI output.
- **PANIC**: Momentarily stops and resets the MIDI output.
- **CC 1 ... 4**: Mutes the output for this MIDI CC channel.
- **CC Number 1 ... 4**: Chooses the controller number for this MIDI CC channel.
- **CC Output**: Visualizes the output for this MIDI CC channel.

Inputs

- **Pitch**: Input for the pitch signal that is converted to MIDI notes.
- **Gate**: Input for the gate signal that is converted to MIDI notes.
- **CV 1 ... 4**: Signal input for MIDI CC channel 1 ... 4. The signal is converted to MIDI control change messages.

11.9.9. Mix 4 (MIX)



The Util Mix 4

Mix 4 is a stereo mixer with 4 channels, offering volume control and panning including level metering, with additional switches for mute and mono summing per channel. The **Aux** inputs allow you to chain multiple instances to create larger mixers. Each channel has a fixed attenuation of -3 dB to provide a useful amount of headroom. The color selectors let you assign distinct colors to different parts or signal chains in a patch. MIX 4 is very useful as final mixer in patches with multiple voices.

Controls

- **Level 1 ... 4:** Controls the level of this channel.
- **Level Main:** Controls the main output level.
- **Peak Level 1 ... 4:** Displays the peak level for this channel.
- **Peak Level Main:** Displays the peak level for the main output.
- **Level Meter 1 ... 4:** Displays the current level for this channel on a scale from +12 dBfs to -∞ dBfs.
- **Level Meter Main:** Displays the current level for the main output on a scale from +12 dBfs to -∞ dBfs.
- **Level Value 1 ... 4:** Displays the value of the Level 1 ... 4 control.
- **Level Value Main:** Displays the value of the Level Main control.
- **Pan 1 ... 4:** Controls the panning for this channel.
- **Mute 1 ... 4:** Mutes this channel.
- **Mono Sum 1 ... 4:** Sums the stereo input of this channel to mono.
- **Mute Main:** Mutes the main output.
- **Color 1 ... 4 / M:** Sets the color for this channel meter / the main level meter.
- **Level Meter Aux:** Displays the current levels for the signals arriving at the Aux inputs.

Inputs

- **In 1L:** Channel 1 left signal input.
- **In 1R:** Channel 1 right signal input.

- **In 2L**: Channel 2 left signal input.
- **In 2R**: Channel 2 right signal input.
- **In 3L**: Channel 3 left signal input.
- **In 3R**: Channel 3 right signal input.
- **In 4L**: Channel 4 left signal input.
- **In 4R**: Channel 4 right signal input.
- **Aux L**: Auxiliary left signal input.
- **Aux R**: Auxiliary right signal input.

Outputs

- **Out L**: Mix 4 left signal output.
- **Out R**: Mix 4 right signal output.

11.9.10. Note In (INT)



The Util Note In

The Note In receives MIDI note data and converts it into control signals for use in your Blocks patch. It allows you to control the pitch of oscillators according to MIDI notes, trigger envelopes with note on messages, and modulate parameters with the modulation wheel or after touch. The **Pitch** output carries all pitch information including MIDI note, pitch bend and glide, while the **Note** output only carries the MIDI note information. In a similar fashion, the **Gate** output provides gates that combine note on and note off messages with velocity, while the **Vel** output only provides velocity information. Pitch bend is also available separately via the **Pbend** output, as are the modulation wheel (**Mwheel**) and after touch (**Atouch**).

Controls

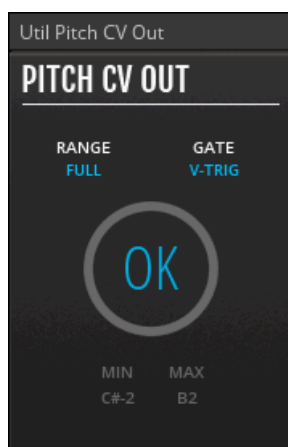
- **CH.:** Selects which MIDI channel the Note In receives note data from. Numbers 1-16 correspond to the standard 16 MIDI channels. OMNI sets the Note In to receive note data from all 16 channels, while INTERN refers to REAKTOR's internal MIDI bus.
- **MUTE:** Mutes all of the Note In's outputs.
- **PB:** Allows you to ignore all pitch bend data. This is useful when multiple Note Ins are used to create key zones on the same MIDI Channel.

- **Pitch Bend Range:** Sets the maximum pitch bend range in semitones.
- **GLIDE:** Sets the glide duration. The glide behaviour depends on whether LEGATO mode is enabled or not.
- **LEGATO:** When enabled, a note on message only generates a new gate event if no other note on message is active at the same time. When disabled, all note on messages generate a new gate event. The LEGATO mode also has an affect on glide behavior. When enabled, glide only occurs if a note on message is received while another note on message is still active. When disabled, glide occurs for every note on message.
- **PRIO:** Since the Note In is monophonic, whenever two or more notes are played simultaneously, one note must be prioritized over the other. The available options are LOW, HIGH, and LAST. In LOW mode, the lowest note is given priority. In HIGH mode, the highest note is given priority. In LAST mode, the most recent note is given priority.
- **RANGE:** Allows you to specify the active keyboard zone, along with the HI and LO parameters. When enabled, only note events in this zone are received, and any note events outside of this zone are ignored. When disabled, the Note In receives all note events.
- **HI:** Sets the highest note of the active keyboard zone.
- **LO:** Sets the lowest note of the active keyboard zone.
- **Color:** The color selector helps to visually organize patches, for example when using multiple Note Ins to create different keyboard zones.

Outputs

- **Pitch:** Pitch output, includes pitch bend and glide. Range [0, 1].
- **Gate:** Gate output. Carries velocity information by varying gate strength. Range [0, 1].
- **Note:** Note output, does not include pitch bend and glide. Range [0, 1].
- **Pbend:** Pitch bend output. Range [-0.5, 0.5].
- **Vel:** Velocity output. Range [0, 1].
- **Mwheel:** Modulation wheel output. Range [0, 1].
- **Atouch:** Aftertouch output. Range [0, 1].

11.9.11. Pitch CV Out (INT)



The Util Pitch CV Out

The Pitch CV Out allows you to use Blocks for controlling hardware synthesizers with analog pitch and gate CV (Control Voltage) inputs via DC-coupled audio interfaces or D/A converters. An automatic measurement routine calibrates the control voltages sent to the external oscillator, perfectly matching its response to the pitches in Blocks across the full MIDI note range from C-2 to G8.



To learn how to calibrate an external oscillator using the Pitch CV Out, see [Using the Pitch CV Out Block](#).

Controls

- **RANGE:** Switches the range of the **Pitch** output between **FULL** [-1, 1] and **HALF** [-0.5, 0.5]. Use **HALF** to improve the measurement accuracy when using a DC-coupled audio interface or D/A converter with a very wide voltage range, e.g. an Expert Sleepers Eurorack module.
- **GATE:** Switches between V-TRIG and S-TRIG standards for the **Gate** output. Most contemporary synthesizers with voltage control use the V-Trig standard.
- **OK/RUN/!!!:** Starts the calibration by placing the mouse on the control and clicking **RUN**. Aborts the calibration by placing the mouse on the control and clicking **!!!**. If the blue circle surrounding the control is not complete after calibration, the external VCO is tuned too high and does not allow calibration of the entire MIDI note range from C-2 to G8. In this case, tune the VCO down and run the calibration again.
- **MIN:** Displays the lowest successfully calibrated note.
- **MAX:** Displays the highest successfully calibrated note.

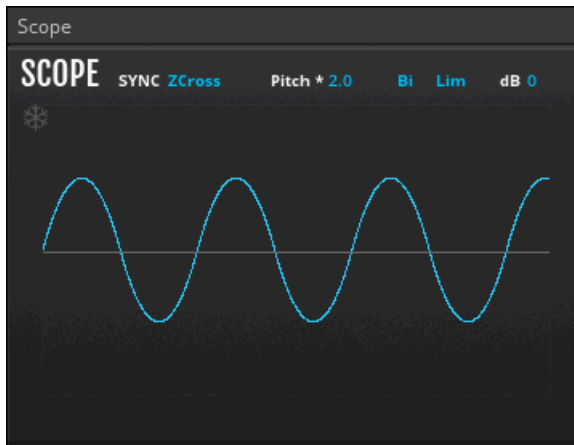
Inputs

- **In:** Measurement input for the signal from your external oscillator. Only used for calibration.
- **Pitch:** Input for the pitch signal you want to use for controlling your hardware synthesizer.
- **Gate:** Input for the gate signal you want to use for controlling your hardware synthesizer.

Outputs

- **Pitch CV:** Output for controlling the pitch of your external oscillator. Needs to be routed through a DC-coupled audio interface or D/A converter.
- **Gate CV:** Output for controlling the envelopes on your hardware synthesizer. Needs to be routed through a DC-coupled audio interface or D/A converter.

11.9.12. Scope (AUX)



The Util Scope

Scope allows you to monitor any signal in a Blocks patch. It shows how signals change over time: The vertical axis represents the amplitude of a signal and the horizontal axis represents time. You can use the Scope to visualize modulation signals or waveforms from oscillators, or to troubleshoot issues in complex patches by checking for the range and shape of a signal. Watching the signals change as you add modulation and processing to your patch is a great learning experience, too. The behavior of the Scope can be adjusted with the controls at the top of the panel.

Controls

- **dB**: Sets the amplitude range (vertical zoom) in dB.
- **Lim**: Turns the limiter on. This way, the waveform cannot exceed the display frame. When hitting the limit, orange lines appear to indicate the value range from -1 to 1 (or 0 to 1 in Unipolar mode).
- **Bi / Uni**: Switches between Bi mode (bipolar, positive and negative values) and Uni mode (unipolar, positive values only).
- **Pitch / mSec**: Switches the setting for the horizontal range between wave cycles (Pitch) and milliseconds (mSec). In mSec mode, the display can be dragged up and down to zoom horizontally.
- **Pitch value**: Sets the horizontal range (time) expressed in wave cycles.
- **mSec value**: Sets the horizontal range (time) expressed in milliseconds.
- **SYNC**: Selects one of six modes for the timing behavior of the display relative to the incoming signal:
 - **Free**: no sync.
 - **ZCross**: Syncs to zero crossings of the input signal.
 - **Slope**: Syncs to peaks of the input signal.
 - **SC ZCross**: Syncs to zero crossings of the signal arriving at the SC input.
 - **SC Slope**: Syncs to peaks of the signal arriving at the SC input.
 - **Pitch in**: Syncs to an internal sync oscillator, the frequency of which is set by the Pitch input.
- **Freeze**: Freezes the display of the Scope. In mSec mode, the display can be dragged up and down to zoom horizontally.

Inputs

- **In**: Scope signal input.
- **SC**: Scope side-chain input.
- **Pitch**: Input for controlling pitch of the internal sync oscillator. An increase of 0.1 equals a 1 octave increase in pitch (ideal for quantized signals from Pitch outputs).

11.9.13. Trig In (INT)



The Util Trig In

The Trig In receives MIDI note data and converts it into gate signals for use in your Blocks patch. Each of its 6 channels is assigned to a MIDI note that triggers a gate to be sent from the associated output whenever it is played. This way you can trigger multiple envelopes and reset sequencers or LFOs all from the same MIDI keyboard or MIDI pad controller in an intuitive manner. The gates carry velocity information, which allows you to play your patches dynamically when used with Blocks such as the [ADSR Envelope \(MOD\)](#) and the [MONARK ADS ENV \(MOD\)](#), which can be configured to respond to gates at varying strength.

Controls

- **CH.:** Selects which MIDI channel the Trigger In receives note data from. Numbers 1-16 correspond to the standard 16 MIDI channels. OMNI sets the Trigger In to receive note data from all 16 channels, while INTERN refers to REAKTOR's internal MIDI bus.
- **MUTE:** Mutes all of the Trigger In's outputs.
- **NOTE:** Transposes all of the current note assignments simultaneously.
- **Note 1 ... 6:** Assigns a MIDI note to the associated Gate output. Only events arriving from this note cause a new gate to be sent, all other note events are ignored.
- **V 1 ... 6:** When enabled, all gates sent from the associated Gate output carry the velocity of the received note events. When disabled, all gate events ignore velocity information, resulting in simple on/off type gate signals.
- **G 1 ... 6:** Toggles between the available output modes for the associated Gate output. When enabled, the gates sent from the Gate output have the same duration as the received note event. When disabled, the gates sent from the Gate output are triggers with a fixed length of 1 ms, independent of the duration of the received note event.

- **MUTE 1 ... 6:** Mutes the associated Gate output.
- **Color 1 ... 6:** The color selector helps to visually organize patches.

Outputs

- **Gate 1:** Channel 1 gate output. Range [0 .. 1]. If V is enabled, it carries velocity information by varying gate strength. Range [0, 1].
- **Gate 2:** Channel 2 gate output. Range [0 .. 1]. If V is enabled, it carries velocity information by varying gate strength. Range [0, 1].
- **Gate 3:** Channel 3 gate output. Range [0 .. 1]. If V is enabled, it carries velocity information by varying gate strength. Range [0, 1].
- **Gate 4:** Channel 4 gate output. Range [0 .. 1]. If V is enabled, it carries velocity information by varying gate strength. Range [0, 1].
- **Gate 5:** Channel 5 gate output. Range [0 .. 1]. If V is enabled, it carries velocity information by varying gate strength. Range [0, 1].
- **Gate 6:** Channel 6 gate output. Range [0 .. 1]. If V is enabled, it carries velocity information by varying gate strength. Range [0, 1].

11.10. West Coast

West Coast Blocks are inspired by Don Buchla's unconventional and playful approach to synthesis.



The West Coast line of Blocks is included in the product [Blocks Primes](#), which is included in REAKTOR 6, but can also be purchased separately for use in the free REAKTOR 6 PLAYER.

11.10.1. CFG (MOD)



The West Coast CFG

Inspired by the so-called function generators found in classic Buchla synthesizers, the West Coast QFG combines four envelopes in interesting ways. It can be used as the central modulation hub of a patch, providing a set of modulation signals that can be triggered independently, in relation to each other, or by themselves for self-cycling operation. The QFG is suitable for creating sounds ranging from evolving drones to complex percussion.

Controls

- **SPEED:** Multiplies the RISE and FALL times for all four envelopes at the same time.
- **Configuration:** Selects between three different configurations for the four envelopes.
 - **4x1:** All envelopes are independent. Each envelope is triggered by a gate received at the corresponding Gate input (One Shot, Hold modes), or cycles on its own (Cycle mode). In Cycle mode, a trigger received during the fall stage restarts the cycle from its current value.
 - **2x2:** Envelopes 1/2 as well as envelopes 3/4 are combined in pairs and triggered in a serial fashion. The second envelope of a pair is either triggered when the first envelope has completed its RISE stage (Quadrature mode), or when the first envelope has completed its FALL stage (End mode). The first envelope of a pair is triggered by a gate received at the first envelope's Gate input. The envelopes are not triggered as long as their respective RISE stages are active.
 - **1x4:** All envelopes are combined and triggered in a serial fashion. Each envelope is either triggered when the previous envelope has completed its RISE stage (Quadrature mode), or when the previous envelope has completed its FALL stage (End mode). The first envelope is triggered by a gate received at the Gate 1 input. The envelopes are not triggered as long as their respective RISE stages are active.
- **Gate 1 ... 4:** Is highlighted when a gate is received at the Gate input, and can be used to manually trigger the envelope.
- **4x1 Mode 1 ... 4:** Selects between three different behaviors for the envelope in the 4x1 configuration:
 - **One Shot:** A positive zero crossing at the input triggers an AD (Attack, Decay) envelope, with the attack being set with the RISE parameter and the release being set with the FALL parameter. Gates received during the RISE stage are ignored.
 - **Hold:** A positive zero crossing at the input triggers an ASR (Attack, Sustain, Release) envelope, with the attack being set with the RISE parameter and the release being set with the FALL parameter. The sustain stage is held for as long as the input signal remains positive after the zero crossing is detected.
 - **Cycle:** The envelope is cycling on its own. Unlike an LFO, the output signal consists of positive values only, creating a unipolar waveform. The shape of the waveform depends on the Rise and Fall settings. If Rise and Fall have the same setting, the QFG outputs a triangle wave. If Rise is set to 0 and Fall is turned CW, it outputs a saw wave. If Fall is set to 0 and Rise is turned CW, it outputs a ramp wave. A positive zero crossing received at the input during the fall stage restarts the envelope from its current value.
- **2x2 Mode:** Selects between two different modes for the 2x2 configuration. In Quadrature mode, the second envelope of the pair is triggered when the first envelope has completed its RISE stage. In End mode, the second envelope of the pair is triggered when the first envelope has completed its FALL stage.
- **2x2 Cycle:** Enables Cycle mode for the 2x2 configuration. When enabled, the first envelope of the pair triggers the second envelope in as specified by the 2x2 Mode 3/4 parameter. In the

same way, the second envelope of the pair triggers the first envelope, creating a cycling function composed of both envelopes.

- **1x4 Mode:** Selects between two different modes for the 1x4 configuration. In Quadrature mode, the each envelope is triggered when the previous envelope has completed its RISE stage. In End mode, each envelope is triggered when the previous envelope has completed its FALL stage.
- **1x4 Cycle:** Enables Cycle mode for the 1x4 configuration. When enabled, the four envelopes trigger each other in series as specified by the 1x4 Mode parameter. In the same way, the final envelope triggers the first envelope, creating a cycling function composed of all four envelopes.
- **SHAPE Rise 1 ...4:** Fades between linear, exponential, or logarithmic shapes for the RISE stage.
- **TIME Rise 1 ...4:** Sets the attack time of the envelope.
- **SHAPE Fall 1 ...4:** Fades between linear, exponential, or logarithmic shapes for the FALL stage.
- **TIME Fall 1 ...4:** Sets the decay or release time of the envelope.
- **MIX / MAX:** Selects how the four envelopes are combined for the Out 5 output. MIX outputs the sum of the envelope signals. MAX outputs the current maximum value from any of the four envelopes.
- **LEVEL 1 ...4:** Adjusts the level at which the envelope is sent to the Out 5 output.

Inputs

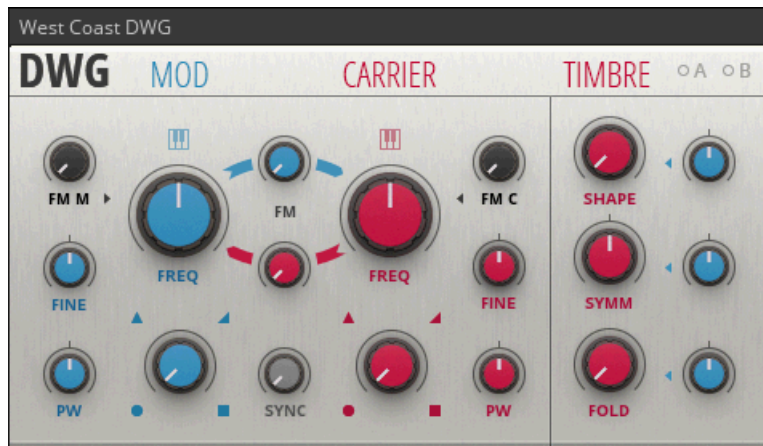
- **Gate 1:** A positive zero crossing at this input triggers envelope 1 in the 4x1 configuration, both envelopes 1 and 2 in the 2x2 configuration, and all four envelopes in 1x4 configuration.
- **Gate 2:** A positive zero crossing at this input triggers envelope 2 in the 4x1 configuration.
- **Gate 3:** A positive zero crossing at this input triggers envelope 3 in the 4x1 configuration, and both envelopes 3 and 4 in the 2x2 configuration.
- **Gate 4:** A positive zero crossing at this input triggers envelope 4 in the 4x1 configuration.
- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out 1:** Envelope 1 signal output. Range [0, 1].
- **Out 2:** Envelope 2 signal output. Range [0, 1].
- **Out 3:** Envelope 3 signal output. Range [0, 1].
- **Out 4:** Envelope 4 signal output. Range [0, 1].
- **End 1:** Outputs a trigger pulse when envelope 1 completes its cycle. Range [0 .. 1].
- **End 2:** Outputs a trigger pulse when envelope 2 completes its cycle. Range [0 .. 1].
- **End 3:** Outputs a trigger pulse when envelope 3 completes its cycle. Range [0 .. 1].
- **End 4:** Outputs a trigger pulse when envelope 4 completes its cycle. Range [0 .. 1].
- **Out 1+2:** Combined signal output of envelopes 1 and 2 as set with the MIX / MAX and LEVEL 1-4 controls.
- **Out 3+4:** Combined signal output of envelopes 3 and 4 as set with the MIX / MAX and LEVEL 1-4 controls.

- **Out All:** Combined signal output of all envelopes as set with the MIX / MAX and LEVEL 1-4 controls.

11.10.2. DWG (OSC)



The West Coast DWG

The West Coast DWG builds on a long history of dual oscillator designs pioneered by synthesizer legend Don Buchla. It combines two oscillators (**MOD**, **CARRIER**) and additional wave shaping (**TIMBRE**) with extensive internal modulation possibilities, allowing you to intuitively explore complex timbres by means of frequency modulation synthesis (**FM**), blending of waveforms (**SHAPE**) as well as wave folding (**FOLD**, **SYMM**). Together with the West Coast LPG it forms the core voice of any synthesizer in the Buchla tradition.

Controls

- **FM MOD:** Controls the amplitude of the signal arriving at the FM M input, also known as the FM index. When the knob is at its lowest setting, the FM input signal is fully attenuated and the oscillator's frequency is not being modulated. As the knob is turned clockwise, the amplitude of the signal present at the FM M input increases, applying a greater amount of frequency modulation to the MOD oscillator.
- **FREQ MOD:** Adjusts the base frequency of the MOD oscillator.
- **COARSE MOD:** Coarse tuning control in semitones.
- **RATIO MOD:** Adjusts the base frequency of the MOD oscillator as a ratio of the CARRIER oscillator's frequency.
- **FINE MOD:** Fine tuning control in cents. When Ratio tuning mode is enabled, this knob becomes a LAG control, adjusting how quickly the MOD oscillator responds to changes in the CARRIER oscillator's frequency.
- **LAG:** Only available in Ratio tuning mode. Adjusts how quickly the MOD oscillator responds to changes in the CARRIER oscillator's frequency. In all other tuning modes, this knob acts as a fine tuning control in cents.
- **MOD Tuning:** Selects one of three tuning modes for the MOD oscillator:
 - **Free:** The oscillator runs at a fixed rate that can be adjusted with the FREQ MOD and FINE MOD controls.

- **Key Tracking:** The frequency of the oscillator is controlled from the Pitch M input. Additional tuning in semitones and cents is available via the COARSE MOD and FINE MOD controls.
- **Ratio:** The oscillator runs at a defined ratio of the CARRIER oscillator's frequency. The ratio can be adjusted with the RATIO control, and the response time can be adjusted with the LAG control.
- **Internal FM MOD:** Controls the amplitude of the signal arriving internally from the CARRIER oscillator, also known as the FM index. When the knob is at its lowest setting, the internal FM signal is fully attenuated and the oscillator's frequency is not being modulated. As the knob is turned clockwise, the amplitude of the FM signal increases, applying a greater amount of frequency modulation to the MOD oscillator.
- **PW MOD:** Adjusts the width of the pulse waveform.
- **Wave MOD:** Morphs smoothly between the four classic waveforms: sine, triangle, saw and pulse. The result is sent from the Out M output, as well as being used for the internal modulation buses.
- **FM CARRIER:** Controls the amplitude of the signal arriving at the FM C input, also known as the FM index. When the knob is at its lowest setting, the FM input signal is fully attenuated and the oscillator's frequency is not being modulated. As the knob is turned clockwise, the amplitude of the signal present at the FM C input increases, applying a greater amount of frequency modulation to the CARRIER oscillator.
- **FREQ CARRIER:** Adjusts the base frequency of the CARRIER oscillator
- **COARSE CARRIER:** Coarse tuning control in semitones.
- **FINE CARRIER:** Fine tuning control in cents.
- **Key Tracking:** When key tracking is enabled, the frequency of the oscillator is controlled from the Pitch C input. Additional tuning in semitones and cents is available via the COARSE B and FINE B controls. When key tracking is disabled, the oscillator runs at a fixed rate that can be adjusted with the FREQ B and FINE B controls.
- **Internal FM CARRIER:** Controls the amplitude of the signal arriving internally from the MOD oscillator, also known as the FM index. When the knob is at its lowest setting, the internal FM signal is fully attenuated and the oscillator's frequency is not being modulated. As the knob is turned clockwise, the amplitude of the FM signal increases, applying a greater amount of frequency modulation to the CARRIER oscillator.
- **PW CARRIER:** Adjusts the width of the pulse waveform.
- **Wave CARRIER:** Morphs smoothly between the four classic waveforms: sine, triangle, saw and pulse. The result is sent from the Out C output, as well as being used for the internal modulation bus. Additionally, this signal is passed on to the TIMBRE section for additional wave-shaping.
- **SYNC:** Adjusts the strength at which the MOD oscillator is synced to the CARRIER oscillator.
- **SHAPE:** Morphs smoothly between the wave selected by the Wave CARRIER knob (SHAPE set fully counter-clockwise) and two additional waveforms, a spike wave (SHAPE set at center position) and a triangle wave with a spike (SHAPE set fully clockwise). The result is passed through the wavfolder (SYMM and FOLD parameters) and sent from the Out T output.
- **SHAPE Modulation Amount:** Controls the amplitude and polarity of the signal arriving internally from the MOD oscillator. When the knob is in the center position, the signal is fully attenuated and the SHAPE parameter is not being modulated by the MOD oscillator. Turning the control clockwise increases the amplitude of the signal in a positive fashion. Turning the control

counter-clockwise increases the amplitude of the signal in a negative fashion, effectively inverting its polarity. Increasing the amplitude of the signal results in a greater amount of modulation being applied to the SHAPE parameter.

- **SYMM**: Applies an offset to the wavefolder's input signal, introducing asymmetry into the waveform and altering the balance between odd and even harmonics.
- **SYMM Modulation Amount**: Controls the amplitude and polarity of the signal arriving internally from the MOD oscillator. When the knob is in the center position, the signal is fully attenuated and the SYMM parameter is not being modulated by the MOD oscillator. Turning the control clockwise increases the amplitude of the signal in a positive fashion. Turning the control counter-clockwise increases the amplitude of the signal in a negative fashion, effectively inverting its polarity. Increasing the amplitude of the signal results in a greater amount of modulation being applied to the SYMM parameter.
- **FOLD**: Increases the amplitude of the wavefolder's input signal. As the amplitude increases, the wave is repeatedly folded back into itself, resulting in an greater degree of harmonic content and complexity.
- **TIMBRE Modulation Amount**: Controls the amplitude and polarity of the signal arriving internally from the MOD oscillator. When the knob is in the center position, the signal is fully attenuated and the FOLD parameter is not being modulated by the MOD oscillator. Turning the control clockwise increases the amplitude of the signal in a positive fashion. Turning the control counter-clockwise increases the amplitude of the signal in a negative fashion, effectively inverting its polarity. Increasing the amplitude of the signal results in a greater amount of modulation being applied to the FOLD parameter.

Inputs

- **Pitch M**: Input for controlling the pitch of the MOD oscillator. An increase of 0.1 equals a 1 octave increase in pitch (ideal for quantized signals from Pitch outputs).
- **Pitch C**: Input for controlling the pitch of the CARRIER oscillator. An increase of 0.1 equals a 1 octave increase in pitch (ideal for quantized signals from Pitch outputs).
- **FM M**: Linear modulation input for the MOD oscillator's frequency, amount is set with the FM MOD control.
- **FM C**: Linear modulation input for the CARRIER oscillator's frequency, amount is set with the FM CARRIER control.
- **Mod A**: Modulation bus **A** input.
- **Mod B**: Modulation bus **B** input.

Outputs

- **Out M**: MOD oscillator signal output.
- **Out C**: CARRIER oscillator signal output.
- **Out T**: TIMBRE section signal output.

11.10.3. LPG (FLT)



The West Coast LPG

The West Coast LPG represents the Blocks take on the low pass gate, a peculiar filter circuit originally designed by synthesizer legend Don Buchla for controlling the dynamic properties of a sound. It uses an optocoupler that smoothens the LPG's response to external control. When excited by a sharp-edged signal via its **Pluck** input (see section [Pluck](#)), the LPG opens and closes in a natural sounding manner, giving sounds a plausible quality that is reminiscent of a drum hit or a plucked string.

Controls

- **VANILLA / SNAPPY / SMOOTH**: Selects between different models for the optocoupler used in the LPG. Each model responds differently to gates received at the Pluck input as well as modulation signals applied to the LEVEL control.
 - **VANILLA**: Response with a punchy attack and a medium decay.
 - **SNAPPY**: Response with a sharp attack and a short decay.
 - **SMOOTH**: Response with a soft attack and a long decay.
- **LP**: Enables LP mode, which is slightly resonant and has a steeper frequency roll-off for high frequencies.
- **LEVEL**: Sets the amount of gating applied to signals passing through the Lowpass Gate. As LEVEL decreases, high frequency content is reduced and the amplitude of the signal is reduced accordingly, gating the signal more and more. At the lowest settings, the input signal is fully gated.
- **PLUCK**: Attenuates the gates received at the Pluck input for exiting the LPG. The amount of attenuation applied to a gate drastically changes LPG's response to the excitation.
- **DAMP**: Changes the LPG's sound and its response to modulation signals, mimicking a dampening effect on a drum or string instrument.

Inputs

- **In**: LPG signal input.
- **Pluck**: Input for exciting the LPG so it opens and closes in a natural sounding manner, reminiscent of a drum hit or a plucked string. The Pluck input responds well to sharp-edged signals,

like gates, pulse waveforms, and stepped sequences. The signal arriving at the Pluck input can be attenuated with the PLUCK control, drastically changing the LPG's response to excitation.

- **Mod A:** Modulation bus **A** input.
- **Mod B:** Modulation bus **B** input.

Outputs

- **Out:** LPG signal output.

11.10.4. XYS (SEQ)



The West Coast XYS

Drawing inspiration from an innovative modular sequencer concept born in the Blue Ridge Mountains, the West Coast XYS can be used for generating complex modulation and gate signals as well as note sequences. It is based on a two-dimensional grid of 16 steps that can be accessed by a combination of two clock and two modulation signals, controlling both the direction and the tempo of the sequence. Even though it can be used as a basic 16 step sequencer, the XYS excels at creating evolving patterns and constantly shifting sequences.

Controls

- **RESET:** Clicking this button resets the sequencer to the first step. Additionally, the button indicates when the internal counter is reset when receiving a positive zero crossing at the Reset input.
- **XG:** Selects the XG page for the sequencer grid, showing the X gate toggle buttons for all steps. When the sequencer advances in X-direction to a step with an active X gate, a gate is sent to the Gate X output and the LOGIC section.
- **YG:** Selects the YG page for the sequencer grid, showing the Y gate toggle buttons for all steps. When the sequencer advances in Y-direction to a step with an active Y gate, a gate is sent to the Gate Y output and the LOGIC section.
- **VAL:** Selects the VAL page for the sequencer grid, showing the Value knobs for all steps. When the sequencer advances to a step, the corresponding value is sent out from the Out and Out Logic outputs. The Out Logic output values are latched to the gates sent from the Gate Logic output.

- **Mode:** Selects between different modes for advancing the sequencer via the Gate inputs:
 - **Zigzag 1:** Gates received at the Gate X input advance the sequencer in a horizontal zigzag pattern from step A1 to step 4D. Gates received at the Gate Y input are ignored.
 - **Zigzag 2:** Gates received at the Gate X input advance the sequencer in a vertical zigzag pattern from step A1 to step 4D. Gates received at the Gate Y input are ignored.
 - **XY:** Gates received at the Gate X input advance the sequencer in X-direction, and gates received at the Gate Y input advance the sequencer in the Y-direction.
- **GATE X:** Lights up when a gate is received via the Gate X input, and can be used to manually advance the sequencer one step in the X-direction.
- **GATE Y:** Lights up when a gate is received via the Gate Y input, and can be used to manually advance the sequencer one step in the Y-direction.
- **DIR X:** Allows you to select between five different playback modes for the X-direction:
 - **Forwards:** The sequencer advances from left to right.
 - **Backwards:** The sequencer advances from right to left.
 - **Pendulum:** The sequencer advances forwards until it reaches the final step, at which point it immediately changes direction, advancing backwards until the first step is reached, and so on.
 - **Ping-Pong:** The sequencer advances in the same way as in Pendulum mode, however the first and the last step are repeated when changing direction.
 - **Random:** The sequencer advances in a random fashion. Every time a gate is received, the sequencer randomly chooses one of the active steps to advance to.
- **DIR Y:** Allows you to select between five different playback modes for the Y-direction:
 - **Forwards:** The sequencer advances from left to right.
 - **Backwards:** The sequencer advances from right to left.
 - **Pendulum:** The sequencer advances forwards until it reaches the final step, at which point it immediately changes direction, advancing backwards until the first step is reached, and so on.
 - **Ping-Pong:** The sequencer advances in the same way as in Pendulum mode, however the first and the last step are repeated when changing direction.
 - **Random:** The sequencer advances in a random fashion. Every time a gate is received, the sequencer randomly chooses one of the active steps to advance to.
- **OFFSET X:** Allows you to shift the position in the X-direction, including the first step (reset position).
- **OFFSET Y:** Allows you to shift the position in the Y-direction, including the first step (reset position).
- **Value 1A ... 4D:** Sets the value for this step. When Quantize is enabled, the value is set in semitones.
- **X Gate 1A ... 4D:** Toggles the X gate for this step on or off.
- **Y Gate 1A ... 4D:** Toggles the Y gate for this step on or off.
- **LOGIC:** Selects between different output modes for the Gate Logic output. Boolean logic operations are used to combine the X and Y gates in interesting ways. The following modes are available:
 - **X:** Outputs X gates only.
 - **Y:** Outputs Y gates only.

- **OR**: Outputs both X gates and Y gates.
- **XOR**: Outputs a gate when either an X gate or a Y gate is active, but does not output a gate when both X and Y gates are active.
- **AND**: Outputs a gate when both X and Y gates are active.
- **Quantize**: Enables quantization for the Value knobs. In this mode, a pitch signal is sent out from the Out and Out Logic outputs, quantized to semitones.

Inputs

- **Gate X**: A positive zero crossing at this input causes the sequencer to advance on step in X-direction.
- **Gate Y**: A positive zero crossing at this input causes the sequencer to advance on step in Y-direction.
- **Reset**: A positive zero crossing at this input causes the sequencer to reset.
- **Mod A**: Modulation bus **A** input.
- **Mod B**: Modulation bus **B** input.

Outputs

- **Out**: Sequencer value output. Range [0, 1].
- **Gate X**: X gate output. Sends a gate when the sequencer advances in X-direction to a step with an active X gate. Range [0 .. 1].
- **Gate Y**: Y gate output. Sends a gate when the sequencer advances in Y-direction to a step with an active Y gate. Range [0 .. 1].
- **Out Logic**: Sequencer value output, latched to the gates sent from the Gate Logic output. Range [0, 1].
- **Gate Logic**: Logic gate output. Sends a combination of X and Y gates according to the selected LOGIC output mode. Range [0 .. 1].

12. USING BLOCKS IN ENSEMBLES

When Blocks were introduced with REAKTOR 6.0, Racks did not exist. Since Blocks are REAKTOR **Instruments** (REAKTOR *.ism* file format), they can be combined in REAKTOR **Ensembles** (REAKTOR *.ens* format) to create patches. In an Ensemble, connections are made in the **Structure**, while the parameters are controlled on the **Panel**.

Since the introduction of Racks in REAKTOR 6.3, Ensembles can be considered a legacy format in the context of Blocks. However, they still serve a special purpose by bridging patching with Blocks and building User Blocks. Contrary to Racks, Ensembles provide access to the the internal structure of a Block. Therefore, it is possible to modify existing Blocks, or combine patching and building when creating a new User Block.



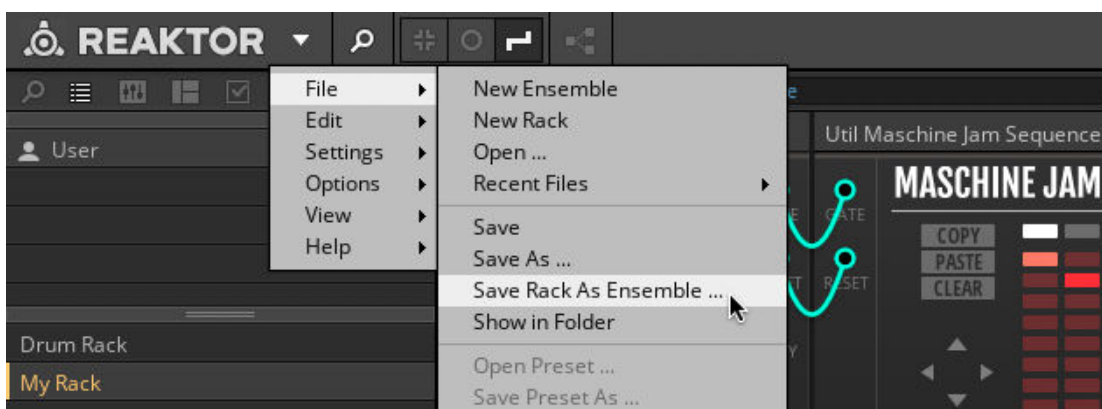
For more information about using and creating User Blocks, see [User Blocks](#).

Racks can be [saved as an Ensemble](#), allowing you to open and modify any of the contained Blocks. You can also start a patch in the [New Ensemble](#) and use it as a starting point to modify or build Blocks. Additionally, the [factory Ensembles](#) from the original Blocks products can still be loaded in REAKTOR. The following sections also explain how loading, saving, and other basic workflows differ from using Racks.

12.1. Saving a Rack as Ensemble

Saving a Rack as Ensemble allows you to dive into the internal structure of each included Block or refine your own Blocks in the context of a patch.

1. Click on the drop-down menu next to the REAKTOR logo and select **Save Rack As Ensemble ...** in the **File** submenu.



2. In the file dialog, browse to the location on your hard drive you want to save the Ensemble in.
3. Type in a name for your Ensemble.
4. Click **Save** to save the Ensemble.

The Ensemble is saved in the chosen location on your hard drive.

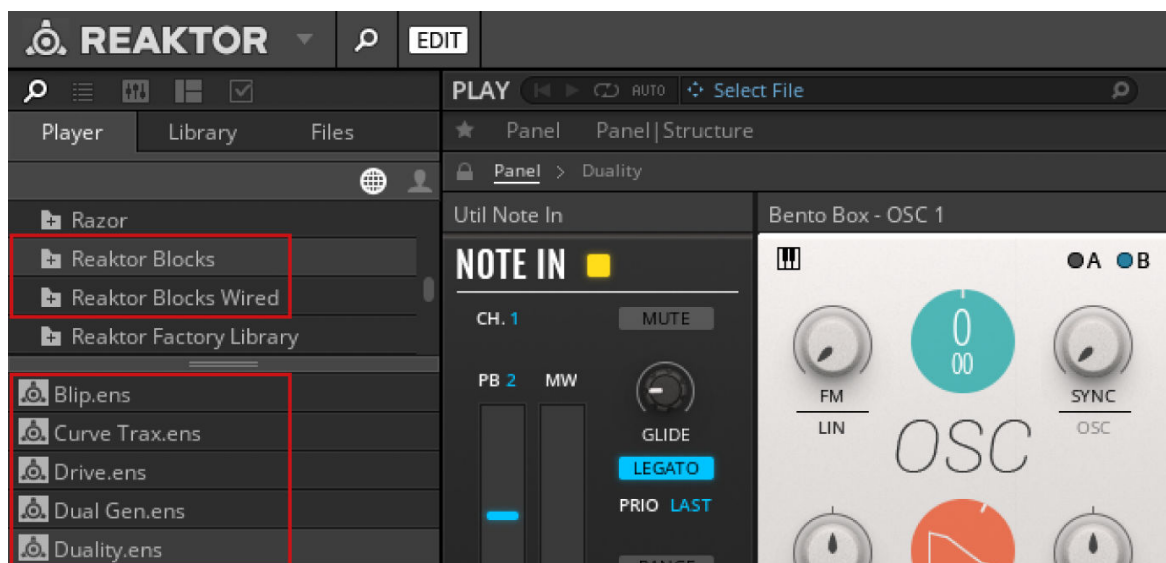


After saving a Rack as Ensemble, the Rack remains opened in REAKTOR. To use the Ensemble instead, you have to open it manually.

12.2. Factory Ensembles

The pre-built Ensembles from the original Blocks products are still available in REAKTOR if **REAKTOR Blocks** and **REAKTOR Blocks Wired** are installed using Native Access. They offer rich patches with many refined Snapshots and can serve as an inspiration when looking for new patching ideas.

You can access them in the Player tab of the REAKTOR Browser by selecting the **Reaktor Blocks** and **Reaktor Blocks Wired** entries:

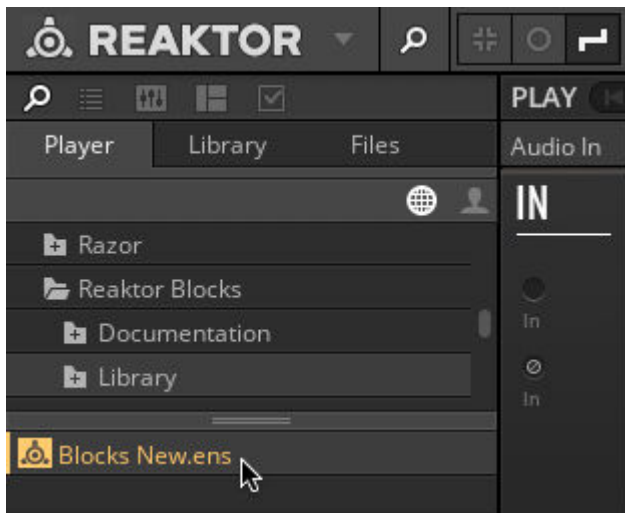


The Factory Ensembles are also available in the MASCHINE and KOMPLETE KONTROL Browser.

12.3. The New Ensemble

Ensembles are the basic project files in REAKTOR that hold all relevant information of a session. Ensembles used for Blocks are based on a dedicated New Ensemble, containing a number of modules that provide the infrastructure for your patch. More importantly, the New Ensemble includes optimizations for the automation handling and Panel view that are required for Blocks.

1. Go to the **Player** tab in the REAKTOR Browser.
2. Open the **Reaktor Blocks** folder under **Native Instruments**.
3. Double-click *Blocks New.ens* in the lower section of the Browser.

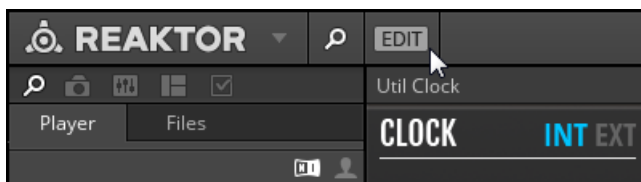


The New Ensemble opens, ready for patching with Blocks.

12.4. Play Mode vs. Edit Mode

You can play your patch and control its parameters in REAKTOR's Play mode, however if you want to add Blocks to your patch, change its Structure, or edit some of its advanced settings, you have to enable Edit mode.

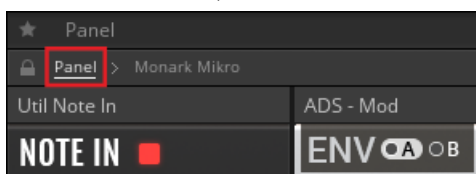
- To toggle Edit mode on or off, click on the **EDIT** button in the REAKTOR header.



12.5. Navigating between Panel and Structure

You can switch between the Panel and the Structure using so-called Breadcrumbs in the navigation bar at the top of the main area.

- To view the Panel, click on the **Panel** Breadcrumb in the navigation bar.

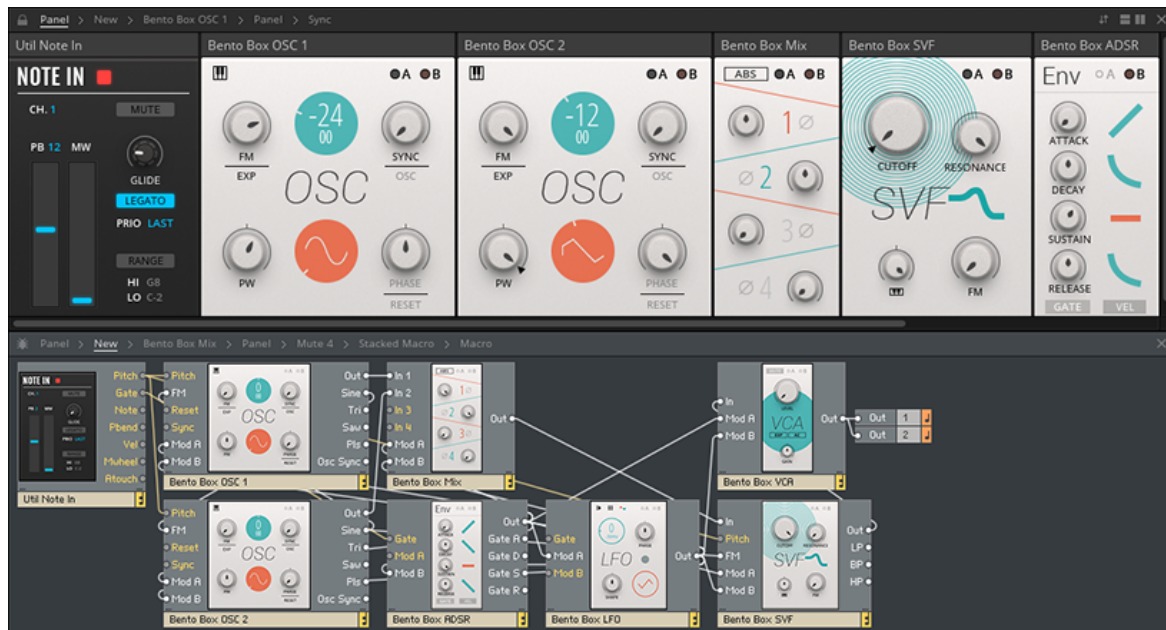


- To view the Structure, click on the Breadcrumb carrying the name of the Ensemble in the navigation bar.



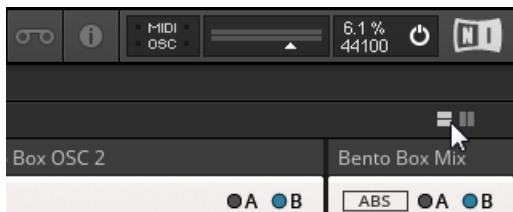
Split View

You can also use REAKTOR's split view to view both the Panel and the Structure at the same time:



A patch in REAKTOR Blocks, viewed in split view

- To enable split view, click on the Horizontal Split or Vertical Split buttons in the upper-right corner of the main area.



12.6. Saving Patches and Parameter Settings

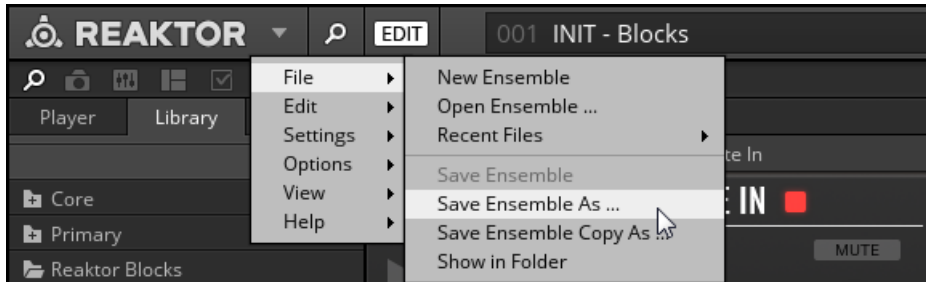
All connections between Blocks and their arrangement in the Structure are saved in the Ensemble, so you have to save a new Ensemble for each of your patches. See sections [Saving and Loading Ensembles in the Stand-Alone Application](#) and [Saving and Loading Ensembles in the Plug-in](#) for more information.

The parameter settings on the Panel can be saved in presets, called **Snapshots** in REAKTOR. You can create global Snapshots for all Blocks in a patch at once, on the Ensemble level, or individual Snapshots for each Block, on the Instrument level. This allows you to save several sets of settings for a patch and recall them in an instant. See section [Storing and Recalling Snapshots](#) for more information.

12.6.1. Saving and Loading Ensembles in the Stand-Alone Application

In the REAKTOR stand-alone application, you can save and load Ensembles from the main menu in the header.

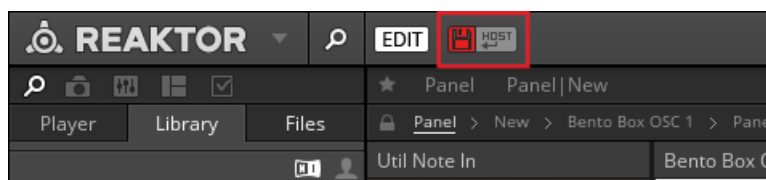
- To open the main menu, click the arrow next to the **REAKTOR** logo.



Alternatively, you can use the **Files** tab in the REAKTOR Browser to browse for your Ensembles and load them by drag and drop, or by double-clicking.

12.6.2. Saving and Loading Ensembles in the Plug-in

When you are using the REAKTOR plug-in in Edit mode, changes made in the Ensemble are not saved in the DAW project. In this case, the Save Ensemble button in the header turns red, indicating that changes will be lost when closing the project:

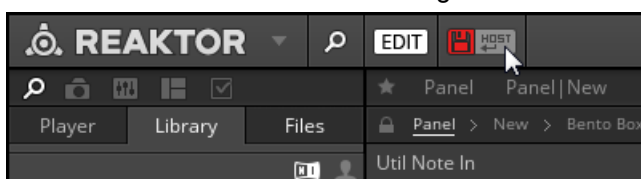


The red Save Ensemble button

To preserve the changes you have made, you need to create a local copy of the Ensemble. A link to this local copy is saved in the DAW project and the Ensemble will be automatically loaded the next time you open the project. REAKTOR allows for automatic saving of this local copy every time you save the DAW project. Therefore, you do not have to keep track of the Ensemble's save status as long as your DAW project is saved.

To save a local copy of your Ensemble and enable automatic saving:

1. Click on the Enable Automatic Saving with Host button in the header



- In the file dialog, choose the desired location and name for the local copy of your Ensemble and click **Save**.

The Enable Automatic Saving with Host button lights up, indicating that automatic saving of your Ensemble is now enabled.

12.6.3. Storing and Recalling Snapshots

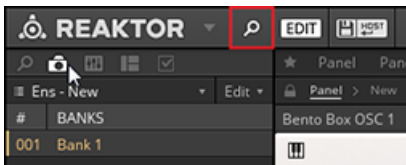
Snapshots, which are REAKTOR's sound preset format for Ensembles, enable you to store and recall the state of your Instrument's or Ensemble's panel controls. You can use this to store the parameter settings of entire Blocks patches (Ensemble) or individual Blocks (Instruments).



The connections between Blocks in the Structure are not stored in Snapshots. They are saved in the Ensemble, which means that changes made to the Structure of a patch will affect all Ensemble Snapshots.

You can view and edit Snapshots in the Snapshots tab of the side pane.

- To access the Snapshots tab, click on the Snapshots icon in the side pane.

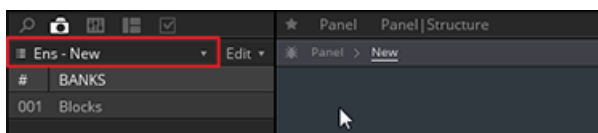


Recalling Snapshots

The Snapshots tab in the side pane shows either the Snapshots for the Ensemble or for an individual Block.

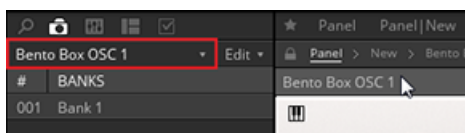
- To view the Snapshots for the Ensemble, including all settings of the included Blocks, click anywhere in the background of the Structure.

The side pane shows all Snapshots saved for the Ensemble, indicated by the menu at the top of the Snapshots tab.



- To view the Snapshots for an individual Block, click on the header of the Block to select it.

The side pane shows all Snapshots stored for the Block, indicated by the menu at the top of the Snapshots tab.



Alternatively, you can select between viewing the Snapshots for the Ensemble or any of the included Blocks by using the Select Instrument for Snapshot menu at the top of the Snapshots tab:



You can also recall Ensemble Snapshots from the Snapshot menu in the header. This will recall the stored parameter settings for all Blocks included in the Ensemble.



Storing Snapshots

Storing and editing Snapshots can be done from the Snapshots tab in the side pane.



In order to be able to store and edit Snapshots, you have to enable Edit mode by clicking on the EDIT button in the header.

At the bottom of the Snapshot pane are three buttons for storing Snapshots:



- **Add:** Creates a new Snapshot with the current settings of the Ensemble or Instrument at the end of the Snapshot list.
- **Store:** Overwrites the currently highlighted Snapshot with the current settings of the Ensemble or Instrument.

- **Insert:** Creates a new Snapshot with the current settings of your Ensemble or Instrument directly after the currently highlighted Snapshot, moving the other Snapshots down one slot.



Snapshots are stored in the Ensemble, so the Ensemble needs to be saved in order to make changes to the Snapshots permanent.

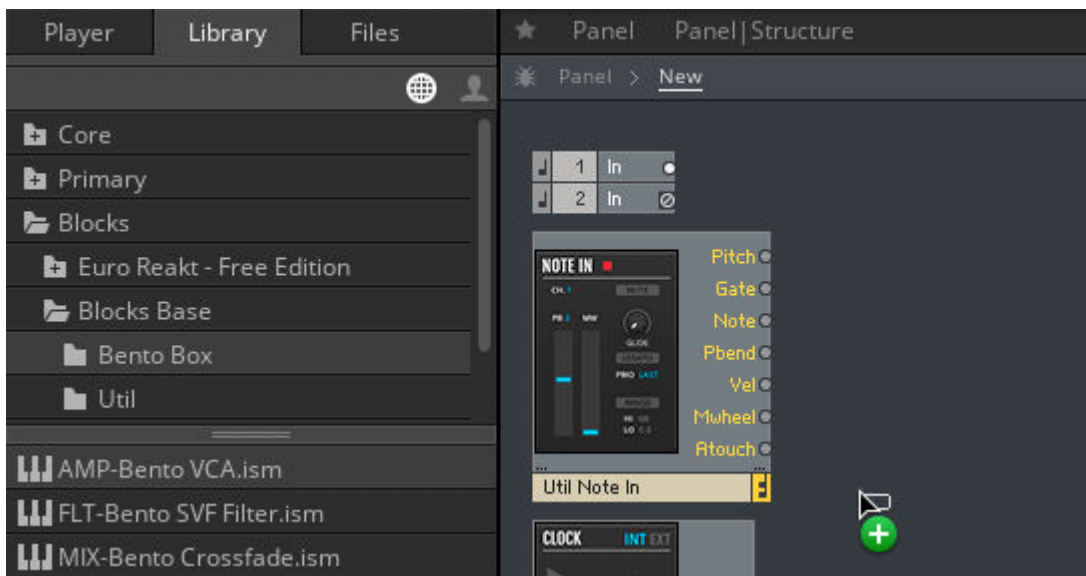
12.7. Adding Blocks to an Ensemble

Blocks can be added to an Ensemble by using drag and drop from the Library tab in the Browser.



The Library tab is only visible in Edit Mode. For more information, see section [Play Mode vs. Edit Mode](#).

1. Go to the **Library** tab in the REAKTOR Browser.
2. Open any of the subfolders under **Blocks**.
3. Drag and drop the Block from the lower section of the **Library** tab into the Structure.



12.8. Organizing and Arranging Blocks in an Ensemble

Using Blocks in Ensembles adopts the fundamental REAKTOR interface paradigm, which is based on a Structure and a Panel view. Connections are made in the Structure, while the parameters are controlled on the Panel.

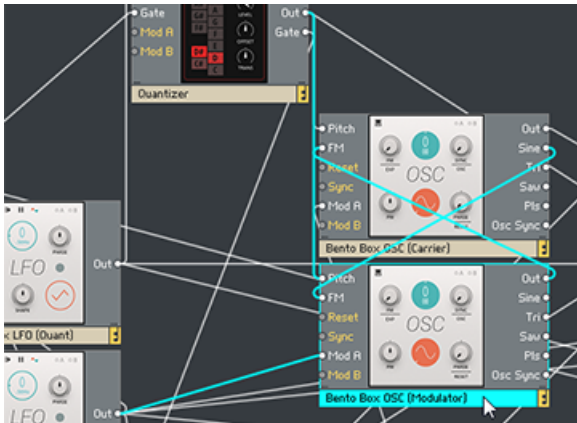
The arrangements of the Blocks in both views are independent. This means that you can organize the Blocks in Structure view to achieve the best possible overview of your signal flow, while setting up a different arrangement in Panel view that suits your way of playing with the parameters.

Arranging Blocks in the Structure

In the Structure, Blocks can be freely arranged, supported by thumbnail pictures and REAKTOR's flexible wiring system. Blocks can be placed on top of each other, however it is recommended to give each Block its own dedicated space in the Structure to maintain a better overview.

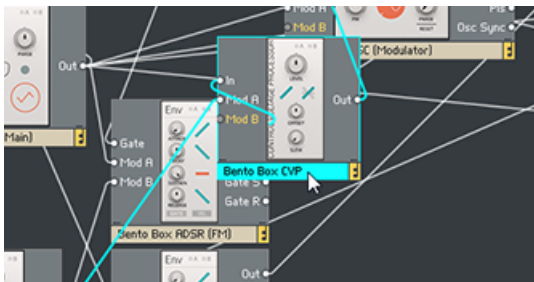
The wires connecting the Blocks always follow the shortest possible path between the ports. If a wire crosses a Block, the overlapping portion of the wire is hidden behind the Block.

- To highlight all wires connected to a Block, including those being hidden behind other Blocks, select it.

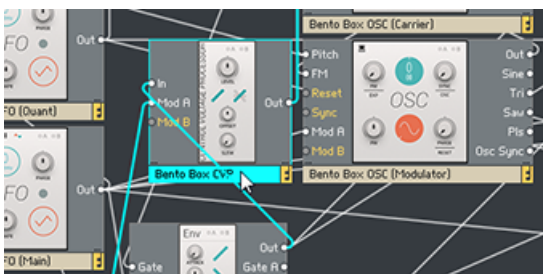


Moving a Block to a new position in the Structure is done by using drag and drop. While moving a Block, all connected wires are rerouted automatically to fit the new arrangement. To move a Block to a new position in the Structure view:

- Drag and drop the Block to a new position.



The Block is placed in its new position.



Arranging Blocks on the Panel

On the Panel, the Blocks are organized in a virtual rack that consists of as many rows as you need. All rows of Blocks have the same, fixed height. The width of the rows is determined by the number of Blocks you add to them. Blocks can be conveniently rearranged by using drag and drop:

1. To place a Block to the left of another Block, click on its header and drag it onto the left half of the other Block.



2. To place a Block to the right of another Block, click on its header and drag it onto the right half of the other Block.



The only exception is if you want to place a Block in the empty area on the right side of a row.

- In this case, click on the Block's header and drag it onto the empty area.



REAKTOR's Panelsets allow you to save multiple Panel configurations for a patch and recall them at an instant. For more information, refer to the REAKTOR 6 Diving Deeper document.