

REAKTOR 6

WHAT IS NEW

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, Mac OS, GarageBand, Logic, iTunes and iPod are registered trademarks of Apple Inc., registered in the U.S. and other countries.
Windows, Windows Vista and DirectSound are registered trademarks of Microsoft Corporation in the United States and/or other countries.
All other trade marks are the property of their respective owners and use of them does not imply any affiliation with or endorsement by them.

Document authored by: Adam Hanley, Jan Ola Korte

Software version: 6.3 (04/2019)

NATIVE INSTRUMENTS GmbH
Schlesische Str. 29-30
D-10997 Berlin
Germany
www.native-instruments.de

NATIVE INSTRUMENTS North America, Inc.
6725 Sunset Boulevard
5th Floor
Los Angeles, CA 90028
USA
www.native-instruments.com

NATIVE INSTRUMENTS K.K.
YO Building 3F
Jingumae 6-7-15, Shibuya-ku,
Tokyo 150-0001
Japan
www.native-instruments.co.jp

NATIVE INSTRUMENTS UK Limited
18 Phipp Street
London EC2A 4NU
UK
www.native-instruments.co.uk

NATIVE INSTRUMENTS FRANCE SARL
113 Rue Saint-Maur
75011 Paris
France

www.native-instruments.com

SHENZHEN NATIVE INSTRUMENTS COMPANY Limited
5F, Shenzhen Zimao Center

111 Taizi Road, Nanshan District, Shenzhen, Guangdong

China

www.native-instruments.com

NATIVE INSTRUMENTS

© NATIVE INSTRUMENTS GmbH, 2019. All rights reserved.

Table of Contents

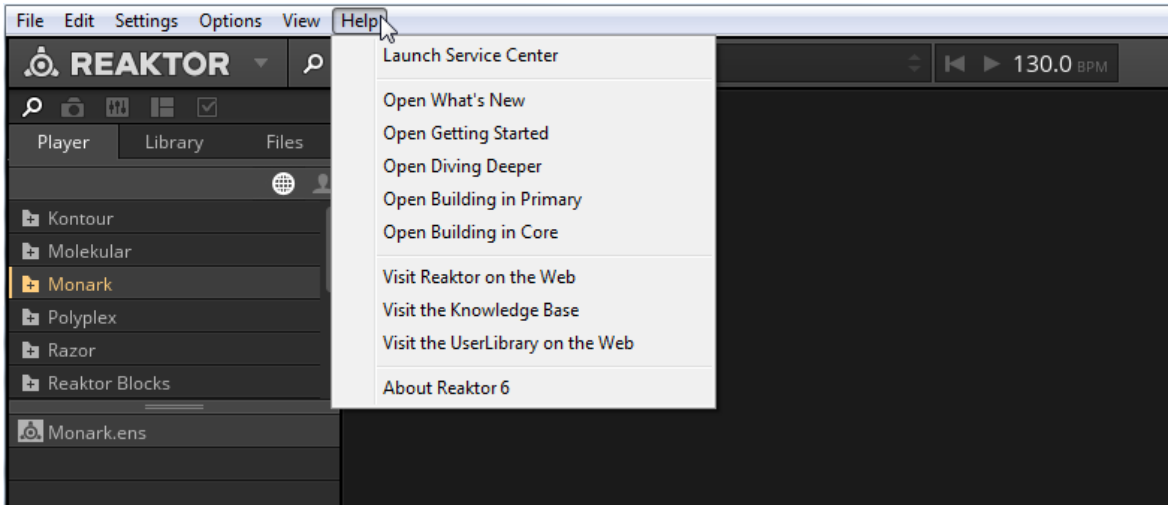
1 What Is New in REAKTOR 6 6
1.1 The REAKTOR 6 Documentation 6
1.2 Manual Conventions 7
2 REAKTOR 6.0 9
2.1 New Look and Feel 9
2.1.1 Updates to the Side Pane 9
2.1.2 Updates to the Structure View 10
2.2 Updates to Building 13
2.2.1 New Macro Libraries 13
2.2.2 In-place Editing 13
2.2.3 Panel Editing 14
2.2.4 Table Framework 15
2.2.5 New and Updated Primary Modules 15
2.2.5.1 Table Framework: New Modules 15
2.2.5.2 Table Framework: Changes to Existing Modules 16
2.2.5.3 Panel Modules 18
2.2.6 New Additions to Core 19
2.2.6.1 Unified Core Cell 19
2.2.6.2 Bundles 19
2.2.6.3 Scoped Bus 19
2.2.6.4 Sample Rate and Control Rate Clocks 19
2.3 GUI Core Cell 20
3 REAKTOR 6.0.3 23
3.1 Overview 23
3.2 Updates to the Side Pane 24

3.2.1 Presets and Snapshots Tab 24
3.2.2 Ensemble Bank Path 25
3.3 Updates to Panel Editing 26
3.3.1 Panel Element Grouping 26
3.3.2 Panel Element Layering 27
3.3.3 Updated Borders Property 28
3.3.4 GUI Core Cell 28
3.4 New and Updated Modules 33
3.4.1 Structure Comment Module 33
3.4.2 Anti-Alias Line Objects for Multi Display and Poly Display 34
3.4.3 System Info Updates 35
3.5 General Updates 37
3.5.1 Usage Data Tracking 37
4 REAKTOR 6.1 39
5 REAKTOR 6.2 40
5.1 Overview 40
5.2 Protected Content 41
6 REAKTOR 6.2.2 42
7 REAKTOR 6.3 43

1 What Is New in REAKTOR 6

In this document, REAKTOR 5 users can learn about the new features and improvements that have been added to REAKTOR 6 and subsequent versions, including references to the corresponding manuals in the REAKTOR 6 documentation. For new REAKTOR users, it is suggested to start with the REAKTOR 6 Getting Started document.

1.1 The REAKTOR 6 Documentation



The REAKTOR documentation is accessible from the Help menu

The documentation for REAKTOR 6 is divided into five separate documents, guiding you from loading and playing pre-built Ensembles to building your own Instruments.

- **REAKTOR 6 What Is New** is written for users who are already familiar with previous versions of REAKTOR and only describes the latest features in brief.

- **REAKTOR 6 Getting Started** is for new users. It is the only document needed for users who intend to use REAKTOR for loading and playing pre-built REAKTOR instruments and effects.
- **REAKTOR Blocks Quick Start Guide** is an online resource showing you how to use REAKTOR Blocks in the Rack format. Find it here: www.native-instruments.com/reaktor-blocks-quickstart
- **REAKTOR 6 Diving Deeper** expands on the concepts introduced in the Getting Started document. It provides more detail on subjects like Snapshots (REAKTOR's preset system), and introduces advanced topics like OSC control and combining multiple Instruments in one Ensemble.
- **REAKTOR 6 Building in Primary** shows you how to build your own Instruments in REAKTOR's Primary level. It focuses on a series of tutorials that guide you through building your first synthesizers, effects, and sequencers.
- **REAKTOR 6 Building in Core** describes the Core level of REAKTOR with its low-level building features, which can be used for implementing custom DSP algorithms. It includes reference information about the Core Macro Library, an comprehensive collection of DSP building blocks.

With the exception of the What Is New document, each of the documents listed above builds on the knowledge of the previous documents. While it is not necessary to read all of every document, some of the more advanced documents, like Building in Primary, assume knowledge from the previous documents.



In-depth information about using REAKTOR Blocks in Ensembles can be found in the REAKTOR Blocks Manual. To learn about opening manuals for REAKTOR content, refer to REAKTOR Content Documentation.

1.2 Manual Conventions

This section introduces you to the signage and text highlighting used in this manual.

- Text appearing in (drop-down) menus (such as Open..., Save as... etc.) and paths to locations on your hard disk or other storage devices is printed in italics.

- Text appearing elsewhere (labels of buttons, controls, text next to checkboxes etc.) is printed in blue. Whenever you see this formatting applied, you will find the same text appearing somewhere on the screen.
- Important names and concepts are printed in **bold**.
- References to keys on your computer's keyboard you'll find put in square brackets (e.g., "Press [Shift] + [Enter]").

1. Single instructions are introduced by this play button type arrow.

⇨ Results of actions are introduced by this smaller arrow.

An indented, gray paragraph contains additional, contextual information.

Furthermore, this manual uses particular formatting to point out special facts and to warn you of potential issues. The icons introducing these notes let you see what kind of information is to be expected:



The speech bubble icon indicates a useful tip that may often help you to solve a task more efficiently.



The exclamation mark icon highlights important information that is essential for the given context.



The red cross icon warns you of serious issues and potential risks that require your full attention.

2 REAKTOR 6.0

The following new features and changes have been added to REAKTOR 6.0.

2.1 New Look and Feel

REAKTOR's user interface has been completely reworked with a new look that provides better visibility of all elements. Most controls and options remain in the same place, so the overall layout of REAKTOR 6 will look familiar to REAKTOR 5 users. The following section provides an overview over the most important changes in the user interface.

2.1.1 Updates to the Side Pane

Updated Properties View

The Properties view has been updated to have a more uniform layout across all objects.



Information about the Properties can be found in the REAKTOR 6 Diving Deeper document, as well as the Building in Primary and Building in Core documents.

Updated Browser

The File Browser has been updated with a new design similar to the one used in other Native Instruments products.



Information about the Browser can be found in the REAKTOR 6 Getting Started document.

Controllers and Panelsets Tabs Separated

The Controllers tab is no longer a sub-tab in the Panel Control tab.



Information about the Controllers and Panelsets tabs can be found in the REAKTOR 6 Diving Deeper document.

2.1.2 Updates to the Structure View

The Structure View in REAKTOR has undergone a large redesign with many updates that enhance usability and readability:

- New color scheme.
- Wires now change color depending on their signal type.
- Anti-aliased wires.
- New icons for Macros and Core Cells.
- Optimization of the main context menu.

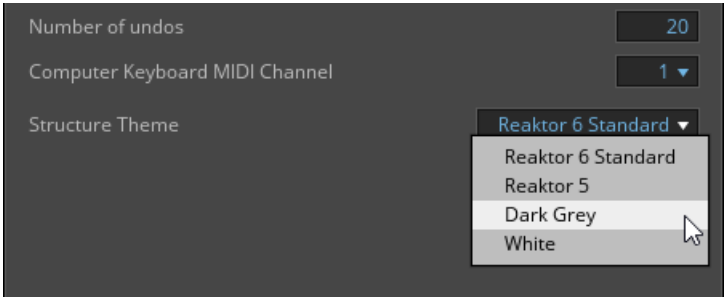
Additionally, users can now choose from a selection of four different color schemes and the appearance of Instruments and Macros has been improved with the new Flexible Look option.



In addition to the following section, information about the Structure View can be found in the REAKTOR 6 Diving Deeper document.

Color Scheme

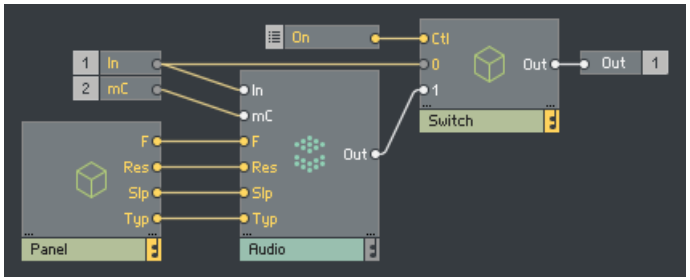
It is now possible to change the color scheme of the Structure View from the Preferences Window.



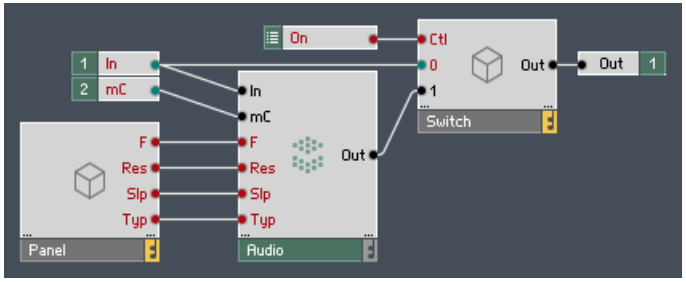
The Structure Theme menu in the Preferences Window.

There are four options available:

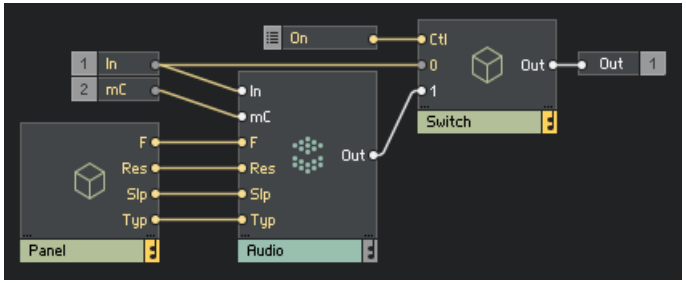
- Reaktor 6 Standard: The new default color scheme.



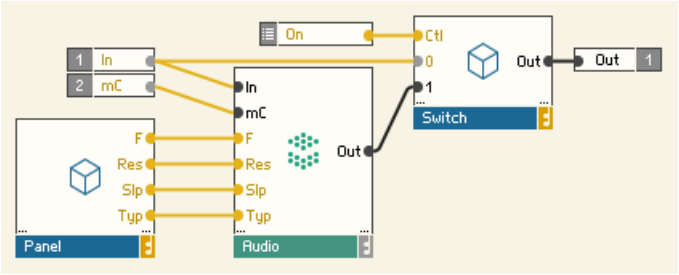
- Reaktor 5: The old default color scheme.



- Dark Grey: A darker variant of the Standard color scheme.



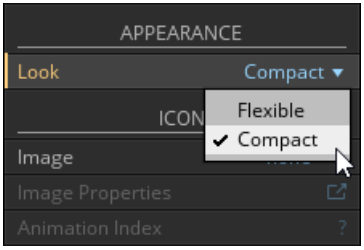
- White: A bright color scheme with white Modules and Macros.



The Reaktor 6 Standard option is selected by default.

Appearance of Instruments and Macros

Ensembles, Instruments, and Macros now have a new Look property for their appearance in Structure View.



The Appearance menu in a Macro's Properties.

This property has two options:

- Compact: This is the standard look as known from REAKTOR 5.
- Flexible: This new look allows for more flexibility in the way the Ensemble, Instrument, or Macro is displayed in the Structure. It lets you give longer names to Ports, and larger icons to Macros and Instruments.



The same Instrument displayed with Look set to Compact (left) and Flexible (right)

2.2 Updates to Building

2.2.1 New Macro Libraries

The Macro Libraries for both Primary and Core have both been completely redesigned and rebuilt. In both cases the new libraries have been designed to offer better tools and building blocks for creating your own Instruments.

Information about the Macro Libraries can be found in the REAKTOR 6 Building in Primary and Building in Core documents.

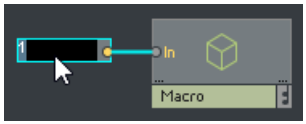
2.2.2 In-place Editing

It is now possible to edit the names of Modules and Macros from the Structure View, without entering their Properties. This also applies to the value of *Constant* Modules as well as the names of QuickBuses and QuickConstants in Core Cells.

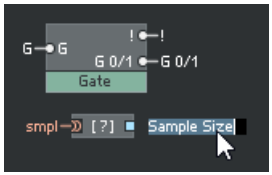
1. Double-click on a Module or Macro's label to rename it from the Structure View.



2. Double-click a *Constant* to edit its value from the Structure View.

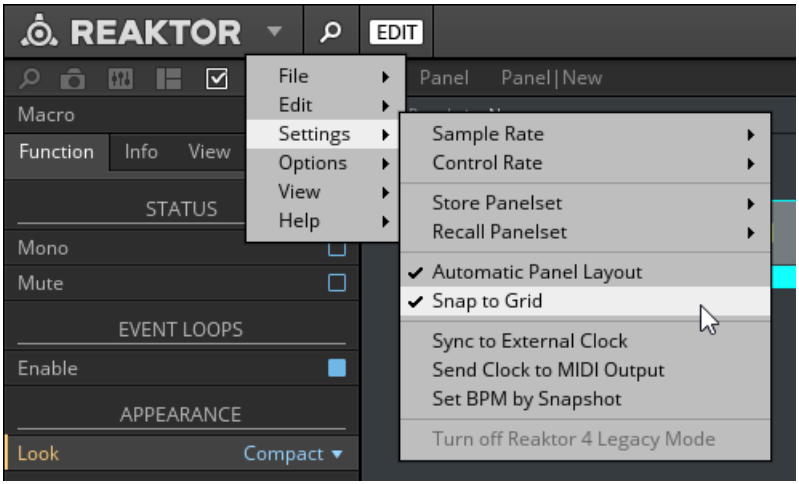


- Double-click a QuickBus or a QuickConstant to rename it from the Structure View.



2.2.3 Panel Editing

A new option has been created that allows you to bypass the 4 pixel grid when editing the panel of an Instrument or Ensemble. This way, you can move and resize panel elements with 1 pixel accuracy. This option is found in the Settings menu and is called Snap to Grid.



The Snap to Grid option in the Settings menu.

Information about panel editing can be found in the REAKTOR 6 Building in Primary document.

2.2.4 Table Framework

Table References are a new signal type in REAKTOR 6. They allow flexible and efficient sharing of data in the structure.

A table is a two-dimensional array of data, and Table References allow you to access this data anywhere in the structure. These properties make the Table Framework ideal for working with samples.

Core Cells read tables from a Table Reference like they have always handled tables and arrays. The advantage is that a Table Reference can exist in the Primary level, thus it can be used to share data between Core Cells, and can be stored in a Snapshot.

The Table Framework also makes the following possible:

- Sample information can be sent from the Sample Map to a Core Cell as a Table Reference.
- Drag and drop of samples onto the Instrument/Ensemble Panel.



In addition to the following sections, information about the Table Framework can be found in the REAKTOR 6 Building in Primary document.

2.2.5 New and Updated Primary Modules

2.2.5.1 Table Framework: New Modules

The Table Framework has introduced 3 new Modules:

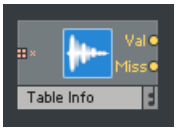
Table List



The Table List Module

This module is the main table manager and lets you maintain the list of currently active tables. It is the only table reference module with Snapshot support.

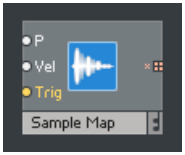
Table Info



The Table Info Module

The *Table Info* Module is used to read out a table's meta-information such as size, number of channels, sample rate, loop points, etc.

Sample Map



The Sample Map Module

This Module is a lot like the *Sampler* Module, but outputs a reference to the sample table, rather than the audio from the sample. In practical terms, this Module allows you to take sample data from the Sample Map and send it into a Core Cell.

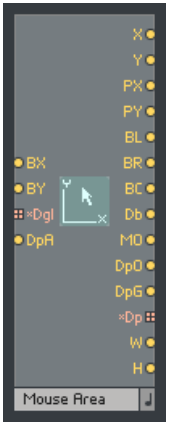
2.2.5.2 Table Framework: Changes to Existing Modules

A number of Modules have been updated to accept the Table Reference signal type:

- Merge
- Order
- Routers (all types)
- Value
- Switch
- To/From Voice

Mouse Area

The *Mouse Area* Module has undergone the most changes with the introduction of the Table Framework.



The Updated Mouse Area Module

It can now accept the dragging and dropping of samples into the Table List. As such it has been given a number of new inputs and outputs for this:

- **Inputs**
  - *DgI (Drag Input): Connect a Table Reference here to use the Module as a source for drag and drop.
  - DpA (Drop Accept): A value of 1 or greater at this input tells the Module to accept Table Reference drops.
- **Outputs**
  - DpO (Drop Over): Outputs a value of 1 when a Table Reference is dragged over the Mouse Area; outputs a 0 otherwise.
  - DpG (Drop Gate): Outputs different values depending on the drag and drop state: A value of 2 is sent when a file is dragged over the *Mouse Area*, followed by a 0; a value of 1 is sent when a file is dropped on the *Mouse Area* (if the DpA input is set to 1 or higher), followed by a 0.

- *Dp (Drop Output): Outputs the Table References that are dragged and dropped on the *Mouse Area*.

2.2.5.3 Panel Modules

Width and Height Outputs

Several Panel Modules have been updated to include two new outputs that send the value of the Module's Width (W) and Height (H) on the Panel.

These ports have been added to the following Modules:

- Mouse Area
- Poly Display
- Multi Display

Fonts

Both the Text and Multi Text Modules have been updated to include additional font options, including:

- Font Typeface: There are nine font types to choose from, illustrated below.



- Choose Font Color: Font color can be selected using the same pop-up window used in the Instrument/Ensemble Color Scheme Properties.
- Font Size (Pixels): Sets the font size. This option is disabled for the Legacy font.

Information about the Panel Modules can be found in the REAKTOR 6 Building in Primary document.

2.2.6 New Additions to Core

The following is a very brief overview of the main changes to the Core building language.

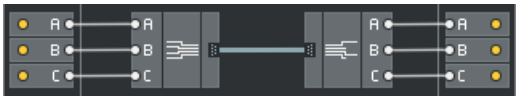
In addition to the following sections, information about Core can be found in the REAKTOR 6 Building in Core document.

2.2.6.1 Unified Core Cell

Core Cells are no longer divided into Audio and Event; there is now only one type of Core Cell. Like the old Audio Core Cell, the new Core Cell can accept both Audio and Event signals as inputs, but the new Core Cell can also output both Audio and Event signals.

2.2.6.2 Bundles

Bundles are a method of combining multiple wires into one. Wires can be packed into a Bundle and then unpacked into the individual wires later on.



Core Bundles.

2.2.6.3 Scoped Bus

Scoped Buses allow 'wireless' connections between several structure layers. Scoped Buses can also send Bundles between different parts of the structure.

2.2.6.4 Sample Rate and Control Rate Clocks

The Sample Rate is no longer treated like a QuickBus, but is instead bundled with its clock as a Standard Distribution Bus.



The Sample Rate Bundle using Scoped Buses.

A Control Rate clock is also included as a Standard Distribution Bus, giving you a separate clock for event signals. However, by default this clock is set to be the same as the Sample Rate, so it needs to be defined.

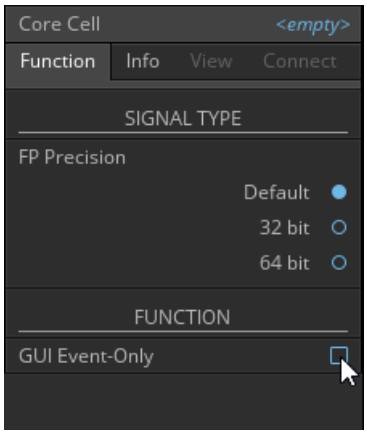
2.3 GUI Core Cell

In REAKTOR, Event and Audio signals are both processed in the Audio Thread, so if you created a Core Cell to process Mouse Area Events to manipulate a Multi Display, all of those calculations would happen in the Audio Thread.

The GUI Core Cells move their processing from the Audio Thread to the GUI Thread. This lightens the load on the Audio Thread, leading to better performance overall.

To create a GUI Core Cell:

1. Create a Core Cell as you would normally.
2. Enter the Structure of the Core Cell.
3. Navigate to the Function tab of the Core Cell's Properties.
4. Check the GUI Event-Only option (if the option is not visible, click in an empty space in the Core Cell's structure to make it visible).



⇨ The Core Cell is now a GUI Core Cell, and will be processed in the GUI Thread.



A GUI Core Cell

GUI Events

GUI Core Cell will only accept GUI Events, and will ignore all other Event and Audio signals.

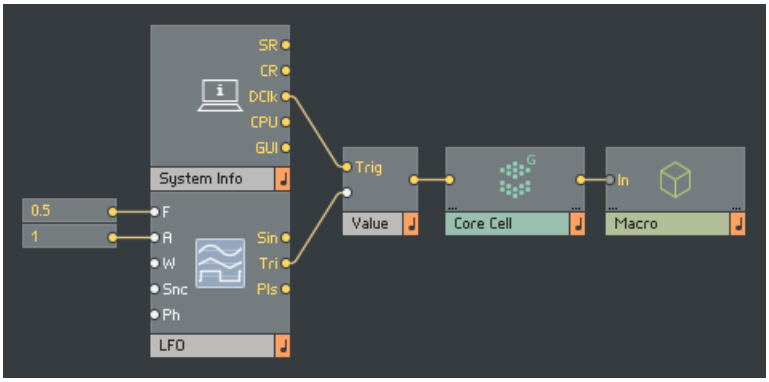
GUI Events include:

- Any Event produced via mouse interaction, for example: the user interacting with the Mouse Area module.
- Events produced from the DClk (Display Clock) output of the System Info Module.
- Iteration Events that were triggered by the Display Clock.
- Init Events, for example: Events from Constant Modules.

The following Events will be ignored by the GUI Core Cell:

- Event signals from Modules other than the System Info DClk.
- Event signals from regular Core Cells
- All MIDI messages.
- Event Signals from Panel elements that do not come from mouse interaction (for example: an automated Knob)

If you wish to use an Event Signal that is not a GUI Event, you can use the Display Clock and a Value Module to “re-clock” the signal.



Using the Display Clock to create GUI Events from an LFO

3 REAKTOR 6.0.3

The following new features and changes have been added to REAKTOR 6.0.3.

3.1 Overview

Updates to the Side Pane

- Presets and Snapshots are both accessible in the updated ↑3.2.1, Presets and Snapshots Tab
- ↑3.2.2, Ensemble Bank Path for Presets can be edited via the Ensemble Properties
- The Side Pane is resizable
- Keyboard navigation/editing of properties has been improved
- Certain options in the Properties tabs have been updated to Radio Buttons



In addition to the linked sections, information about the Side Pane can be found in the REAKTOR 6 Getting Started and Diving Deeper documents.

Updates to Panel Editing

- ↑3.3.1, Panel Element Grouping
- ↑3.3.2, Panel Element Layering
- Entering the Structure View via the panel has been changed to [Ctrl] + [Shift] + double-click for Windows, or [Cmd] + [Shift] + double-click for OS X.
- ↑3.3.3, Updated Borders Property
- ↑3.3.4, GUI Core Cell



In addition to the linked sections, information about panel editing can be found in the REAKTOR 6 Diving Deeper document.

New and Updated Modules

- ↑3.4.2, Anti-Alias Line Objects for Multi Display and Poly Display
- ↑3.4.3, System Info Updates
- ↑3.4.1, Structure Comment Module



In addition to the linked sections, information about Modules can be found in the REAKTOR 6 Building in Primary document.

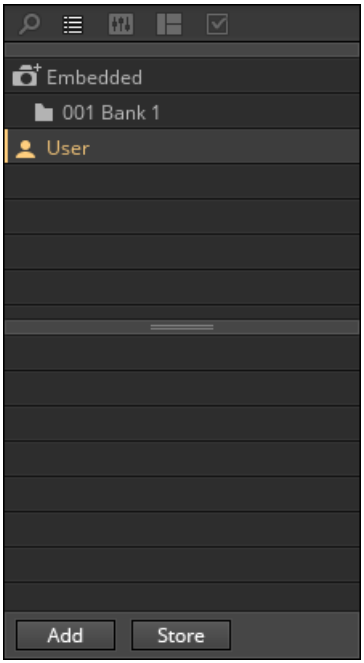
General Updates

- ↑3.5.1, Usage Data Tracking

3.2 Updates to the Side Pane

3.2.1 Presets and Snapshots Tab

With REAKTOR 6.0.3, Presets have been integrated into what was formerly the Snapshots tab of the Side Pane. This makes them easier to browse and access when you have an Ensemble open in REAKTOR.



The New Presets Browser

For those familiar with the Snapshots tab, the new Preset Browser should be easy to understand, as it follows the same paradigm.

One major change, besides the inclusion of Presets, is that the Edit menu has been removed. Functions from the Edit menu are now contained in a context menu accessed by right-clicking on a Snapshot or Snapshot Bank.

For more information on the Presets Browser, refer to the **Snapshots and Presets** chapter of the **REAKTOR 6 Getting Started** document.

3.2.2 Ensemble Bank Path

User Presets for Ensembles are located in the following directories:

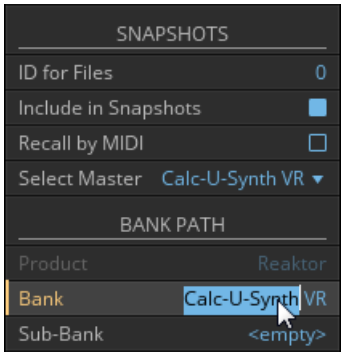
- On Windows: C:\Users\<User>\Documents\Native Instruments\User Content\Reaktor \<Bank Name>\

- On OS X: Macintosh HD:\Users\<User>\Documents\Native Instruments\User Content\Reaktor\<Bank Name>\

The Bank Name defines the folder into which Presets will be saved, and also the name that will appear in the KOMPLETE KONTROL and MASCHINE browsers. By default this will be the Ensemble name, but you can change that if you wish.

To change the Bank name:

1. Select the Ensemble and open the Properties tab.
2. Navigate to the Function sub-tab.
3. Scroll to the area called BANK PATH.
4. Enter the name you wish to use in the Bank property.



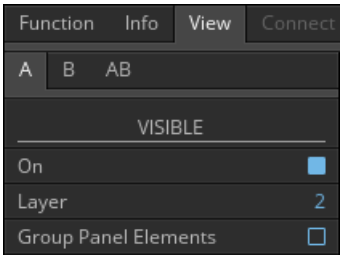
⇨ The Bank name will be used as Preset folder name, and the Bank name in the KOMPLETE KONTROL and MASCHINE Browsers.

3.3 Updates to Panel Editing

3.3.1 Panel Element Grouping

In previous versions of REAKTOR, if you wanted to move the Panel Elements contained within a Macro as one, you would have to click and drag on the Macro frame. This was fine for some circumstances, but if the Macro frame was not being used, it required an awkward shifting back and forth between a hidden and unhidden frame.

A new option has been added to all Macros: Group Panel Elements, located in the VISIBLE section of the View tab, allows you to group the Panel elements within a Macro and move them together as one.



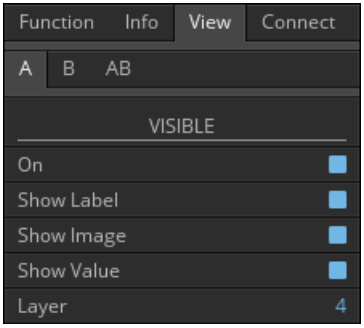
Group Panel Elements Property

When Group Panel Elements is active, you can move all Panel elements contained within the Macro by clicking and dragging only one of the elements on the Panel. This is especially useful for layered elements or elements that should be treated as one thing (for example, the modulatable knobs in the REAKTOR BLOCKS framework).

For more information on grouping Panel Elements, refer to the **Creating and Customizing** Interfaces chapter in the **REAKTOR 6 Building in Primary** documentation.

3.3.2 Panel Element Layering

A property has been added to the View tab of every Panel Element: Layer, located in the VISIBLE section, allows you to define how Panel Elements are layered within a Macro.



The Layer Property of a Knob

While you can still define how Panel Elements are layered by using Macros and sub-Macros, the Layer property allows you to assign an Element to one of 7 layers (0 to 6) within a Macro.

- Layer 0 is the lowest layer, with all other layers being placed on top of it.
- Layer 6 is the top layer, with all other layers being placed below it.

Panel Elements that are in the same layer will be ordered as in previous versions of REAKTOR, with the selected Element being placed on top of the others.

For more information on layering Panel Elements, refer to the **Creating and Customizing** Interfaces chapter in the **REAKTOR 6 Building in Primary** documentation.

3.3.3 Updated Borders Property

In previous versions of REAKTOR, the borders of Macros and Instruments had a fixed minimum, which users were unable to edit. This meant that Panel elements could never be placed at the very edge of the interface, even if the borders properties were set to zero.

In REAKTOR 6.0.3, this minimum has been removed and it is now possible to move Panel elements to the absolute edge of a Macro or Instrument.

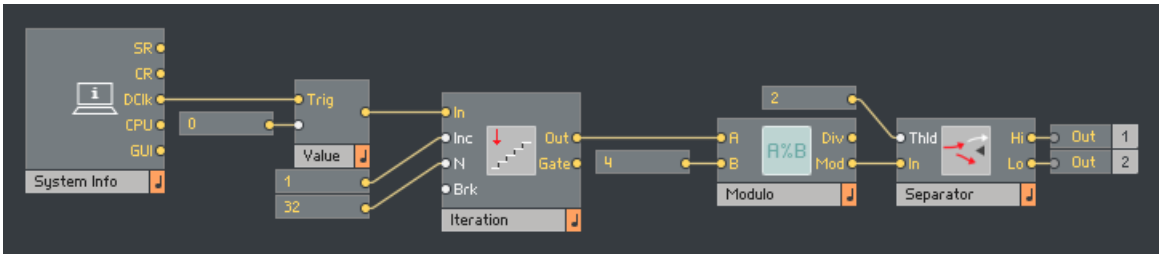
3.3.4 GUI Core Cell

REAKTOR uses two threads to process signals: the Audio Thread and the GUI Thread.

While all Audio Signals and all Audio Modules are processed in the Audio Thread, some Event processes will use the Audio Thread and others will use the GUI Thread. This depends on the source of the signal:

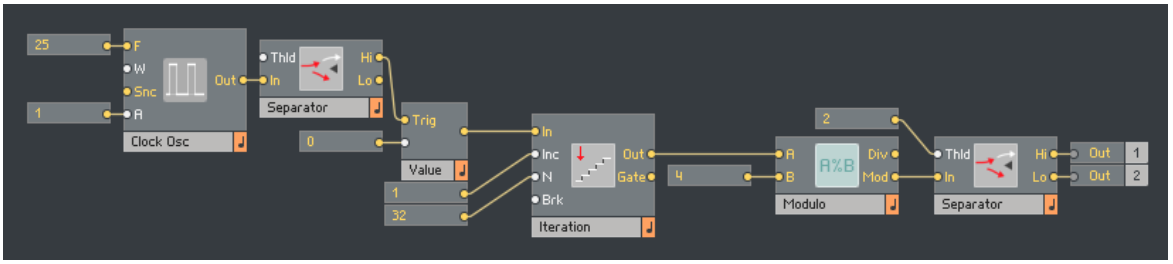
- Events generated from mouse and keyboard interaction will be processed by the GUI Thread.
- Events generated by the Display Clock will be processed by the GUI Thread.
- Events generated from the Control Rate Clock (e.g. the *LFO* Module) will be processed in the Audio Thread.
- Events generated from control sources outside REAKTOR (i.e. MIDI, OSC, and Host Automation) will be processed in the Audio Thread.

For example, the following structure will all be processed in the GUI Thread, as the triggering Event is from the Display Clock:



An Example of a Structure that will be processed in the GUI Thread

The next example, although functionally very similar, will all be processed in the Audio Thread, as the Events from the *Clock Oscillator* Module are generated in the Audio Thread:



An Example of a Structure that will be processed in the Audio Thread

While the Primary level of REAKTOR can deal with these two threads simultaneously, Core Cells will only process one of these threads at a time.

Since a Core Cell must always be ready to receive signals from either thread, it manages this by “locking” its inputs while it processes an Event. If the Core Cell receives an Event from the Audio Thread, it will start to process that Event; if it then receives an Event from the GUI Thread before it has finished the audio process, it will lock the GUI Thread until the audio process has finished. This lock is a global lock, so it will affect everything trying to use the locked thread in the Ensemble structure.

Generally this is not a problem, as modern CPUs can process the Core Cell structures fast enough so that a user would not notice; however, in some very CPU heavy structures this can lead to low frame rates and/or audio dropouts as one thread stops the other for too long.

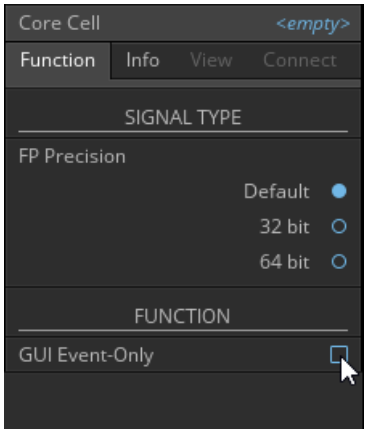
To combat this REAKTOR 6.0.3 has introduced the GUI Core Cell, which only processes Events from the GUI Thread and ignores all others. It will never lock the Audio Thread, as it has no interaction with that thread.

Using GUI Core Cells leads to better performance if you have a particularly complex GUI process in your structure. A common example being a display that uses a high number of iterations to drive a *Multi Display* Module.

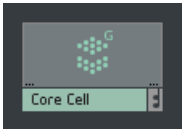
Creating a GUI Core Cell

To create a GUI Core Cell:

1. Create a Core Cell as normal.
2. Enter the structure of the Core Cell.
3. When in the Core Cell's structure, click on an empty space in it to select the Core level properties.
4. In the Function tab of the Core Cell's Properties, check the GUI Event-Only option.



⇨ You have now created a GUI Core Cell.



A GUI Core Cell

Notes on Valid GUI Signals

The following signals will be accepted and processed by the GUI Core Cell:

- Events from Panel Elements as long as they were generated by mouse or QWERTY keyboard interaction.
- Events generated by the DClk (display clock) output of the *System Info* Module.
- Events from Event processing Modules, as long as the triggering Event came from the GUI Thread (for example, an *Iteration* Module driven by the display clock)
- Some MIDI Events generated by REAKTOR (see the next sub-section for more information).

The following signals will be ignored by the GUI Core Cell:

- All Audio signals. Audio In Ports in the GUI Core Cell are disabled and marked in red.
- MIDI Events generated from MIDI controllers, external programs, or DAWs.
- OSC Events.
- Events from Panel Elements generated by automation, MIDI, or OSC control.
- Events from Modules that use the Control Rate Clock (e.g. the *LFO* Module)
- Events from any Event processing Module if the triggering Event came from the Audio Thread.

Other cases:

- IC (Internal Connection) signals are essentially like wires, so an Event from an IC Module will be the same as the Event that triggered it.
- Clock signals from the Standard Distribution Buses (i.e. the Sample Rate Clock) are not valid in GUI Core Cells and will display an error if you try to use them.

Notes on MIDI

As listed in the previous section, a GUI Core Cell will sometimes be able to process MIDI, depending on the source of the MIDI event.

As a general rule, MIDI events generated by devices and programs other than REAKTOR will always be processed in the Audio Thread. MIDI events generated inside REAKTOR depend on the source of the MIDI:

- The QWERTY keyboard does not natively send MIDI, so REAKTOR takes the keyboard signals in the GUI Thread, and then generates a MIDI message in the GUI Thread; therefore any MIDI from the QWERTY keyboard is processed in the GUI Thread.
- If a MIDI output Module receives its Events from a GUI Module, or a Panel Element that is being controlled with the mouse, then the MIDI signal will be part of the GUI Thread.

As long as the MIDI signal is inside REAKTOR (for example, MIDI connections between Instruments within an Ensemble) the MIDI will remain in the thread from which it was generated.

This information is important as sometimes you may make a GUI Core Cell that is accepting MIDI, test it on your machine using the QWERTY keyboard and find it to be working, but when you send it to a collaborator, they may find it to be broken, as they are testing it in a DAW. So always be careful when using MIDI with GUI Core Cells.

Notes on Panel Elements

Similar to MIDI Events, the Events from Panel Elements can be from either the Audio Thread or the GUI Thread depending on how the Panel Element was used.

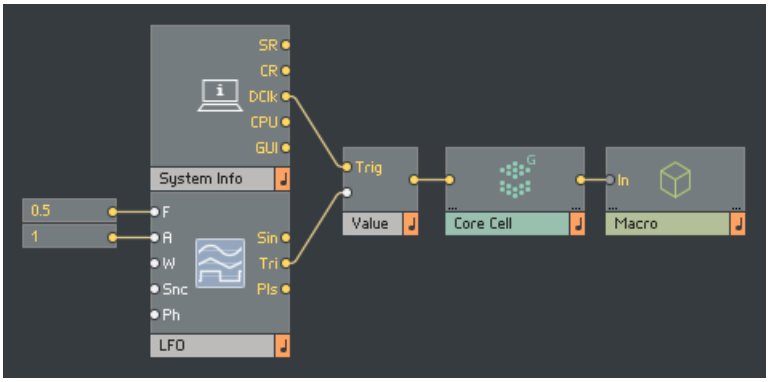
- Events from Panel Elements that were generated by mouse or QWERTY keyboard interaction will be generated in the GUI Thread.
- Events from Panel Elements that were generated by MIDI, OSC, or Automation control will be generated in the Audio Thread.

And so, just like with MIDI Events, it is important to make sure your GUI Core Cells will work in both scenarios.

Note that the *Mouse Area* Module, which only sends signals from direct mouse interaction, is one of the few Panel Elements Modules guaranteed to send valid GUI Events.

Creating Valid Events with the Value Module

If you wish to connect an invalid signal, or a signal from a source that is not guaranteed to be from the GUI Thread (for example a *Knob* Module), you can “re-clock” the signal to the GUI Thread by using the *Value* Module and the Display Clock.



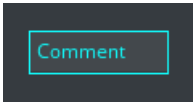
A Structure for Attaching an LFO to a GUI Core Cell

3.4 New and Updated Modules

3.4.1 Structure Comment Module

The Structure Comment Module is a Module that allows you to annotate your Ensemble's Structure. It appears as floating text within the Structure View of REAKTOR.

The Comment Module is located in the Auxiliary section of the Built-In Module submenu.

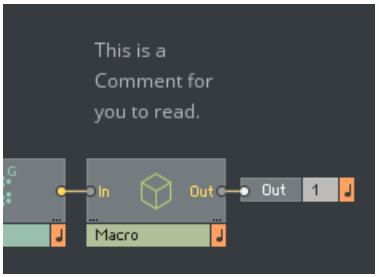


The Comment Module

1. Double-click on the Comment Module to edit the Comment text.



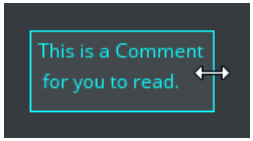
⇨ When unselected, the Comment will appear as floating text in your Structure.



A Comment in the Structure View

Comments automatically resize vertically to display all of the text, but you are able to define the width manually:

1. Click and drag on the Comment's edge to resize it.



Comments are currently only available in the Primary level of the Structure; they are not available inside Core Cells.

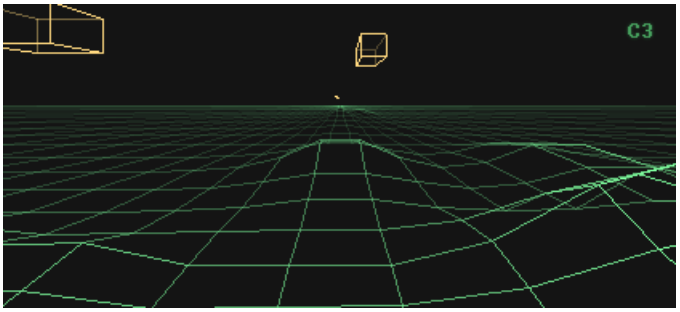
3.4.2 Anti-Alias Line Objects for Multi Display and Poly Display

The Multi Display and Poly Display Modules now have two additional object types:

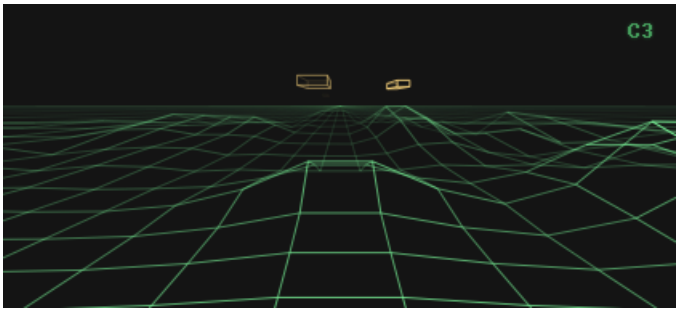
- **Object value -5**: Anti-aliased line from (X1,Y1) to (X2,Y2)
- **Object value -6**: Anti-aliased line from (X1,Y1) to (X1,Y1) of the next object.

If you are familiar with the Multi and Poly Displays, then you will notice that these are duplicates of the line objects, but with anti-aliasing.

Anti-aliasing smooths the lines, making them less pixilated.



A Display Using Regular Line Objects



The Same Display Using Anti-Aliased Lines

With Anti-aliasing, a new input port has been added to the Multi and Poly Displays:

- LW (Line Width): defines the pixel width of an anti-aliased line.

3.4.3 System Info Updates

The System Info Module has been updated with a new output port and a new property option.



The System Info Module

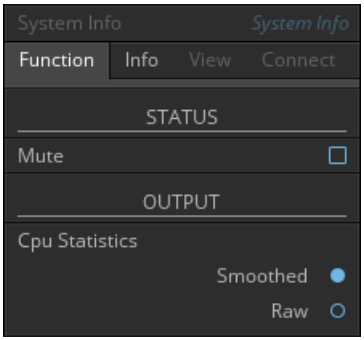
GUI Port

The new GUI Port outputs the load on the GUI Thread as a percent.

This port can be used to measure the performance of GUI Core Cells and Panel elements.

CPU Statistics

A new option in the Function tab of the Module's Properties lets you define how the CPU load is measured.



The New System Info Properties

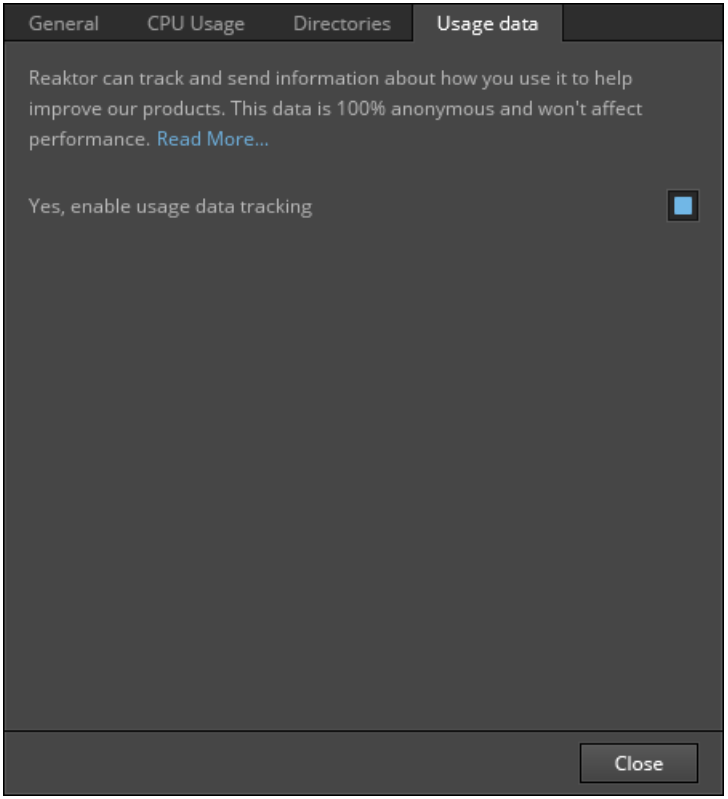
- Smoothed: the CPU measurement will be averaged over small periods of time, giving you a general measurement of CPU performance. This option produces more readable results, but is less accurate.
- Raw: the CPU measurement is constantly updated with the precise CPU load at that time. This option is more helpful when searching for CPU peaks, but is difficult to read without additional processing.

3.5 General Updates

3.5.1 Usage Data Tracking

REAKTOR tracks anonymous usage data and sends that data to Native Instruments so that they can improve future updates by seeing which features are most often used and in which environments REAKTOR is most commonly used.

Usage Data options are located in the Preferences dialog.



The Usage Data Tab of the Preferences Dialog

- In the Usage Data tab, click on the Read More… link to be directed to the Native Instruments webpage with more information on Usage Data tracking.
- You can opt in or out of Usage Data tracking by clicking on the checkbox labelled Yes, enable usage data tracking.

Data Being Tracked

The following data will be transmitted by Usage Data Tracking:

- System profile (your Operating System, Audio Drivers, etc)
- Current location (country and city)
- Data about your interaction with the software (e.g. actions you perform or audio interfaces you connect)

Data Not Being Tracked

The following data will not be gathered by Usage Data Tracking:

- Personal data (Name, email address, payment details, serial numbers, etc.)
- User content (file names being referenced, samples, songs, tags, comments, mappings etc.)
- Any user activity outside of the NI application

4 REAKTOR 6.1

The following new features and changes have been added to REAKTOR 6.1.

Synchronization using Ableton Link

REAKTOR 6.1 adds Ableton Link to its set of synchronization features, complementing MIDI Clock and OSC Sync.

Ableton Link is a protocol that synchronizes beat, phase and tempo of Link-enabled applications on the same computer or over a shared network. This means you can conveniently keep applications synchronized across different devices or join a group jam with others with minimal setup. Connecting applications via Link provides you with the ability to synchronize them to a shared timeline and change the tempo globally from each of the connected applications.



Information about Ableton Link can be found in the REAKTOR 6 Getting Started document.

5 REAKTOR 6.2

The following new features and changes have been added to REAKTOR 6.2.

5.1 Overview

Content Protection

- ↑5.2, Protected Content by third-party developers and Native Instruments.
- Structure locking for official content products.
- Sample protection for official content products.
- Player tab sorts products by vendor.
- Demo screen can display multiple products (instruments) and their demo time.
- Native Access activation callback implemented in REAKTOR.



In addition to the linked section, information about protected content can be found in the REAKTOR 6 Getting Started document.

General Updates

- Fixed: Enabling [caps lock] prevents deletion of modules with [delete] key.
- Fixed: Deleting one character in search box deletes whole unmarked word.
- Fixed: No MIDI input from computer keyboard after renaming a bookmark.
- Fixed: Performance regression when using Resynth, Graincloud and Pitchformer Modules.
- Fixed: Drag & drop of Primary Module files into Structure.

5.2 Protected Content

REAKTOR 6.2 adds the option to protect official content products by third-party developers and Native Instruments. This means that the Structure of the Ensemble – or Instrument in the case of a protected module for REAKTOR Blocks – cannot be accessed by the user.



If you are a developer and want to contact Native Instruments about your content products, please use the email address provided on the following website: https://www.native-instruments.com/en/specials/komplete/developer/



Information about protected content can be found in the REAKTOR 6 Getting Started document.

6 REAKTOR 6.2.2

The following new features and changes have been added to REAKTOR 6.2.2.

Core Macro Library Updates

The Core Macro Library has received significant updates, including new filters, phaser, flanger, overdrive, bit and sample reduction effects, a compressor, synced LFOs, and the LFO toolkit.



Information about the Core Macro Library can be found in the REAKTOR 6 Building in Core document.

7 REAKTOR 6.3

The following new features and changes have been added to REAKTOR 6.3.

REAKTOR Racks

You can now patch with REAKTOR Blocks in the new Rack format. Racks facilitate analog-style modular synthesizer patching. In a Rack, you can arrange, connect, and control Blocks directly in the Panel. The patch is also mirrored in the Structure, allowing you to choose your preferred patching workflow. All settings including the arrangement of Blocks and the connections made between them are saved and recalled with your host sequencer project (host chunk saving).

- The **REAKTOR 6 Getting Started** document provides information about the new Rack format, gives you an overview of the Racks, and shows you how to manage them.
- The **REAKTOR Blocks Quick Start Guide** shows you how to use REAKTOR Blocks in the Rack format. Find it here: https://www.native-instruments.com/reaktor-blocks-quickstart/
- Learn all about the Blocks and their controls in the application by using REAKTOR's Info Hints. More information about Info Hints can be found in the **REAKTOR 6 Getting Started** document.

Panel Ports

Ports of Instruments and Macros can now be made visible and used on the panel. This way you can add modular-style patching to your Instrument. More information can be found in the **REAKTOR 6 Building in Primary** document.

Switching Views for Multiple Instruments

You can now switch between the A and B view of all Instruments contained in an Ensemble at the same time. This is useful for switching between views with and without Ports on the panel, for example. More information can be found in the **REAKTOR 6 Building in Primary** document.

Borders

The borders at the edges of an Instrument panel can now be adjusted. This way you can define how close panel elements can be placed to the edges. More information can be found in the **REAKTOR 6 Building in Primary** document.