

## Reaktor 5 Mode - New Initialization Algorithm

This document describes a new initialization scheme that is introduced with Reaktor 5. It is used, if the Reaktor 4 Legacy Mode is disabled in the Ensemble Properties.

**Attention! We urgently recommend to use this new mode for future compatibility!**

The following describes the new module and signal initialization step by step.

### First Step - Sort Modules

Sorting-Algorithm: According to a list of sinks, activate modules from each sink leftwards.

If a Modul

- has no other leftward Module, or

- if all the leftward Modules it has are already being active or

- are marked as being passed by the current search,

the module gets marked as active and is put into the Sorted-Modules-List.

This Sorting can be displayed thru a new Menu-Entry:

"System"->"Debug"->"Show Event Initialization Order"

Remark: for Modules in Loops (Event- and also Audio-Loops) the order of the Sinks in the Sinks-List can be relevant for the Sorting of the Modules!

The order of the Sinks is determined by searching Sinks recursive in all Instruments and Macros, starting from Ensemble-Level.

Remark: Because on Ensemble-Level the only Sink is the DAC-Module, the DAC-Modules is always the first one in the Sinks-List.

Additionally, in this Sorting-Algorithm a List of Modules having implicit Constants is generated.

Implicit Constants are OutPorts of Modules which do not change thru any processing of the Module. For example the Snap Value Array has an OutPort for the number of Elements in its array, which can only be changed in the Properties.

These OutPorts are considered as Constants and are Initialized separately from the rest of the Module.

Here is a List of the Modules behaving in this way:

Snap Value Array

OutPort "N" - Number of Elements

Multi Picture

OutPort "N-1" - Number of pictures in the animation - 1

Multi Display

OutPort "NP" - Number of pictures in the animation

Multi Display Array

OutPort "NP" - Number of pictures in the animation

OutPort "NO" - Number of Elements

Audio Table, Event Table

OutPort "DX" - X size of the table

OutPort "DY" - Y size of the table

XY

OutPort "MX" - last Position

OutPort "MY" - last Position

## **Second Step - Process Init-Functions**

For each Implicit Constants and for each Module of the Sorted-Modules-List execute a Init-Function.

If the Module has an Event-Connection to another Modules InPort, notify this InPort that a Value has arrived. If the InPort is a polyphonic one and the Module is polyphonic, this is done for each voice and the InPort has a notification flag for each voice.

The Modules have are divided in Groups:

### **Source-Modules**

For Faders, Buttons and so on, the Init-Function executes basically only a recall of the last Value, putting it to the Modules OutPort and notify the connected Event-InPorts.

They are called Source-Module, which reflects that their Output-Values are not a result of any Event- or Audio-Processing.

Most of these Modules do not have any InPort.

Here is the List of Modules behaving in this way:

### **Modules with one OutPort**

Fader, Knob  
List  
Button  
Note Pitch  
PitchbendSingle Trig. Gate  
Gate  
Single Trig. Gate  
Sel. Note Gate  
On Velocity  
Off Velocity  
Controller  
Ch. Aftertouch  
Poly Aftertouch  
Sel. Poly AT  
Snap Value  
Constant

OutPort is set;

### **Modules with multiple OutPorts**

Start/Stop

only OutPort "G" is set;

The Value is equal to

- 0.0 if the Clock is not running or

- the chosen OUTPUT VALUE from the Properties-  
Function-Page if the Clock is running.

Master Tune/Level  
Voice Info  
Tuning  
System Info  
Note Range  
Midi Channel

all OutPorts are set;

Sampler FM  
Sampler Loop

Sample Lookup  
Tapedeck 1-Ch  
Tapedeck 2-Ch

The OutPort "Len" is set;

Remark: for Sampler FM and Sampler Loop this depends on the current selected Sample;

Snapshot:

OutPorts "Snp", "Bnk", "Mph", "Amt" are set;

OutPort "A" is set when there is currently a A-Snap for morphing selected

OutPort "B" is set when there is currently a B-Snap for morphing selected

OutPort "Sw" is set to the last selected Switch-Position (see Snapshot-Window)

### **Modules with Event-InPorts**

e.g. for a Hybrid Math Module like the Add-Module, the Init-Function works like this:

If the Module is in Event-Mode (green or red coloured Ports) then do for each InPort

if the InPort is not wired or if it is wired but the wiring is not active (e.g. a Port is Muted) then pass a 0.0 Value to the InPorts Event-Function.

otherwise this InPort is "really" wired (it is wired to a active non-Transparent Module and the wiring is active).

Then the algorithm looks up for each Voice if a notification has arrived. If this is so, pass the value to the InPorts Event-Function.

### **special Processing-Modules**

Event Smoother

this Module has no Event-InPort, nevertheless the current value at its Audio-InPort is set to the OutPort;

Core Cell Audio, Core Cell Event

these have special Init-Functions; see Reaktor Core Cell-Documentation;

#### Order

the incoming Event is set only to the 1st (topmost) OutPort. The other OutPorts process later, see "Fourth Step - Process all Order-Modules".

#### Iteration

the incoming Event is set to the OutPort, the rest of the iteration is done later, see "Fifth Step - Process all Iteration-Modules".

Examples when it is possible, that no Value arrives at an Event-InPort in spite there is a active connection:

The InPort is wired to a Separator-Modules "Low" or "High" OutPort. Only one of these two OutPorts will fire an Event at Initialization. If both OutPorts are connected to two InPorts, only one InPort will not receive a notification.

The InPort is wired to a To-Voice-Module. As said above, the notification takes place for each Voice. A To-Voice-Module only sends for one Voice at Initialisation, so the InPort gets only a notification for that Voice.

While this algorithm processes, it can happen that it tries to notify a Event-InPort of a Module which has already been passed by the algorithm. This is what we call a Event-Loop.

These InPorts where an Event-Loop occurred are stored in a special List and are processed in the next step.

### **Third Step - Process all Eventloop-InPorts**

For each InPort in the Eventloops-InPort List the notified Value is fired as a normal Reaktor Event.

### **Fourth Step - Process all Order-Modules**

For all active Order Modules, set the 2nd and 3rd OutPort and send Events as normal Reaktor Events.

### **Fifth Step - Process all Iteration-Modules**

For all active Iteration Modules, iteration-processing is now done.

### **Sixth Step - Process all SnapValueArray**

For all active SnapValueArray Modules, iteration-processing over all Elements is done if the Flag "Self Iteration" from the Function-Properties Page is set.