TVSTInstrument API

Version 1.4 22 April 2020

This update only supports VST3 and architecture: Separate processor/controller but combined to one TVST3Instrument.

(To distingush from the earlier framework for VST3, this repository uses ‘VST5’ as identifier.)

In this framework you find Delphi wrapper code to create VST3 Instruments.

In this version there is a seperation between a processor and controller, as defined in the VST 3 specification. However you only have to create one class supplying the required functionality.

In the document Description SimpleSynth the working of the simplesynth is explained.

You can read this and that document in any order.

To make the TMyVSTPlugin available as plugin, you have to describe it in a VSTInstrumentInfo record and make it available through the function GetVSTInstrumentInfo. If all goes well your plugin will be instantiated in your DAW. From there you have control over it with this API.

Please use the RMSMyVST5 as a starting base as it include the startup code.

Declaration:

The function GetVSTInstrumentInfo must be implemented delivering the following information:

vst3processorid, vst3controllerid: Unique UUIDs for you’re a VST3 plugin

(remark: This is a bit strange, but for now this is necessary…)

vst3instument:

the instrument class, derived from TVST3Instrument, which implements the plugin

Ecl The class name for your editor: here: TFormMyVST

isSynth true if this is a synth.

softMidThru if you want to pass all midi events to the next plugin

and a few other information fields.

API. The API is split into two parts:

**Processor**

Methods concerned with the audio processing part. You should not update the UI from a Processor call.

**Controller**

Methods concerned with the parameters, presets and editor

|  |  |
| --- | --- |
| **Processor** | All methods are virtual and optional for processing |
| proc OnSysexEvent(s:string); | s starts with $F0 and ends with $F7 |
| proc OnMidiEvent(processor:boolean;byte0,byte1,byte2:byte); | Remark: when an event arrives, this method is called twice, once for the processor and once for the controller |
| proc MidiOut(const b1, b2, b3: byte); | To send midi out |
| proc Process32(samples,channels:integer;  inputp, outputp: PPSingle); | Here you process your audio. Inputp and outputp are arrays of array with the first subscribe the channel and the second the sampleposition. See example. |
| proc OnSamplerateChanged(samplerate:single); |  |
| proc OnPlayStateChanged(playing:boolean;ppq:inte); | Called when the DAW changes play state, or position |
| proc OnTempoChanged(tempo:single); |  |
| proc updateParameter(id,value) | Virtual method called when a Host parameter changes (NON-UI thread). You should update your ‘Processor’ or ‘Model’ |
| Procedure GetParameterDescription | Called to retrieve the parameters and some definition info you use. |

|  |  |
| --- | --- |
| **Controller** |  |
| func getParameterAsString(id:integer;value:double):string; | Value is between 0 and 1! |
| proc DoupdateHostParameter(id,value) | Updates a parameter on the Host |
| proc updateParameter(id,value) | Virtual method called when a Host parameter changes (UI thread). You should update your UI. |
| proc ResendParameters | Resends all parameters through UpdateEditorParameter (must be called from UI thread, e.g. when opening your plugin editor) |
| function getEditorClass:TformClass; | Virtual method. You can override the default creation of the editor class (as defined in GetVST3InstrumentInfo) here (normally not needed) |
| proc OnMidiEvent(byte0,byte1,byte2:byte); | Remark: See processor |
| proc OnEditOpen | Virtual method. Called when the editor opens. It could be wise to call ResendParameters and to create a mechanism so when a UI element changes you send the changed value to the host. |
| proc OnProgramChange(prgm:integer); | Virtual method. Only needed if you want to show the preset number in the UI. |
| proc OnEditClose | Virtual method. For cleanup, but normally not necessary. |
| proc OnFinalize | Virtual method. For cleanup, you should release all resources, especially timers. |
| Proc OnEditIdle | Called regularly if the Editor is open |

Missing from the API (will be implemented on request)

* Setting/Getting preset names. Setting the preset number (to host)
* MidiIn stuff like MPC, aftertouch (for VST3)