Description SimpleSynth

SimpleSynth is an example of how to use TVSTInstrument.

As you can see in the RMSVST2, RMSVST3 and FruityPlug directories, you can create various versions of a plugin for various frameworks.

This document describes the architecture of TMyVstPlugin

TMyVSTPlugin inherits from TVSTInstrument to create a simple synth plugin: It only reacts to Midi Note On/Off and has a few parameters for Cutoff, Resonance and PulseWidth.

**UMyVST.pas**

Defines the TMyVSTPlugin. In GetVSTInstrumentInfo it describes the architecture:

The class to create is TMyVSTPlugin and the editor class is TformMyVST.

In OnInitialize

* the real synth is created: TSImpleSynth.Create.
* Three parameters are added.

In Process we need to fill the audiobuffer.

For this we call FsimpleSynth.process which generates 1 sample. We copy it to all channels and call this function ‘samples’ time.

In OnEditOpen we are notified that the editor will open. Note that the actual editor is an instance of TFormMyVST. This form gets two callbacks when a parameter is changed in the editor or a key is pressed. Then we call resendParameters which resends the parameter values.

In UpdateEditorParameter we update the editor

In UpdateUpdateProcessorParameter we update the synth

When a key is pressed this is forwarded to the SimpleSynth (onKeyEvent)

**UMyVSTDSP.pas**

Here TsimpleSynth is implemented. Allthough the internals might be interesting, the main focus are the public methods, updateParameter, Process and onKeyEvent. These have been discussed in UMyVST.

**UMyVSTForm.pas**

The OnKeyEvent and UpdateHostParameter are filled in from UMyVST and make it possible to send keys to the SimpleSynth and send parameter changes to the Host. UpdateEditorPreset and SetKey are used to show the correct settings. SetPreset is just a simple call showing the current preset. Note that when changing preset, the parameters will automagically update here and in the SimpleSynth.