

Example Plugin Summary
Bruce Krell, PhD
BKrell@SWArchitects.com

>> WPFExampleStereo

- A: extends simplest plugin to stereo inputs as well as stereo outputs without stereo inputs, pan does not appear to work
- B: replicating a mono input to stereo output and applying the pan does not shift sound from left -> right, right -> left
- C: implements exception guard to copy inputs to outputs if algorithm becomes unstable based on choice of algorithm parameters

>> WPFExampleGUI

Libraries: WpfEditorView
Slider

- A: define a custom user interface by employing a UserControl, Canvas, and overriding GetEditorView, library component WpfEditorView
- B: employs WFP controls TextBlock, Button
- C: demontstrates how to service handlers for those controls within plugin
- D: with Canvas, uses SetLeft, SetTop to position controls anywhere in window
- E: implements a slider wrapper class to enable addition of a slider with single line of code
- F: uses threshold to safely abort processing during silence
host may call plugin method Process and cause algorithm instability

>> WPFExampleGUICustom

Libraries: WpfEditorView

- A: define a custom user interface by employing a UserControl, Canvas, and overriding GetEditorView, library component WpfEditorView
- B: employs WFP controls TextBlock, Button AND ComboBox
- C: demontstrates how to service handlers for those controls within plugin
- D: shows how to retrieve selected value in ComboBox

>> WPFExampleGUIGraphics

Libraries: WpfEditorView

- A: define a custom user interface by employing a UserControl, Canvas, and overriding GetEditorView, library component WpfEditorView
- B: demonstrates use of an embedded canvas for draw lines/curves using application specific data
- C: implements hit test to locate mouse click location inside drawing canvas
- D: shows conversion from app data value to canvas location and from canvas location to app specific data

NOTE: many plugins present a polyline to user that can be manipulated by mouse clicks. B,C, and D are necessary to implement this support

>> WPFExampleGUIKnobs

Libraries: WpfEditorView
DialMgr
Rotary Control

- A: define a custom user interface by employing a UserControl, Canvas, and overriding GetEditorView, library component WpfEditorView
- B: place controls onto Canvas at specific locations using SetTop, SetLeft
- C: employ your own non-WPF user control in the custom interface (Rotary Control)
- D: ensure that your own non-WPF user control can be accessed across thread boundaries
- E: mix a non-WPF (RotaryControl) user control with WPF controls
- F: ensure that your user interface is stable, given that the host executes plugin methods in ways that are not specified and arbitrarily implemented by the host
- G: ensure that your plugin is stable if your algorithm fails due to inappropriate parameter values set by the user or when executing method Process during silence
- H: do not update local variables if user has not changed them in user control

NOTE: vstsdk executes plugin in a separate thread
any user control implement in GetEditorView has thread affinity to UI thread
plugin access to value of user control crosses thread boundaries
user control implementation must wrap access to value in Dispatcher.Invoke

THE STEPS TO FOLLOW ARE NUMBERED THROUGHOUT THE CODE

>> Deployment Batch Files

- A: created folder CopyFilesToDeployment
- B: added a bat file for each sample VST plugin to deploy appropriate files to Cackewalk search folder
- C: edit Source variable in each file to point to your folder where VST Plugin file executables store
- D: bat file MUST run in administrator mode because Cakewalk search folder on drive C: