

**Expert Sleepers  
Spectral Conquest  
v1.0.0  
User Manual**

Copyright © 2010 Expert Sleepers. All rights reserved.

This manual, as well as the software described in it, is furnished under licence and may be used or copied only in accordance with the terms of such licence. The content of this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Expert Sleepers. Expert Sleepers assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

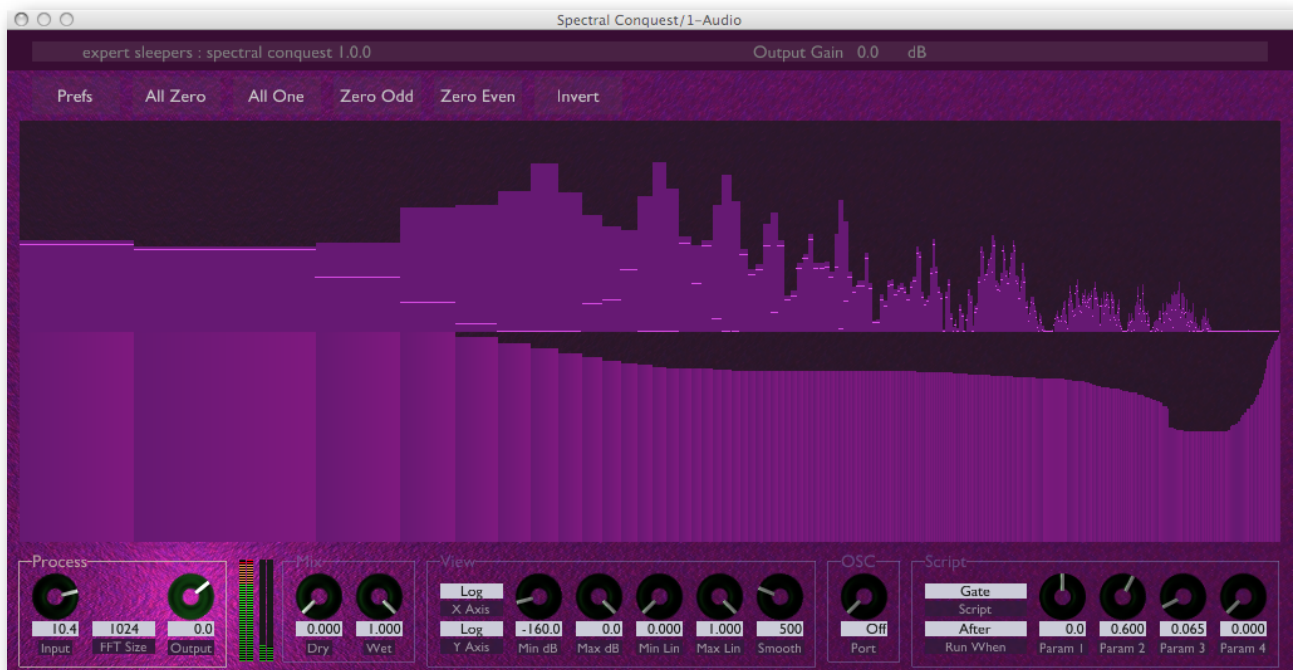
# Contents

<b>Introduction</b>	<b>6</b>
Acknowledgements	6
<b>Installation</b>	<b>7</b>
Mac OS X, Audio Unit (AU)	7
Mac OS X, VST	7
Windows (VST)	7
<b>System Requirements</b>	<b>7</b>
Mac OS X	7
Windows	7
<b>Registration</b>	<b>8</b>
<b>Using Spectral Conquest</b>	<b>9</b>
<b>Using the controls</b>	<b>9</b>
Knobs	9
Value edit boxes	9
Name/value display	9
<b>Overview</b>	<b>10</b>
<b>Spectrum Plot</b>	<b>10</b>
<b>View</b>	<b>10</b>
X/Y Axis	10
Min/Max dB	10
Min/Max Lin	10
Smooth	11
<b>Process</b>	<b>11</b>
Input	11
FFT Size	11
Output	11
<b>Mix</b>	<b>11</b>
Dry	11
Wet	11

<b>Buttons</b>	<b>11</b>
<b>Script</b>	<b>12</b>
Script	12
Run When	12
Param 1-4	12
<b>Built-in Scripts</b>	<b>12</b>
Shift	12
Reverse	12
Low Pass	12
High Pass	12
Band Pass	13
Gate	13
<b>Preferences</b>	<b>14</b>
OSC Base Port	14
Eye candy	14
Constant redraw	14
Floating tooltip	14
<b>MIDI control</b>	<b>15</b>
<b>OSC Control</b>	<b>16</b>
<b>Reference</b>	<b>16</b>
<b>MIDI &amp; OSC Scripting</b>	<b>16</b>
<b>Reference</b>	<b>16</b>
<b>Script locations</b>	<b>16</b>
<b>Spectral Scripting</b>	<b>17</b>
addSpectralScript( name, function )	17
copyBin( from, to )	17
multiplyBin( index, multiplier )	17
swapBins( index1, index2 )	18
getBin( index )	18
setBin( index, re, im )	18
getBinMagnitudeSqr( index )	18
zeroBin( index )	18
<b>Built-in Scripts</b>	<b>18</b>

<b>Version History</b>	<b>21</b>
<b>Contact</b>	<b>22</b>
<b>Acknowledgements</b>	<b>23</b>
<b>Lua</b>	<b>23</b>
<b>oscpack</b>	<b>23</b>
<b>glew</b>	<b>24</b>
<b>FreeType</b>	<b>25</b>
<b>FTGL</b>	<b>25</b>
<b>libpng</b>	<b>25</b>
<b>zlib</b>	<b>25</b>

# Introduction



Spectral Conquest is an effect plug-in that lets you directly manipulate the frequency spectrum of audio signals.

The incoming signal is analysed via a FFT (Fast Fourier Transform), resulting in the familiar spectrum plot where the full audio spectrum is split into a number of narrow frequency bands. The outgoing audio is then regenerated by an Inverse FFT. The interesting bit is that the plug-in lets you modify the frequency spectrum in between.

Spectral Conquest allows you to modify the spectrum in two ways:

- By setting a multiplier for each frequency band. You can simply draw out the multipliers in the GUI.
- By applying a script to the spectrum. Some basic scripts are included with the plug-in to perform operations such as filtering and gating, but you can also write scripts yourself to do whatever you like.

## Acknowledgements

The underlying technique used by Spectral Conquest is based on the Cycling '74 Pluggo effect 'Spectral Filter', itself based on work by Zack Settel and Cort Lippe.

# Installation

## Mac OS X, Audio Unit (AU)

The plug-in file is called `ExS1SpectralConquest.component`.

Simply copy the file to the folder:

`Library/Audio/Plug-Ins/Components`

## Mac OS X, VST

The plug-in file is called `ExS1SpectralConquest.vst`.

Simply copy the file to the folder:

`Library/Audio/Plug-Ins/VST`

## Windows (VST)

The plug-in file is called `spectralconquest.dll`.

Simply copy the file to your VST plug-ins folder.

# System Requirements

## Mac OS X

Spectral Conquest requires at least Mac OS X version 10.5.8. The plug-in is Intel-only, and built for both 32 and 64 bit. The Audio Unit version will work in any Audio Unit host. The VST version requires a “VST 2.4” compatible host.<sup>1</sup>

## Windows

Spectral Conquest has been developed and tested with Windows XP SP2. It may work with other versions of Windows (Windows 7 included) but this is by no means guaranteed. The plug-in requires a “VST 2.4” compatible host.

---

<sup>1</sup> VST is a trademark of Steinberg Media Technologies GmbH.

# Registration

The downloadable version of Spectral Conquest stops working after 15 minutes every time you use it. To stop this happening, you need to buy a registration.

You can buy a registration key online using a credit card or PayPal from the Expert Sleepers Licence Manager application. See [here](#) for more information. Note that you need at least version 1.0.16 of the Licence Manager.

The e-commerce side of things is handled by [eSellerate](#). If you have any security concerns, have a look at their website which is pretty informative.

Your registration key allows you to install Spectral Conquest on up to 3 different computers (useful if for example you have a desktop computer in the studio and a laptop for live use).

You need an internet connection to activate the software, though not necessarily on the computer on which you want to use it.



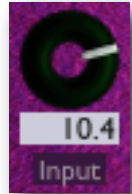
# Using Spectral Conquest

## Using the controls

### Knobs

Basic use of the knobs is to click on them and drag the mouse up and down. However you can obtain different results by holding keys as follows:

- Shift : Values change more slowly as you move the mouse.
- Command<sup>1</sup> (Mac OS X) / Alt (Windows) : The knob assumes its default position.
- Option<sup>2</sup> (Mac OS X) / Control (Windows): The knob assumes integer values only.



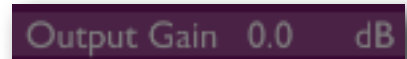
### Value edit boxes

These boxes (below each knob) let you enter parameter values directly. Clicking on the value highlights it in green - you can then type the desired value using the keyboard. Press enter to finish and accept the new value.

While you're typing the value, the box goes red to indicate that the value you see has not yet been accepted.

### Name/value display

As you move the mouse around the interface, the name and current value of the control currently under the mouse is displayed in the top right of the window. This area also provides tool-tips for buttons.



---

<sup>1</sup> The 'Command' key is also known as the 'Apple' key - the one next to the spacebar.

<sup>2</sup> The 'Option' (alt) key is the one between the Control (ctrl) key and the Command (cmd) key.

# Overview

Most of the plug-in GUI is devoted to the spectrum plot and the area that controls the frequency band or 'bin' multipliers. Below this is a row of controls, broken down into sections. The 'Process' and 'Mix' sections control the basic audio processing. The 'View' section controls the appearance and behaviour of the spectrum plot. The 'Script' section controls the scripted functions. Above the spectrum plot is a row of buttons most of which quickly set the bin levels to preset values.

## Spectrum Plot

The upper half of the large graphical area in the middle of GUI shows the spectrum plot of the incoming and outgoing audio. The solid rectangles show the incoming spectrum; the small horizontal lines show the outgoing spectrum.

The lower half of this area shows the bin levels. It essentially consists of a very large number of vertical sliders, one per bin in the spectrum. You can use each slider individually, or you can draw large numbers of levels at once simply by 'painting' with the mouse.

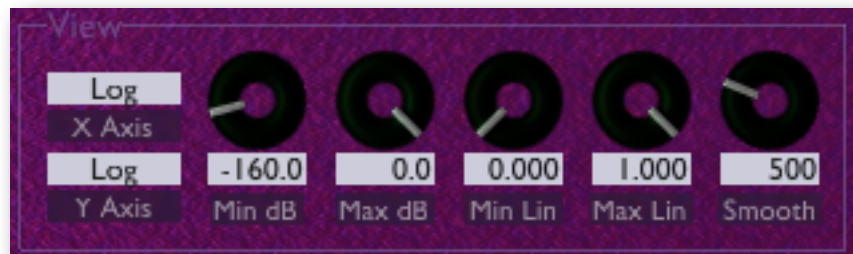
You can restrict the mouse to adjust one level only, regardless of horizontal movement, by holding Option<sup>1</sup> (Mac OS X)/Control (Windows).

## View

The View controls affect the appearance and behaviour of the spectrum plot.

### X/Y Axis

These two drop-down menus set whether the X and Y axes of the spectrum plot use Logarithmic or Linear scales.



### Min/Max dB

When the Y axis is showing a logarithmic scale, these knobs set the range of values (in dB) displayed.

### Min/Max Lin

When the Y axis is showing a linear scale, these knobs set the range of values displayed.

---

<sup>1</sup> The 'Option' (alt) key is the one between the Control (ctrl) key and the Command (cmd) key.

## Smooth

The 'Smooth' knob controls the filtering of the spectrum display. Note that this affects the visual appearance of the display only - it does not affect the audio processing in any way.

## Process

The 'Process' section controls some fundamental aspects of the audio processing.

### Input

The 'Input' knob controls the gain applied to the signal on the way in to the plug-in.



### FFT Size

This drop-down menu controls the 'size' of the FFT algorithm applied, which relates to the number of bins that the frequency spectrum is divided into.

### Output

The 'Output' knob controls the gain applied to the signal on the way out of the plug-in.

## Mix

### Dry

Sets the level of the 'dry' (i.e. unprocessed) signal leaving the plug-in.

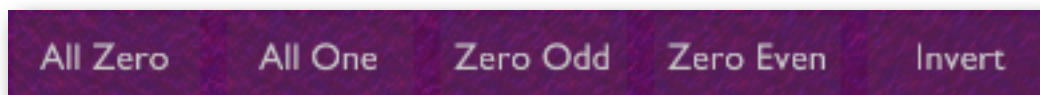
### Wet

Sets the level of the 'wet' (i.e. processed) signal leaving the plug-in.



## Buttons

The five buttons above the spectrum display change the bin levels in fairly self-explanatory ways.

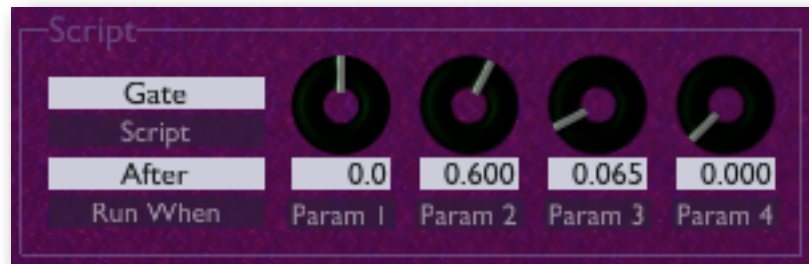


# Script

The 'Script' section offers control over the scripted functions.

## Script

This drop-down menu lets you select a script to use, or 'Off' for no script.



## Run When

This drop-down menu lets you choose when the script is run - before or after the bin multipliers have been applied.

## Param 1-4

These knobs set the parameter values that are passed to the script. Their exact meaning depends on the script chosen.

# Built-in Scripts

The following scripts are defined by the plug-in itself. New scripts may be added by the user - see [below](#).

## Shift

Shifts the values of bins up or down by a whole number of bins.

Param 1 sets the number of bins to shift by.

## Reverse

Flips the bin levels horizontally i.e. low frequencies become high frequencies and *vice versa*. Since this often results in unusably high frequencies, a bin shift is also applied.

Param 1 sets the number of bins to shift by.

## Low Pass

A 'brick wall' low pass filter. All bins above a certain point are set to zero.

Param 2 sets the filter frequency.

## High Pass

A 'brick wall' high pass filter. All bins below a certain point are set to zero.

Param 2 sets the filter frequency.

### **Band Pass**

A band pass filter. Bin levels are faded off around a centre frequency.

Param 2 sets the filter centre frequency.

Param 3 sets the falloff rate.

### **Gate**

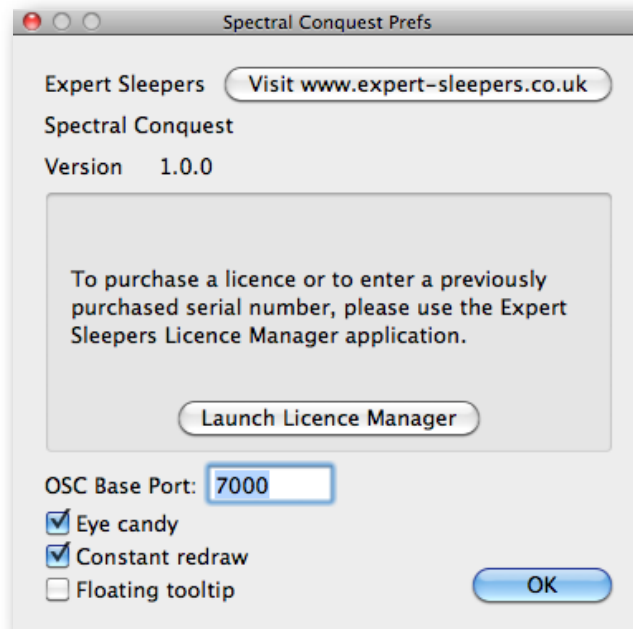
A gating effect. Depending on a mode setting, each bin level is set to zero if it is either below or above a certain threshold level.

Param 2 sets the threshold level.

Param 3 sets the mode. If it is zero, a bin is set to zero unless its level is above the threshold; if it is non-zero, a bin is set to zero unless its level is below the threshold.

# Preferences

Pressing the 'Prefs' button brings up a dialog where various preferences are set. These settings are shared by all instances of Spectral Conquest, and are not stored with presets.



The top section shows the product version.

The central section will show your serial number once you've bought a registration.

## OSC Base Port

Sets the base port number for OSC. See the section on OSC, [below](#).

## Eye candy

Enables the pretty graphics. Turn off if you don't like them, or if your computer has compatibility issues with drawing such things.

## Constant redraw

Is on by default. If turned off, the GUI is only redrawn when a control changes. Use this if you're concerned that the GUI is wasting your CPU resources.

## Floating tooltip

Causes the parameter name and value display (usually in the top right of the GUI) to be displayed above the mouse pointer.

# MIDI control

All of Spectral Conquest's parameters can be controlled via MIDI CC's (Continuous Controllers) according to the table below.

0	OSC Port Offset
2	Dry Level
3	Wet Level
4	FFT Size
5	View X Axis
8	View Y Axis
9	View Min dB
11	View Max dB
12	View Min Linear
13	View Max Linear
14	View Smoothing
15	Input Gain
16	Output Gain
17	Script Index
18	Script Position
19	Scripted 1
20	Scripted 2
21	Scripted 3
22	Scripted 4

# OSC Control

Spectral Conquest can be controlled via the Open Sound Control (OSC) protocol.

If you're new to OSC, start by visiting [opensoundcontrol.org](https://opensoundcontrol.org).

Two settings control what port the plug-in uses to listen on for OSC commands. One is the base OSC port, set in the [preferences](#). The second is the OSC Port Offset control. If the port offset is set to something other than 'Off', then the two numbers are added together and the result used as the port number. E.g. if the base port is 6000 and the port offset is 1, then the plug-in will listen on port 6001.



## Reference

All Expert Sleepers plug-ins that support OSC share a common implementation. This is documented in the 'OSC Control Manual', available from the Expert Sleepers website.

# MIDI & OSC Scripting

It is possible to extend the plug-in's MIDI & OSC functionality via user-writable scripts. Indeed, the standard MIDI functionality described above has been re-implemented using such a script, which you can use as reference for your customisations.

The language used for the MIDI scripts is Lua. You will find a complete description of the language, and some useful tutorials, at the Lua website: [www.lua.org](http://www.lua.org)

All the standard language features of Lua are available in the scripts, plus some extra functions specific to the Expert Sleepers system.

## Reference

All Expert Sleepers plug-ins that support MIDI/OSC scripting share a common implementation. This is documented in a the 'MIDI & OSC Scripting Manual', available from the Expert Sleepers website.

## Script locations

The plug-in name for constructing the script locations is 'Spectral Conquest'.



# Spectral Scripting

The scripts that are available via the 'Script' controls as described above are defined in exactly the same way as the MIDI/OSC scripts. Please review the 'MIDI & OSC Scripting Manual' for details.

In addition to the functions defined for MIDI control, Spectral Conquest defines the following functions for writing scripts to process the spectrum.

## **addSpectralScript( name, function )**

This is the main function that lets you add a new script to the range available. Simply call it in the main body of your Lua file.

E.g.

```
local function scriptedShift( state )  
    -- etc.  
end  
addSpectralScript( "Shift", scriptedShift )
```

The first argument is the name of the script to display in the plug-in GUI. The second is the function to process the spectrum.

The function passed to addSpectralScript() should accept one argument, which is a state table, used to pass various information from the plug-in to the script. The members of the state table are:

state.maxBin - the maximum bin index i.e. the bins are numbered from 0 to state.maxBin.  
state.param1/2/3/4 - the four parameter values as set by the GUI.

A complete function might look like this (which is the plug-in's built-in Low Pass Filter script):

```
local function scriptedBrickWallLPF( state )  
    local maxBin = state.maxBin  
    local cutoff = state.param2 * maxBin  
    local setBin = setBin  
    for i=cutoff,maxBin,1 do  
        setBin( i, 0, 0 )  
    end  
end
```

## **copyBin( from, to )**

Copies the value of bin index 'from' to bin index 'to'.

## **multiplyBin( index, multiplier )**

Multiplies the value of bin 'index' by 'multiplier'.

### **swapBins( index1, index2 )**

Swaps the values of bin 'index1' with bin 'index2'.

### **getBin( index )**

Returns the value of bin 'index'. The bin values are complex numbers, represented by a pair of values. E.g.

```
local re, im = getBin( i )
```

### **setBin( index, re, im )**

Sets the value of bin 'index' to the complex number represented by the pair of values 're' and 'im'.

### **getBinMagnitudeSqr( index )**

Returns the square of the magnitude of bin 'index'.

### **zeroBin( index )**

Sets the value of bin 'index' to zero.

## **Built-in Scripts**

The full text of the plug-in's built-in scripts is included below for reference.

```
-- define spectral scripts

local function scriptedShift( state )
  local shift = state.param1
  local maxBin = state.maxBin
  local copyBin = copyBin
  local zeroBin = zeroBin
  if shift > 0 then
    for i=maxBin,shift,-1 do
      copyBin( i-shift, i )
    end
    for i=shift-1,0,-1 do
      zeroBin( i )
    end
  elseif shift < 0 then
    for i=0,maxBin+shift,1 do
      copyBin( i-shift, i )
    end
    for i=maxBin+shift+1,maxBin,1 do
      zeroBin( i )
    end
  end
end

local function scriptedReverse( state )
  local maxBin = state.maxBin
```

```

local swapBins = swapBins
for i=0,(maxBin-1)/2,1 do
    swapBins( i, maxBin-i )
end
local shift = state.param1
state.param1 = shift - (maxBin-1)*3/4
scriptedShift( state )
state.param1 = shift
end

local function scriptedBrickWallLPF( state )
    local maxBin = state.maxBin
    local cutoff = state.param2 * maxBin
    local setBin = setBin
    for i=cutoff,maxBin,1 do
        setBin( i, 0, 0 )
    end
end

local function scriptedBrickWallHPF( state )
    local maxBin = state.maxBin
    local cutoff = state.param2 * maxBin
    local setBin = setBin
    for i=0,cutoff,1 do
        setBin( i, 0, 0 )
    end
end

local function scriptedBPF( state )
    local maxBin = state.maxBin
    local cutoff = state.param2 * maxBin
    local depth = state.param3 ^ 2
    if depth <= 0 then return end
    local multiplyBin = multiplyBin
    for i=0,cutoff,1 do
        local factor = 1 - ( cutoff - i ) * depth
        if factor < 0 then factor = 0 end
        multiplyBin( i, factor )
        -- equivalently
        -- local re, im = getBin( i )
        -- setBin( i, re * factor, im * factor )
    end
    for i=cutoff,maxBin,1 do
        local factor = 1 - ( i - cutoff ) * depth
        if factor < 0 then factor = 0 end
        multiplyBin( i, factor )
        -- equivalently
        -- local re, im = getBin( i )
        -- setBin( i, re * factor, im * factor )
    end
end

local function scriptedGate( state )
    local maxBin = state.maxBin
    local threshold = state.param2
    local magnitude = 0
    local getBinMagnitudeSqr = getBinMagnitudeSqr
    local zeroBin = zeroBin
    threshold = threshold * (maxBin-1) / 32

```

```

threshold = threshold ^ 2
if state.param3 > 0 then
  for i=0,maxBin,1 do
    magnitude = getBinMagnitudeSqr( i )
    if magnitude > threshold then
      zeroBin( i )
    end
  end
else
  for i=0,maxBin,1 do
    magnitude = getBinMagnitudeSqr( i )
    if magnitude < threshold then
      zeroBin( i )
    end
  end
end
end
end

-- add the scripts

addSpectralScript( "Shift", scriptedShift )
addSpectralScript( "Reverse", scriptedReverse )
addSpectralScript( "Low Pass", scriptedBrickWallLPF )
addSpectralScript( "High Pass", scriptedBrickWallHPF )
addSpectralScript( "Band Pass", scriptedBPF )
addSpectralScript( "Gate", scriptedGate )

```

# Version History

**1.0.0 1/7/2010**

- First release.

# Contact

The Expert Sleepers website is here:

<http://www.expert-sleepers.co.uk/>

Or you can email

[info@expertsleepers.co.uk](mailto:info@expertsleepers.co.uk)

Or you can use the forum, which is here:

<http://www.kvraudio.com/forum/viewforum.php?f=85>

# Acknowledgements

The software described in this manual makes use of the following open source projects. The author is greatly indebted to them for their efforts and generosity.

Below are reproduced the various copyright notices and disclaimers that accompany these software projects, in accordance with their terms of use.

## Lua



Copyright (C) 1994-2008 Lua.org, PUC-Rio.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## oscpack

oscpack -- Open Sound Control packet manipulation library  
<http://www.audiomulch.com/~rossb/code/oscpack>

Copyright (c) 2004 Ross Bencina <[rossb@audiomulch.com](mailto:rossb@audiomulch.com)>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Any person wishing to distribute modifications to the Software is requested to send the modifications to the original developer so that they can be incorporated into the canonical version.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# glew

The OpenGL Extension Wrangler Library

Copyright (C) 2002-2007, Milan Ikits <milan.ikits@ieee.org>

Copyright (C) 2002-2007, Marcelo E. Magallon <mmagallo@debian.org>

Copyright (C) 2002, Lev Povalahev

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* The name of the author may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Mesa 3-D graphics library

Version: 7.0

Copyright (C) 1999-2007 Brian Paul All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL BRIAN PAUL BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2007 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and/or associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR



PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

## FreeType

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg.

THE FREETYPE PROJECT IS PROVIDED `AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

## FTGL

Copyright (C) 2001-3 Henry Maddocks

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## libpng

<http://www.libpng.org/pub/png/libpng.html>

## zlib

<http://www.zlib.net/>