

# *Spout*



[spout.zeal.co](http://spout.zeal.co)

Version 2.006

Update 2

## User Manual

November 2018

# Table of Contents

<b>TABLE OF CONTENTS.....</b>	<b>2</b>
<b>1. INTRODUCTION.....</b>	<b>5</b>
1.1 What is Spout ?.....	5
1.2 Contact.....	5
1.3 Changes for 2.006.....	6
<b>2. OVERVIEW.....</b>	<b>7</b>
<b>3. SETUP.....</b>	<b>9</b>
3.1 Installation.....	9
3.2 SpoutSettings.....	10
3.3 SpoutCam .....	13
3.4 SpoutCamSettings.....	13
3.5 SpoutPanel.....	14
3.6 Un-installation.....	15
3.7 Re-installation.....	15
<b>4. GETTING STARTED.....</b>	<b>16</b>
4.1 SpoutSender.....	16
Sender name entry.....	16
4.2 SpoutReceiver.....	16
Receiver fit.....	17
Hide status.....	17
Full screen.....	17
4.3 Common.....	18
Show on top.....	18
System capability.....	18
Graphics.....	19
DirectX .....	19
4.4 Problem report.....	19
<b>5. PROGRAMS.....</b>	<b>20</b>
5.1 FreeFrameGL .....	20
5.2 MilkDrop .....	20
5.3 Virtual DJ .....	21
5.4 VVVV.....	22

<b>6.CONTRIBUTED PACKAGES.....</b>	<b>23</b>
6.1 Processing .....	23
6.2 Max/Msp .....	24
6.3 VVVV .....	25
6.4 GeeXLab .....	26
6.5 VIZZable .....	26
<b>7. SPOUT SDK .....</b>	<b>28</b>
<b>8. APPLICATION SUPPORT.....</b>	<b>29</b>
8.1 TouchDesigner .....	29
8.2 Magic .....	29
8.3 Resolume.....	29
8.4 Mapio .....	30
8.5 ModuloPi .....	30
8.6 Xnth .....	30
8.7 Beautiful Chaos .....	30
8.8 FacadeSignage .....	31
8.9 Black Spout .....	31
8.10 Painting with light .....	31
8.11 Fugio .....	31
8.12 VPT .....	32
8.13 Vdome .....	32
8.14 Radome .....	32
8.15 MeshWarpServer .....	32
8.16 VirtualMapper .....	32
8.17 Reymenta mix-n-map .....	32
8.18 Visikord .....	33
8.19 Heavy M .....	33
8.20 7thSense .....	33
8.21 NI Mate .....	33
8.22 Synthclipse .....	33
8.23 GeoPix .....	34
8.24 Isotonik Studios .....	34
8.25 Enttec .....	34
8.26 LightJams .....	34
8.27 S-Mode .....	34
8.28 Simmetri .....	35

8.29 Vioso .....	35
8.30 Notch .....	35
8.31 Zwobot .....	35
8.32 Arkaos .....	35
8.33 KodeLife .....	36
<b>9. DEVELOPER SUPPORT.....</b>	<b>37</b>
9.1 Unity .....	37
9.2 Adobe AIR .....	37
9.3 After Effects .....	37
9.4 TouchDesigner .....	37
9.5 Unreal Engine 4 .....	38
9.6 LightJams .....	38
9.7 Monogame .....	38
9.8 Eclipse .....	39
9.9 Openframeworks .....	39
9.10 Cinder .....	39
9.11 Videodromm .....	39
9.12 Processing .....	39
9.13 FFGLmaker.....	40
9.14 OmniMap .....	40
9.15 GitHub repositories.....	41
<b>10. ERROR REPORTING.....</b>	<b>42</b>
<b>11. CREDITS AND LICENCING.....</b>	<b>43</b>
11.1 Credits.....	43
11.2 Licencing.....	44

# 1. Introduction

## 1.1 What is *Spout* ?

*Spout* allows applications for Microsoft Windows to share video between each other in a similar way to [Syphon](#) for the Mac.

Senders and receivers include [FreeFrameGL](#) plugins, a Library for [Processing](#), [Jitter](#) externals for [Max/Msp](#), [VIZZable](#) modules for [Ableton Live](#), and a Virtual Webcam as a universal receiver.

For compatible graphics hardware, OpenGL textures are shared by way of DirectX using the NVIDIA [DirectX/Opengl interop](#) extension. If hardware is not compatible, *Spout* provides backup by way of shared memory.

*Spout 2* is an extensive revision of the initial release of *Spout*.

Features include :

- Compatible with Spout1 DirectX 9 apps.
- Support for DirectX 11 textures.
- Developer SDK and example code.
- Simplified BSD licencing for use in commercial applications.
- Built in support for major applications.

The *Spout* SDK includes example [openFrameworks](#) and [Cinder](#) code, a dll project. Get more details on GitHub.

<https://github.com/leadedge/Spout2>

If you wish to incorporate *Spout* in your application or assist with development or testing, you are welcome to contact us on the Spout website developer forum.

## 1.2 Contact

Contact us by email [spout@zeal.co](mailto:spout@zeal.co) or on the website forum <http://spout.zeal.co/>.

## 1.3 Changes for 2.006

Fundamentally, textures are shared using DirectX. But, if the graphics driver does not provide the required extension by NVIDIA ([NV\\_DX\\_interop](#)), there is no way of linking the shared DirectX textures with OpenGL.

Previous versions of the Spout SDK have used shared memory as a backup instead of using shared DirectX textures. Version 2.006 introduces an additional backup method using CPU processing to access the shared DirectX texture and copy the data from memory to an OpenGL texture.

This way, the shared DirectX textures still exist and the share handles are always valid. Applications that access the shared DirectX textures directly will now share with OpenGL applications even if the GL/DX interop extension is not available.

Efficiency is optimised for memory copy and it is fast enough for practical use at commonly used resolutions.

Sharing mode can be selected using the "SpoutSettings" utility.

More information in the documentation for Spout SDK.

### Updates since initial 2.006 release

- 16.04.17 - Rebuild Spout Freeframe dlls VS2012 with original IDs for Isadora
- 31.10.17 - SpoutReceiver2.dll
  - close receiver on receivetexture fail
  - <https://github.com/leadedge/Spout2/issues/25>
  - Version 3.031
- 18.11.17 - Spout demo receiver - fullscreen for multiple monitors thanks to Tim Thompson - Version 2.020
  - Change Visual Studio redistributable installation to 32bit only
  - <https://github.com/leadedge/Spout2/issues/31>
- 01.03.18 - SpoutPanel : read MaxSenders registry entry
  - <https://github.com/leadedge/Spout2/issues/33>
  - Version 2.21
- 02.03.18 - Add MaxSenders registry entry
  - Replace SpoutDXmode with SpoutSettings to allow setting maximum senders.
  - Update demo programs : increment the sender name if the same one is started – Sender version 2.020
- 12.11.18 - Rebuild Milkdrop plugin to fix DirectX device release memory leak
  - Update Spout SDK source to align with GitHub

## 2. Overview

The Spout protocol for sharing texture between Microsoft Windows applications has been widely adopted.

### FreeframeGL

Any application with FreeframeGL support can use the "SpoutReceiver2" and "SpoutSender2" plugins. They have been shown to work well with [Resolume Avenue and Arena](#), [Isadora](#), and [Magic](#).

Although Resolume and Magic now support Spout directly, the Freeframe plugins are required for Isadora.

### Max/Msp Jitter

The Spout Jitter externals *jit.gl.spoutsender* and *jit.gl.spoutreceiver* have a similar function to the *Syphon* externals for the Mac and can be used in their place with only slight modifications. The Max package manager now gives direct access to these externals from within a patch.

### Processing

[Processing](#) is supported by a contributed library "Spout for Processing". Example sketches are provided for sending and receiving graphics or images.

### SpoutCam

32 bit applications that support a DirectShow source and also support virtual webcams, can receive the output from Spout senders through "SpoutCam". After installation, "SpoutCam" shows up in the list of webcams for the host program.

### Virtual DJ

Plugins for Virtual DJ 8 allow sending and receiving Spout textures from any other Spout enabled application.

## Milkdrop

*vis\_milk2.dll* is a 32bit visualization plugin for the classic music visualizer [Winamp](#) and has been modified for Spout output. After it is copied to the Winamp plugins folder, and the Milkdrop visualizer selected, a Spout receiver will pick up the Winamp output (see more below).

## Application support

An increasing number of commercial and open-source applications are including the Spout frame sharing protocol. As many as possible are included in this manual.

## Spout SDK

A comprehensive SDK is provided for experienced programmers who would like to incorporate frame sharing into their own applications.

The source files can be used directly in the application or the SDK can be built as a dll. A separate dll project "*SpoutLibrary*" allows compatibility with compilers other than Visual Studio that require C-compatible function exports.

Check the latest copy on GitHub : <https://github.com/leadedge/Spout2>



## 3. Setup

### 3.1 Installation

During installation you will have had the option of installing the virtual camera "*SpoutCam*". Virtual cameras are not compatible with many programs and may interfere with their function. If you subsequently find *SpoutCam* to be a problem, *Spout* can be removed and re-installed without it .

At the end of the installation, you may be prompted to install the Visual Studio 2012 runtime if it is not already installed.

If you prefer not to install the runtime immediately, you can install it later by other means or run the installation again. Be patient because it can take a minute or two. The runtime installation requires a restart but you have the option to restart later.

If you find that Spout programs still report missing dll's, you may need to install the Visual Studio 2012 runtime manually. [Download from here](#).

After installation you will find a *Spout* icon on your desktop :



Open it and you will find the files installed in a "Spout" folder in your Program Files folder with sub-folders :

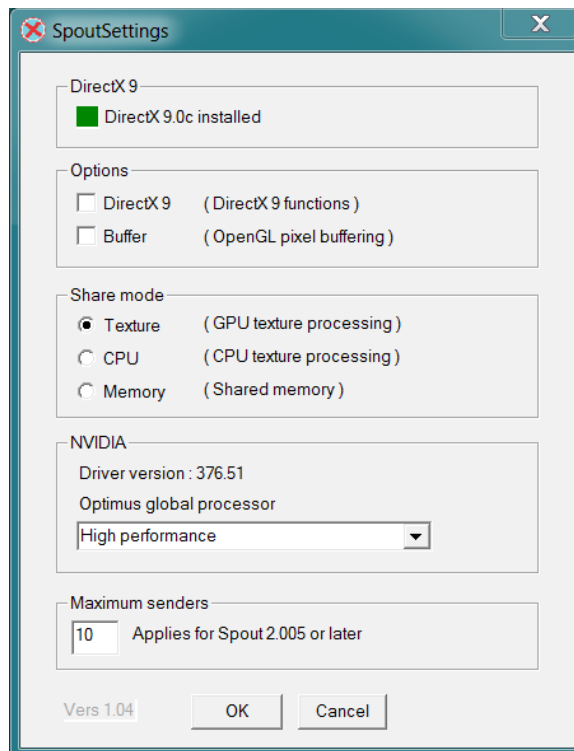
- DEMO
- PROGRAMS
- SPOUTCAM
- SPOUTSETTINGS
- SPOUTPANEL
- SPOUTSDK

## 3.2 SpoutSettings

After installation you will find another icon on the desktop which allows you to configure Spout for your hardware.



Double click to open the program.



### DirectX 9

If DirectX 9 is not available on your system download buttons will be available for the installers from Microsoft. These links are :

[Web installer](#) (easiest) or [Download](#)

## **Options**

### DirectX 9

DirectX 11 is the preferred mode of operation for Spout. But some graphics adapters do not support the DirectX 11 texture sharing functions.

If the "DirectX 9" option is checked, DirectX 9 functions will be subsequently used by all Spout plugins and applications.

This can be confirmed by the Spout Demo programs by observing the the window name. For example the sender will be either "Spout DX11 Sender" or "Spout DX9 Sender".

### Buffer

This activates OpenGL pixel buffering to copy data between GPU and CPU. It will be effective for applications which use images in CPU memory and for CPU texture sharing mode. The function could possibly introduce latency. Check for your application.

## **Share mode**

### Texture

Texture sharing is the default where hardware supports the NVIDIA OpenGL/DirectX interop extension to OpenGL.

### CPU

CPU data transfer from DirectX shared textures is a backup where hardware is not compatible and is not recommended if texture sharing is available.

### Memory

CPU shared memory instead of DirectX shared textures. The option is retained for backwards compatibility with Spout 2.005 and may be removed for future releases.

## **NVIDIA global processor**

This option is intended for laptop computers with dual graphics and NVIDIA "[Optimus](#)" power saving technology, where the system can optimise power consumption vs performance by automatically selecting the appropriate graphics adapter for each application.

It will not have any effect if the computer does not have Optimus graphics or is not NVIDIA.

If your computer uses Optimus graphics, Spout applications may require the NVIDIA processor for texture sharing to be available.

The NVIDIA control panel is preferred way of setting the processor globally for all applications, but SpoutDXmode can be useful if there is difficulty. If it has effect, it will over-ride the NVIDIA control panel global settings.

*Note that not all Optimus systems may respond to the high performance setting in SpoutDXmode.*

## **Optimus profiles**

For Optimus laptops, it is often preferable to set up a profile for each application in order to conserve battery power when they are not in use.

If an application already has a profile, each option in the profile will take account of whether the NVIDIA global setting has been selected, so you have to change the application profile itself.

## **Maximum Senders**

Sets the maximum number of simultaneous Spout senders. For this to apply, all running Spout applications must have been built with the 2.005 SDK or later. Applications built with 2.004 or earlier will be limited to the default of 10 senders.

### 3.3 SpoutCam

*SpoutCam* is a 32 bit [DirectShow](#) filter that simulates a webcam and shows the output of a *Spout* sender.

The camera must have been selected as an option during the *Spout* installation. All the necessary registration of the filter is done during the installation. No further action is necessary.

*The Spout installation will register SpoutCam for 64 bit operating systems, but it will only work with 32 bit host programs.*

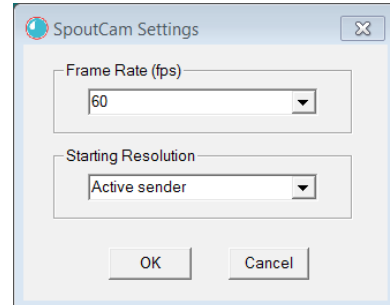
*SpoutCam* will show up in any 32 bit program that accepts a webcam and supports DirectShow sources. Select "*SpoutCam*" from the list of webcams.

Not all webcam hosts support virtual cameras. If you have trouble, un-install *Spout* and re-install without *SpoutCam* selected.

### 3.4 SpoutCamSettings

By default, *SpoutCam* will use the resolution of the active sender running when it starts - as long as that sender is started before *SpoutCam*.

However, some host programs do not accept arbitrary resolutions, so the desired resolution can be set using *SpoutCamSettings*.



Defaults are those used by previous versions of *SpoutCam*, i.e. 60fps and connect to the active sender.

#### Frame Rate

"30 fps" is typical for a webcam and will be expected by most host programs.

You will get better performance by selecting "60 fps" if the host program accepts this rate and if the CPU performance is sufficient for the processing required.

The frame rate selected is the "desired" frame rate. *SpoutCam* will attempt to meet that frame rate and drop frames if it cannot keep up.

### Starting Resolution

The choices are typical of resolutions that host programs expect from a webcam.

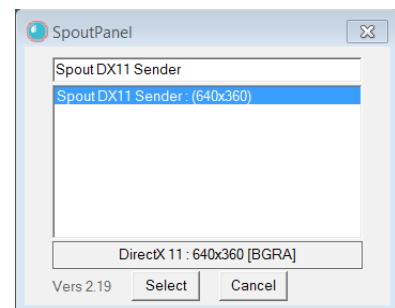
If "Active sender" is selected *and* a sender is running when *SpoutCam* starts, the resolution of that sender will be used. This will give optimum resolution and performance.

If no sender is running when *SpoutCam* starts, and Active Sender has been selected, the default resolution of 640x480 is used and will be retained regardless of the resolution of the sender. Restart *SpoutCam* to adapt to the sender resolution.

If any other resolution is selected, that resolution is used and will be retained. Received frames are resized to that resolution.

## 3.5 SpoutPanel

Applications with Spout support built in may have their own sender selection method, but many Spout receiver applications may use an executable program "*SpoutPanel.exe*" to detect and select senders.

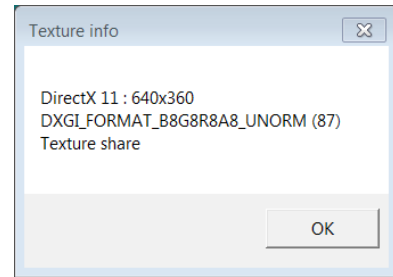


The Spout installer sets up registry entries so that any receiver application built with the Spout SDK will find *SpoutPanel*.

However, if Spout files have been copied from GitHub rather than installed, *SpoutPanel.exe* has to be copied to the path of the executable program for each receiver.

*SpoutPanel* shows the name, texture size and format of Senders.

To see more detailed information about the sender texture format, click on the texture information display at the bottom.



For Spout 2.006, sharing mode is also shown : Texture share, CPU texture share or Memoryshare.

### 3.6 Un-installation

Open the *Spout* distribution folder and run "unins000.exe", or by using the usual Windows control panel. Un-installation will remove all traces including registry entries.

If there is a warning that *SpoutCam* is in use and you can't find the host program that is using it, the easiest thing to do is cancel, restart and un-install again.

### 3.7 Re-installation

The installer will prompt for removal of an existing installation. If there is no need to retain the old one, click YES to this and it will be removed and replaced.

## 4. Getting Started

The *DEMO* folder contains example programs for a *Spout sender* and a *Spout receiver*. They can be run directly from the installation folder, but can be moved elsewhere.

### 4.1 SpoutSender

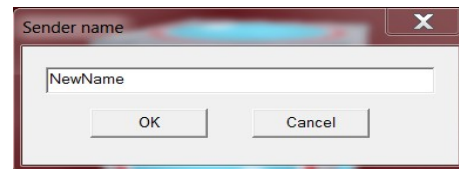
When you run *SpoutSender.exe* it is registered as a *Spout* sender and any Spout receiver can pick it up.

Each application will have it's own user interface for selecting senders. For example, with [Resolume](#), you will see them listed in the Sources tab under "Spout". Just drag a sender to a cell and click activate it. See more for individual applications below.

*NOTE: If your hardware is not compatible with texture sharing, the sender will start in memoryshare mode and indicate this at the top left of the window.*

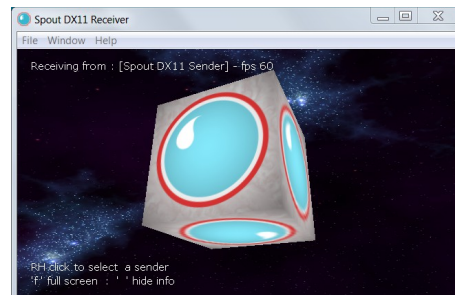
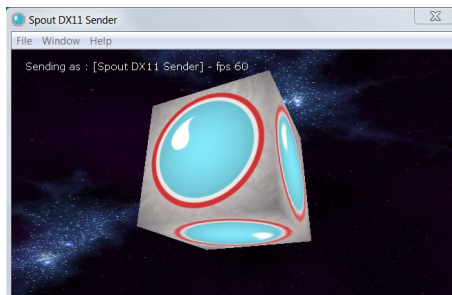
#### Sender name entry

To change the default name for the Sender, select "File > Sender Name" and enter the new name required.



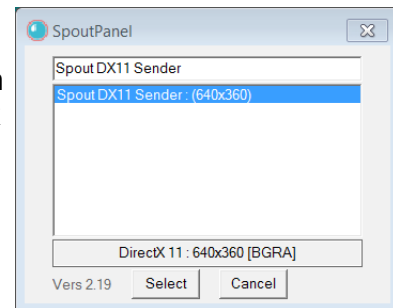
### 4.2 SpoutReceiver

If you run "SpoutReceiver" immediately after "SpoutSender" it will pick up this sender because it is the first and only sender running.



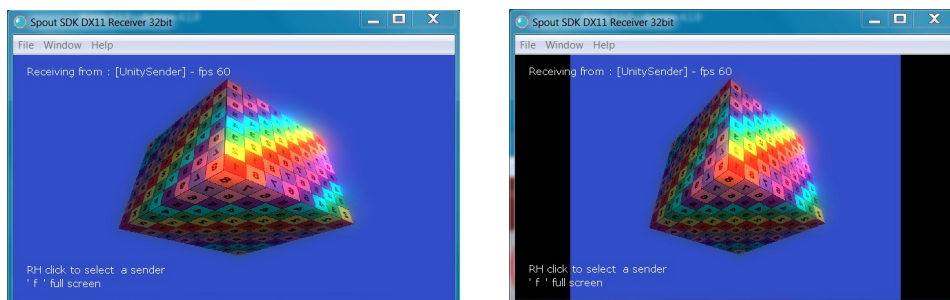


If there are more senders running you can change to any one of them with a RH click over the Receiver window or by selecting *File->Select Sender*. This brings up a dialog that allows you to select a sender.



## Receiver fit

The Receiver can be resized and can also be set so that the received image is fitted within the window or retain the original aspect ratio of the sender.



## Hide status

Press "space" to hide the status information. Useful for screen captures or recording.

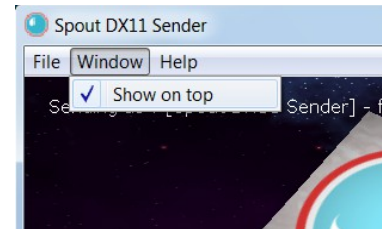
## Full screen

The receiver can display full screen by pressing "f" at any time or by selecting "Window > Full Screen". The cursor is removed while in full screen mode, but a sender can still be selected by clicking the RH mouse button. To quit full screen mode press the "ESCAPE" key.

## 4.3 Common

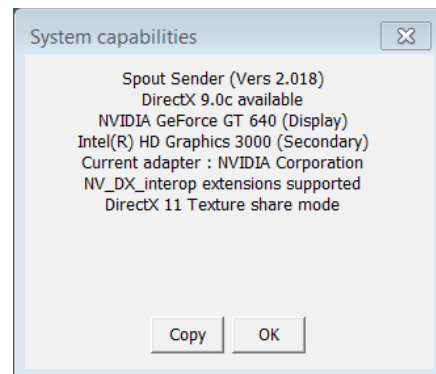
### Show on top

Both the Sender and Receiver windows can be set to be topmost so they are not obscured by other windows.



### System capability

The Spout demo programs can also be used to detect system information and whether your hardware supports texture sharing. From the Help menu, select "System diagnostics". The dialog will show whether your hardware has the capabilities required. If your system is capable of texture sharing you will see the detail like that shown at right.



If the required OpenGL extensions are not available you will see :

*"NV\_DX\_interop extensions not supported."*

There are some graphics drivers that support the required OpenGL extensions, but are not available for use. In this case you will see :

*"NV\_DX\_interop extensions supported."*

Following that, the DirectX and texture sharing modes are shown.

*DirectX 11 or DirectX*

*Texture share mode*

*or*

*CPU texture share mode*

*or*

*Memory share mode*

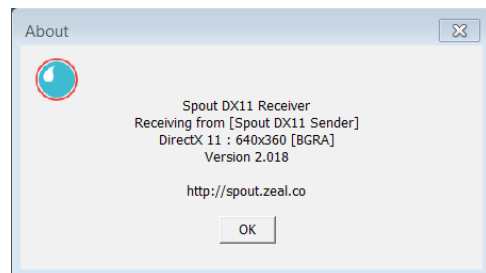
## Graphics

Two graphics adapters will be detected for dual graphics systems. The important one to look for is the primary adapter as NVIDIA.

## DirectX

Spout senders and receivers will operate using either DirectX 9 or DirectX 11 functions depending on whether the "DirectX 9" option has been selected using the utility program "SpoutDXmode".

A receiver can show details of the sender it is connected to. From the Help menu, select "About". The dialog will show whether the sender is DirectX 9 or DirectX 11 and it's size and texture format.



### ***Update your graphics driver***

*Make sure that you have the latest driver available for your graphics card. If it is an NVIDIA card, get the latest [NVIDIA drivers](#). If you have an ATI card, use the latest [Catalyst](#) drivers.*

## 4.4 Problem report

To report problems, click the "Copy" button to copy the text to the clipboard and paste it into a support forum post at :

<http://spout.zeal.co/forums/forum/spout/support/>

## 5. Programs

Programs and plugins provided with the installation are described here.

### 5.1 FreeFrameGL

The Spout FreeFrame plugins operate in the same way for all host applications that support Freeframe. These include [Resolume](#), [Magic](#) and [Isadora](#).

Magic and Resolume have Spout built in. Isadora requires the use of the plugins. [Detailed instructions](#) are available on the [TroikaTronix](#) website.

### 5.2 MilkDrop

Milkdrop (*vis\_milk2.dll*) is a visualization plugin for [Winamp](#). Refer to the [website](#) for details and download of Winamp.

Copy *vis\_milk2.dll* from the Spout distribution to the Winamp plugins folder. Start Winamp and select :

Options > Visualization > Select Plugin  
and choose "MilkDrop v2.25c [*vis\_milk2.dll*]"

Spout options are available in the Visualization configuration control panel :

Options -> Visualizations -> Configure Plugin

MORE SETTINGS tab  
[ ] Enable Spout output (default ON)

Settings are saved with OK.

The Spout output can also be changed when the Visualization is running using Ctrl-Z to enable or disable. The selected settings are saved when the Visualizer is stopped.

### 5.3 Virtual DJ



"VDJSpoutReceiver.dll" and "VDJSpoutSender.dll" are 32bit plugins developed using the [VirtualDJ 8](#) plugin SDK. The plugins are only compatible with VirtualDJ 8 will not work on VirtualDJ 7 or earlier.

VDJSpoutSender.dll must be copied to the VirtualDJ "Plugins>VideoEffect" folder

VDJSpoutReceiver.dll must be copied to the VirtualDJ "Plugins>Visualisations" folder

#### **Sender**

- 1) Start any visuals you want to send out.
- 2) Select "VDJSpoutSender" from the "VideoEffect" plugin folder.
- 3) Activate the effect by clicking once on the effect button.
- 4) Start a Spout demo receiver.

The output from Virtual DJ will be seen immediately by the receiver as "VirtualDJ Spout Sender".

#### **Receiver**

- 1) Start a Spout demo sender.
- 2) Select "VDJSpoutReceiver" from the "VideoEffect" plugin folder.
- 3) Activate the effect by clicking once on the effect button.
- 4) The sender output will immediately be seen in the video window.

To select a sender :

Under the effect selection button, click the "+" button to show the effect GUI at right. Click the "Sender" button. "SpoutPanel" will pop up and allow you to choose a sender.

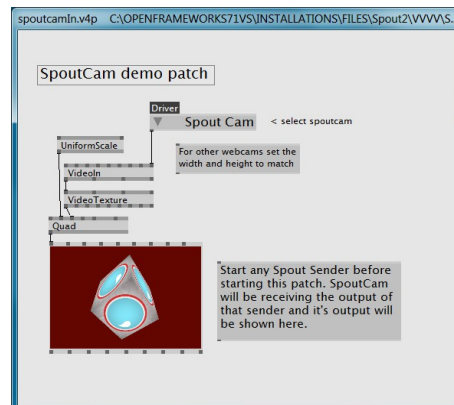
## 5.4 VVVV



### SpoutCam input

There is only one example patch provided with the installation because VVVV ships with Spout nodes. [Details below](#).

For this demonstration patch, *SpoutCam* must have been selected during *Spout* installation or it will not be present in the list of webcams.



Start any Spout sender. Then start the "*spoutcamIn*" patch. The output of the sender is picked up by *SpoutCam* and can be shown in the patch simply by selecting it as the webcam. *SpoutCam* has been pre-selected in this patch.

## 6. Contributed packages

### 6.1 Processing



[Spout for Processing](#) is available as a contributed library and can be imported from within a sketch using the Processing Contribution Manager.

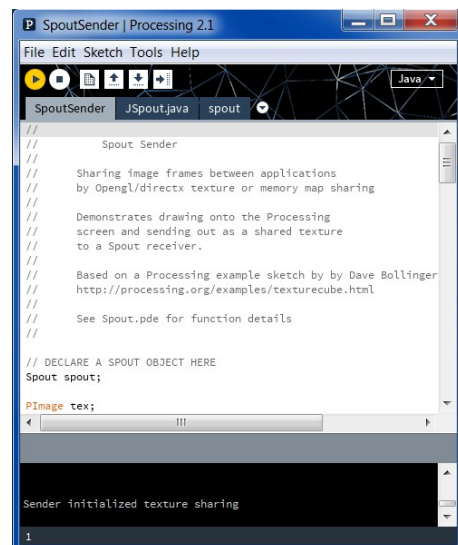
Click "Install" and after it has downloaded you will find it in your "Libraries" folder. Open that folder and you will find examples for sender/receiver and multiple sender/receiver. If you want to create your own Processing sketches, simply use the examples as a guide.

#### Sender

The example "*SpoutSender*" sketch sends frames out for pickup by any *Spout* "Receiver".

Start up the sketch and hit the "Play" button. It is now sending to any receiver.

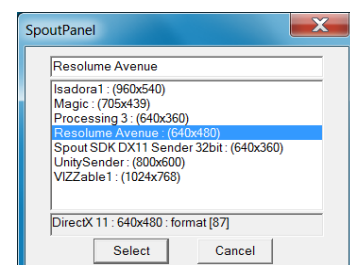
The name for the sender is set within sketch. Explore the sketch code to find out how this is done.



#### Receiver

This is an example sketch to receive the output of any *Spout* Sender. Click "Play" and when the sketch starts it will detect the "active" sender. If there is only one sender running then that will be it.

If there is more than one sender running, to select a sender, RH click inside the render window and you will see a list of all senders running. Click on the one you want and then "Select".

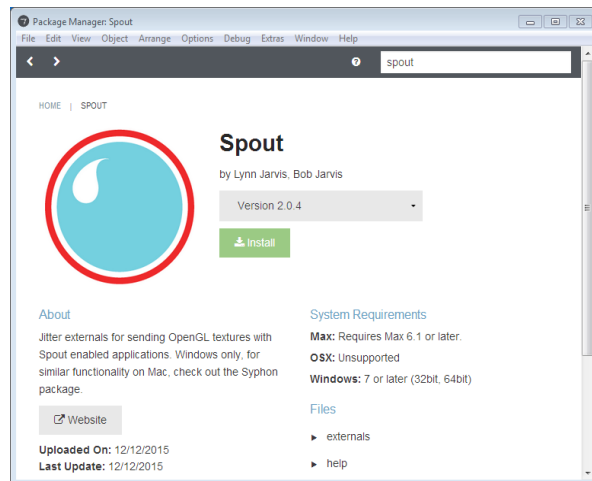


## 6.2 Max/Msp

### Jitter externals

Spout is available through the Max package manager. There are help patches to get you started.

The "*jit.gl.spoutsender*" and "*jit.gl.spoutreceiver*" externals have similar function to "*jit.gl.syphonserver*" and "*jit.gl.syphonclient*", so can be used in the same place with minor changes.



You can create your own Max patches using these externals.

To make your own sender patch using the Jitter external "*jit.gl.spoutsender.mxe*".

1) Create a render window with a context name. e.g. "spoutcontext".

```
jit.window spoutcontext @size 320 240
```

2) Connect "*jit.gl.spoutsender*" to your source. It must have the same context name..

```
jit.gl.spoutsender spoutcontext
```

3) Render to the window as usual.

Note that the render window itself cannot be created hidden. However, an attribute can be set in *jit.gl.spoutsender* to hide the render window after startup.

hidewindow                      Hide render window                      1 - activate, 0 - deactivate, default 0

for example    : *jit.gl.spoutsender* spoutcontext @hidewindow 1



To make your own receiver patch follow a similar process by examining the examples provided.

Note the the "getavailablesenders" attribute that allows you to get a list of available *Spout* senders which can be used to populate a menu.

This has a similar function to the "getavailableservers" attribute for the *Syphon* client external, except that the list only has sender names and no associated application names. See the Max *spoutReceiver* demo patch for an example of this.

## 6.3 VVVV

In latest [alphas](#) at the time of writing, VVVV ships with [two simple Spout nodes](#) which should be all you ever need:

- Spout (EX9.Texture Sender)
- Spout (EX9.Texture Receiver)

Both modules are also available as [DX11 versions](#).

There are help patches to get you started (based on the original patches created by [Elio](#)).

Spout (DX9.Texture Sender) help.v4p  
Spout (DX9.Texture Receiver) help.v4p  
Spout (DX11.Texture Sender) help.v4p  
Spout (DX11.Texture Receiver) help.v4p

At the time of writing, you need to have a Spout sender running **before** you start the VVVV sender patch or the sender will not be detected by a receiver. This can include [Resolume](#) with Spout output enabled, or the Spout demo sender for use with other applications.

Run the sender help patch and the output will be available to any Spout receiver. The example receiver patch will likewise receive from any Spout sender.

## 6.4 GeeXLab



*GeexLabSpout.dll* is a Spout sender dll for [GeeXLab](#) based on the user example in the GeeXLab code sample pack. This allows the shader output to be shared with any Windows Spout receiver application.

A demo shader as well as the plugin is in the "host\_api/Spout\_Sender/" folder in the code sample pack.

<http://www.geeks3d.com/geexlab/downloads/>

If you want to compile the dll yourself, the source is available :

<https://github.com/leadedge/GeeXLabSpout>

See more in [this](#) article.

## 6.5 VIZZable



Live

The Spout Max externals are embedded in the *spoutOut* and *spoutIn* modules of [VIZZable](#), a modular system for effects for Ableton and Max for Live.

### **VIZZable sender**

Drag and drop "*spoutOut*" after the effect you want to send out. You can then pick up the output with a receiver.

The first sender name is automatically set as "VIZZable1". If you add another effect after that, you can drop another *spoutOut* module after this and that is given the name "VIZZable2", and so on.

If you delete any instance of *spoutOut* and add another, the names keep incrementing and do not start over again, so you might delete "VIZZable2" for example, but it is not replaced.

It is best practice to drop *spoutOut* as the last module after a chain of effects.

### **VIZZable receiver**

To receive frames from a Spout sender, use the module "*spoutIn*". Use it as you would a webcam or a movie player and drag and drop it before you start a "*VIEWR*" module.

When it starts, the *spoutIn* module it will pick up the "active sender", the first Spout sender that was started or the one that was last selected.

To change the source, simply click "*Refresh*" to update the list of available senders. Select the sender you want from the drop-down list.

## 7. Spout SDK

a developer SDK



The Spout SDK is a set of C++ class files that allow programmers to embed Spout texture sharing functions into their own application.

The source files can be compiled directly into user developed applications or compiled as a dll for use with Visual Studio.

There is also a library project which can be used to produce a C-compatible dll for use with compilers other than Visual Studio.

The SDK source, library, projects, example code and demonstration programs are provided with the Spout distribution.

Example *openFrameworks* and *Cinder* applications are included. These have been successfully built using openFrameworks 0084 with Visual Studio 2012.

Look at the SDK documentation for details. Comments are also included in the code. The project assumes experience and familiarity with C++.

For the latest code, download from the Spout2 GitHub repository.

<https://github.com/leadedge/Spout2>

## 8. Application support

Many applications include Spout support built in.

### 8.1 TouchDesigner



[TouchDesigner](#) provides [Spout In](#) and [Spout Out](#) TOPS that work in much the same way as the DirectX in and Out TOPS or the Movie In TOP. Full details can be found in the [Wiki](#).

### 8.2 Magic



[Magic](#) provides Spout functions which can be used instead of the Spout FFGL plugins. Details in the [Users Guide](#).

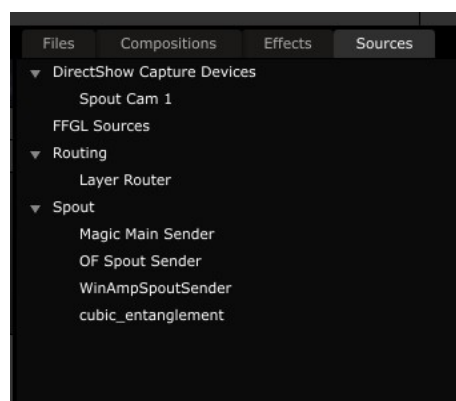
### 8.3 Resolume



Spout output can be activated in [Resolume](#) using "Output > Spout".

Spout input is always enabled and senders are shown in the "Sources" tab under "Spout".

Operation is the same as for Syphon and details are described in the [manual](#).



## 8.4 Mapio



[Mapio](#) provides a Spout output function which is very simple to use. Just select "Destination – Spout". You can then pick up the output with any Spout receiver

## 8.5 ModuloPi



[ModuloPi](#) produce "[Modulo Player](#)" which is a video media Server that provides user interface software with Spout support.

For further details contact [ModuloPi](#).

## 8.6 Xnth



[Xnth](#) software, built on gaming engine technology, generates real time video graphics, typically used by visual artists & VJ's to create screen content for concerts and clubs, festivals and live event productions.

An early release [spout support module](#) has been offered by Xnth.

The [Xnth Manual](#) and [Xnth Community Forum](#) have further information and resources relating to Xnth.

## 8.7 Beautiful Chaos



[Beautiful Chaos](#) is designed for the [Leap Motion](#) gesture controller and supports streaming video output via Syphon (OS X) and Spout (Windows).

Spout output is enabled with a control to enable video output. [Full documentation](#) of all features is available.

## 8.8 FacadeSignage



[FacadeSignage](#) is a novel Digital Signage solution that can be use a PC and one (or more) projectors to “dress” any physical object or architectural structure with digital media. Spout receiver is now in the Standard edition. For further details contact [FacadeSignage](#).

## 8.9 Black Spout



In homage to [BlackSyphon](#), Eric Newman from [Magic Music Visuals](#) has developed a little utility called *BlackSpout* which lets you capture video from any [Blackmagic](#) device and send it in real-time to any application with Spout Receiver capability.

Here is a “temporary” download page in the Magic forums. Please follow the links there for comments/feedback.

<http://magicmusicvisuals.com/forums/viewtopic.php?f=6&t=201>

## 8.10 Painting with light



[Painting With Light](#) is a video mapping tool by digital artist [Alex May](#) that enables artists of any technical ability to paint with static and moving images through any video projector onto 3D physical objects. Versions after [1.3.806](#) include Spout send and receive support.

## 8.11 Fugio



[Fugio](#) by [Alex May](#) is an open visual programming system for building digital art and creative projects quickly, with no programming experience required.

In the spirit of advancing digital art and its implementation and preservation, Fugio is open source and hosted on [GitHub](#).

### 8.12 VPT



[VPT](#) (VideoProjectionTool) is a free multipurpose realtime projection software tool for Mac and Windows created by [HC Gilje](#) which includes Spout sources.

### 8.13 Vdome



[VDOME](#) is an open source application designed to calibrate multiple projectors within a fulldome environment.

### 8.14 Radome



Radome is a platform for realtime interactive visualization on a dome. It manages both 2D and 3D content for projection onto a fulldome installation with multiple projectors. <https://github.com/Dewb/radome>

### 8.15 MeshWarpServer



MeshWarpServer

[MeshWarpServer](#) is a versatile Mapping Tool to create complex Video Projection Installations.

### 8.16 VirtualMapper



[VirtualMapper](#) is a utility tool for making projection mapping.

### 8.17 Reymenta mix-n-map



Mix-n-Map is a [Cinder](#) app to render multiple sources with Spout.



### 8.18 Visikord



[Visikord](#) is software for interactive visuals using the [Microsoft Kinect](#) that generates visual effects from gestures and movements of the guests of an event.

### 8.19 Heavy M



[HeavyM](#) is projection mapping software to create visual animations and project them on real objects.

### 8.20 7thSense



[7thSense Design](#) support Spout to integrate real-time video from external applications into Delta Media Server and Stack Content Management Frameworks. [Contact them](#) for more information.

### 8.21 NI Mate

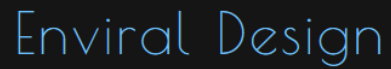


[NI mate](#) is real-time motion capture software that has been designed for anyone to use without markers, complex and expensive camera setups or long render times and includes Spout output.

### 8.22 Synthclipse Synthclipse

[Synthclipse](#) is a GLSL shader prototyping tool based on the [Eclipse IDE](#) and includes [Spout output](#).

### 8.23 GeoPix



[GeoPix by Enviral Design](#) is a LED Pixel Mapper solution based on [TouchDesigner](#).

### 8.24 Isotonik Studios



Isotonik Studios [Ned RUSH Video Collection](#) are visualizers that react to audio and include Spout output.

### 8.25 Enttec



[ENTTEC LED MAPPER](#) is a LED mapping system with Spout support.

### 8.26 LightJams



[Lightjams](#) is a software for controlling lighting in real-time and allows you to map videos, still images and live effects on your lighting fixtures as if they were part of a large matrix. The most common usage is to display content on a RGB LED wall. Use live videos generated by other apps with Spout.

### 8.27 S-Mode



[Smode Synth](#) is a real-time 2D/3D generative content and multi-layer compositing application, a real-time video synthesizer which can work with audio and video inputs, control inputs (midi keyboards and control surfaces, OSC, audio) and it's visual output is refreshed in real-time either on an external display or through Spout.

## 8.28 Simmetri

[Simmetri](#) is a creative platform for building live interactive 3D experiences for your audience - through screens, projections or virtual reality. You can output the entire UI via Spout and/or individual camera angles/experiences. Or you can create receiver textures and use them in materials that render objects/surfaces.

## 8.29 Vioso

[Vioso](#) are developers of software and hardware technologies for Video Mapping Projection and auto-calibration. Spout is integrated in their products, including Anyblend 4 and VIOSO Player

## 8.30 Notch

[Notch](#) is an all-in-one real-time graphics workhorse for production of video content, live visual effects, VR and interactive experiences. Notch supports both Spout Out and In, so it can be used to pipe in images for use in Notch scenes & effects.

## 8.31 Zwobot

[Zwobot Visuals](#) is a system for fully automated, modular and extensible visuals, VJ and movie mixer for Ableton Live. The software uses the latest Max externals for Spout 2.005, available from the Max package manager.

## 8.32 Arkaos

[Arkaos Grand VJ](#) is a comprehensive system to manipulate, trigger and mix video clips with sound, animated text strings, up to 8 layers with a vast library of video effects, transitions and sound driven visual generators. The new release brings support for Syphon and Spout.

### 8.33 KodeLife



[KodeLife](#) is a real-time GPU shader editor, live-code performance tool and graphics prototyping sketchpad with Spout output.

## 9. Developer support

Thanks to the efforts of many developers, Spout is supported by other applications by way of plugins, addons and libraries.

They share their experience and contributions for development with the Spout SDK. To find more information about these, follow the links below.

Apologies for any who have been missed. Email [spout@zeal.co](mailto:spout@zeal.co) to let us know.

### 9.1 Unity unity

Stefan Schlupek manages a GitHub repository for a [Spout plugin for Unity](#) from the work of [Benjamin Kuperberg](#) with the same BSD licence as the Spout SDK.

### 9.2 Adobe AIR

Ben Kuper has developed an [extension for Adobe AIR](#) that lets you share a BitmapData with Spout so you can use it in Processing, Resolume, Unity, etc. and has prepared a [video](#) on it's use.

### 9.3 After Effects

Ben Kuper has developed an [After Effects plugin](#) for sending textures to Spout enabled apps. A demonstration video is [here](#).

### 9.4 TouchDesigner

"[Spoutkit](#)" by Richard Burns is a development using TouchDesigner that provides a control panel with some generative visuals and Spout output. People who don't know TouchDesigner can still go ahead and use generative clips. See more on his [blog](#) and download it [here](#).

## 9.5 Unreal Engine 4



[Michael Allar](#) has created a [Spout plugin](#) for Unreal Engine 4.

His video presentation for building UE4 and the Spout plugin is in two parts. The [first part](#) describes how to build Unreal Engine 4 from source and the [second part](#) guides you through building the Spout plugin and integrating it with UE4. Hopefully these will be useful for anybody interested in doing this.

*Please note that Michael can't offer any support for this plugin.*

[AleDel](#) has created a Spout plugin for Unreal Engine 4 and provided source and installation instructions in a [GitHub repository](#).

[Harvey Buchan](#) has a fork "UE4toSpout" in a [GitHub repository](#) with full installation instructions.

## 9.6 LightJams



Mathieu Jacques , the developer of [Lightjams](#) has created a COM wrapper for Spout to use with [LED mapping](#).

But he has gone the extra mile to make the com object general purpose by creating a [GitHub repository](#) with full source code, a Visual Studio solution and clear instructions on how to use it.

This will be extremely useful for anybody using C# and .NET in general.

## 9.7 Monogame MONOGAME

Itay Gal has developed a C# Spout implementation for MonoGame DirectX version, with some modifications to monogame's RenderTarget2D and Texture2D classes.

The underlying of MonoGame DirectX version is SharpDX and this code can be easily converted to use that.

It has been tested using 64bit TouchDesigner and is available as a [GitHub repository](#).

## 9.8 Eclipse eclipse

Kyle Watson has developed a [Java project with Eclipse](#) utilizing Processing 3.0 to output controllable vj visuals.

## 9.9 Openframeworks

[Eliot Woods](#) has created an [Openframeworks addon](#).

Mat Loz has created an [Openframeworks addon](#).

[Bartosh Polonski](#) has created an [Openframeworks addon](#).

[Bart Moens](#) has created an [Openframeworks addon](#).

[Ben McChesney](#) has created the "ofxOpenVJ" addon to create audio reactive visuals with the Microsoft Kinect.

## 9.10 Cinder

[Bruce Lane](#) has created a [Cinder block](#) for Spout and a [hap codec video player](#) which renders to Spout.

## 9.11 Videodromm

Bruce Lane has created [Videodromm](#), to create multiplatform applications for VJing/installations/VR/web with Spout support. See more in the [documentation](#).

## 9.12 Processing

[Martin Froehlich](#) is contributor to the [Spout For Processing](#) library.

## 9.13 FFGLmaker

[Elad Gariani](#) has created a quick and easy script system for OSX to generate a FreeframeGL plugin from a simple shader text file using [GLSL Sandbox](#) and [ShaderToy](#) shaders.

NO INSTALLATION OF XCODE OR PROGRAMING KNOWLEDGE IS REQUIRED.

This is a [ruby](#) script based on [Thor](#) command-line interface and [ERB](#) templates and uses a [source code template](#) based on [ShaderMaker](#) which is a spin-off from the Spout project. See more detail on the [Github Repository](#).

## 9.14 OmniMap



[OmniMap API](#) by [The Eluminati](#) is a perspective projection mapping plugin and fisheye projection mapping toolkit for OpenGL and DirectX applications.

The API provides optical and geometric correction techniques which prevent video projected onto spherical displays from appearing distorted and includes Spout single channel and spout cubemap output.

The API is available on [GitHub](#).



## 9.15 GitHub repositories



<https://github.com/leadedge/Spout2>



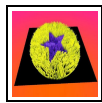
<https://github.com/benkuper>



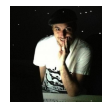
<https://github.com/sloopidoopi/Spout4Unity>



<https://github.com/Reymenta-Visuals/Cinder-Spoutbox>



<https://github.com/brucelane>



<https://github.com/elliotwoods/ofxSpout>



<https://github.com/Mat-Loz/ofxSpout>



<https://github.com/mbechard>



<https://github.com/baku89/VirtualMapper>



<https://github.com/charlesveasey/vdome>

## 10. Error reporting

Report errors on the Spout website forum :

<http://spout.zeal.co/forums/forum/spout/>

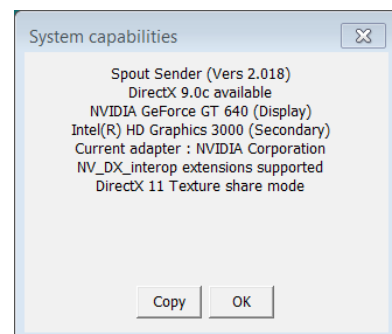
Before reporting an error, please look first for a solution in this manual.  
Please provide the following information:

- Hardware configuration.
- Module affected.
- Detail of how the problem is occurring.
- Error messages or undesirable behavior.
- Steps to reproduce the problem.

The Demo Spout sender and receiver provide information on the system configuration and capabilities for texture sharing.

Select "Help > System capabilities" to show a dialog with information relevant for tracing problems.

Click "Copy" to copy this information to the clipboard. You can then paste this information to a text file or to a forum post.



Report problems to the [Spout Support](#) forum.

## 11. Credits and licencing

### 11.1 Credits

Credit is due to [Elio Wahlen](#) who wrote a FreeFrameGL plugin to take output from [VVVV](#) and provided [open source](#) of his connector code which was the basis for developing *Spout*.

Credit is also due to [Frederik Tilkin](#) for his work on [Wyphon](#), a protocol for texture sharing on Windows systems.

Credit and thanks to :

[Malcolm Bechard](#) for work on the Spout SDK and his contributions of the memory mapping class and DLL project.

[Ben Kuper](#) for the development of plugins for Unity, AIR and After Effects.

[Bruce Lane](#) for Cinder developments and more.

Special thanks to the many who have given their support and advice with the development of Spout.

## 11.2 Licencing

***Spout*** - Copyright (C) 2014-2017 Lynn Jarvis.

Spout is available free of charge under the Simplified BSD licence.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. 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.

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 HOLDER 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.

~~~~~