

# FaustLive - User Manual

## v.2.44

Grame, Centre National de Création Musicale

<research@grame.fr>

[ANR-12-CORD- 0009] and [ANR-13-BS02-0008]

February 12, 2016

FaustLive is an advanced self-contained prototyping environment for the Faust programming language with an ultra-short edit-compile-run cycle. Thanks to its fully embedded compilation chain, FaustLive is simple to install and doesn't require any external compiler, development toolchain or SDK to run.

FaustLive is the ideal tool for fast prototyping. Faust programs can be compiled and run on the fly by simple drag and drop. They can even be edited and recompiled while running without sound interruption or JACK disconnection.

## Contents

<b>1</b>	<b>Dynamic Compilation</b>	<b>3</b>
1.1	Open a DSP . . . . .	3
1.2	Drag and Drop . . . . .	3
1.3	Paste . . . . .	3
1.4	Dynamic source edition . . . . .	3
<b>2</b>	<b>Window-specific actions</b>	<b>4</b>
2.1	Crossfade Characteristics . . . . .	4
2.2	Reversed Drag and Drop . . . . .	4
2.3	View SVG Diagram . . . . .	4
2.4	Duplicate Window . . . . .	4
2.5	Export . . . . .	4
2.6	Control Interfaces . . . . .	4
2.6.1	HTTP Interface . . . . .	4
2.6.2	OSC Interface . . . . .	5
2.6.3	MIDI Interface . . . . .	5
2.7	Polyphonic support . . . . .	5
2.8	Compilation Options . . . . .	6
<b>3</b>	<b>General Actions</b>	<b>6</b>
3.1	Take Snapshot . . . . .	6
3.2	Recall/Import Snapshot . . . . .	6
3.3	Open Component Creator . . . . .	6
3.4	Close All Windows . . . . .	7
3.5	Quit FaustLive . . . . .	7
3.6	General preferences . . . . .	7
3.6.1	Compilation Options . . . . .	7
3.6.2	Audio . . . . .	7
3.6.3	Network . . . . .	7
3.6.4	Style . . . . .	7

# 1 Dynamic Compilation

FaustLive embeds a full compilation chain to provide an ultra-short edit-compile-run cycle. The ensuing features are :

## 1.1 Open a DSP

The Faust file you choose to open is compiled and runs into FaustLive.

- Open ... : you can open a file saved on your disk
- Open Example : you can open an example from the list
- Open Recent File : the recently closed DSP are saved in this list

## 1.2 Drag and Drop

On a running application, you can drop Faust code as :

- A string  
process=+;
- A .dsp file  
/Users/you/Desktop/plus.dsp
- A URL  
<http://faust.grame.fr/www/vumeter.dsp>
- A .wav file  
/Users/you/Desktop/myBird.wav

The dropped code is compiled and the new application replaces the previous one.

## 1.3 Paste

You can also paste Faust code with the same result as Drag and Drop.

## 1.4 Dynamic source edition

You can choose to edit your Faust code. The file is opened in the default application for .dsp files (Don't forget to configure file association : c.f README). Everytime you will save your code in the editor, your running DSP will be updated.

Three cases exist, depending on the origin of your application. In case the origin of your application is

- A file : it is opened
- Not a file (string, url, example, ...) : you are asked to save your Faust content in a new file.

- If you choose to save it, your new file is opened and becomes the new origin of your DSP.
- If you don't want to save it on disk, a temporary file is created and opened. The origin of your DSP remains a string.

## 2 Window-specific actions

### 2.1 Crossfade Characteristics

When the DSP is modified or switched within a window (drag and drop, source edition, ...), the two applications are crossfaded to avoid brutal interruption in the sound. If you are using JACK as audio driver, the connection remain for the common ports. If there are new ports, it is left to the user to connect them.

### 2.2 Reversed Drag and Drop

You can drag your Faust code from your window to another window or another application like your Desktop, Browser, Text editor, ...

### 2.3 View SVG Diagram

You can view your SVG Diagram in the default application for .svg (Don't forget to configure file association : c.f README). Everytime your DSP is modified, the SVG diagram is updated.

### 2.4 Duplicate Window

A Window can be duplicated with all it's parameters (graphical params, audio connections, compilation options, etc)

### 2.5 Export

Thanks to the remote compilation service, FaustWeb, you can export your Faust code into any kind of available application or plugin.

### 2.6 Control Interfaces

You can duplicate the control interface of your application. The local and remote interfaces are synchronized.

#### 2.6.1 HTTP Interface

You can control your application on a browser through an HTTP interface. To enable this feature, you have two possibilities :

- In "Parameters", in the "HTTP Interface" tab, check "Enable Interface"

- In "Preferences", in the "Network" tab, check "Enable HTTP Interface automatically" (any new window will be enabled for HTTP control)

The port it starts on is updated in the parameters. The default port is 5510.

To access the interface, you can enter the address in your browser : `http://yourIPaddress:port`.

Or, you can choose view QRCode then scan the code with your device to access the page directly.

## 2.6.2 OSC Interface

You can control your application remotely with the OSC protocol. To enable this feature, you have two possibilities :

- In "Parameters", in the "OSC Interface" tab, check "Enable Interface"
- In "Preferences", in the "Network" tab, check "Enable OSC Interface automatically" (any new window will be enabled for OSC control)

You can choose the port for incoming/outcoming and error messages. The destination IP for outcoming messages will be automatically set when the first message is received.

## 2.6.3 MIDI Interface

You can control your application with a MIDI controller. To enable this feature, you have the following possibility :

- In "Parameters", in the "MIDI Interface" tab, check "Enable Interface"

## 2.7 Polyphonic support

By convention Faust architecture files with polyphonic MIDI capabilities expect to find a *freq*, a *gain* and a *gate* user interface parameter :

```
midigate = button ("gate"); // MIDI keyon-keyoff
midifreq = hslider("freq[unit:Hz]", 440, 20, 20000, 1); // MIDI keyon key
midigain = hslider("gain", 0.5, 0, 10, 0.01); // MIDI keyon velocity
```

The polyphonic architecture will automatically allocates voice for the instrument and takes care of dynamic voice allocations and control MIDI messages decoding and mapping. By convention all allocated voices are grouped in a global "tabgroup", so that a user interface builder may display them separately.

Then this dsp object can be used as usual and connected with the wanted audio driver and possibly other UI control objects like OSC, HTTP interfaces, etc. Having this UI hierarchical view allows for instance complete OSC control of each single voice and its control parameters.

The number of voices for the polyphonic instrument can be chosen in the "Parameters", in the "Polyphony support". The polyphonic instrument can then be controlled with MIDI messages by checking the "Enable Interface" in the "MIDI Interface" tab.

## 2.8 Compilation Options

- Code Generation Options : In the parameters, you can change the compilation options of your running DSP. Faust compilation options are available in Faust documentation. LLVM optimization level is a number contained between 0 and 3.
- Additional Compilation Step : a new call to the Faust compiler is executed with new output feature (for example : `-lang ajs -o filename.js`). This call will be added to any compilation/recompilation within the window.
- Post-Compilation Script : after all the compilation steps, you can add an arbitrary command line.

## 3 General Actions

### 3.1 Take Snapshot

When you take a snapshot, the state of your application is saved : all the windows, their parameters (graphical parameters, connections, position on screen, compilation options, etc, ...). The created archive is a .tar (platform-independent) that you can then reload on any FaustLive.

### 3.2 Recall/Import Snapshot

You have two different ways to reload a snapshot into FaustLive :

- Recall Snapshot : closes all the current windows to restore the exact state that you saved in your snapshot.
- Import Snapshot : adds the content of your snapshot to the current state of the application.

In case your source files were modified or deleted outside of FaustLive's execution, an internal copy of your files will be used to restore the state.

### 3.3 Open Component Creator

The component creator is a tool to combine your DSPs. The columns represent parallel composition and the rows sequential composition. You can add as many rows and columns as you like. You can also have a feedback DSP. The resulting DSP is a sequence of parallel components.

The parallel composition is "stereoized" before being put in sequence. And the recursive composition with the feedback DSP is "recursivized". This way, the resulting DSP always has 2 inputs and 2 outputs. To know the detail of the functions `stereoize` and `recursivize`, see `music.lib`.

### **3.4 Close All Windows**

You can close every window without quitting FaustLive and without saving them.

### **3.5 Quit FaustLive**

When you quit FaustLive, the state of the application is saved, so that it can be recalled at the next execution of FaustLive.

### **3.6 General preferences**

#### **3.6.1 Compilation Options**

Any changes in the compilation options will be taken into account for all the new windows created. The one already running keep their parameters.

#### **3.6.2 Audio**

Depending on your operating system and on your version of FaustLive, certain drivers are or not available. The default drivers are :

- CoreAudio on OSX
- JACK on Linux
- PortAudio on Windows

When you switch the audio architecture or its parameters, every running window is switching. The audio clients stop, the architecture is switched and the audio clients are restarted. If the update is not successfull, the previous architecture is restored. If it cannot be restored, the windows are closed.

#### **3.6.3 Network**

- Enable HTTP interface automatically
- Enable OSC interface automatically
- Remote Compilation service : by default, FaustLive uses faustservice.grame.fr as compilation service but you can choose any another FaustWeb service.
- Remote Dropping Port : port used to start the internal server, taking care of drops on html remote interfaces.

#### **3.6.4 Style**

The style is automatically changed when you click the style button. You don't have to save the changes to keep the style preference.