

Graph Designer - User documentation

June 17th , 2024

Version 1.0

Introduction	3
How to install and run Graph Designer	3
From SDK	3
User interfaces	4
Main window	4
General Configuration of Graph designer	5
Graph menu	7
Open	7
New	8
Save	8
Publish	9
Compile graph menu	10
Button layout	11
Manipulate	11
Add Component	12
Add connection	12
Property window	13
Known issues	14
Release V0.1	14

Introduction

In this document you will find information about our graph designer human interface tool named Graph Designer (GD).

The purpose of this tool is to be able to design a graph using components (that can be also set using MATLAB or python tools) up to the generation of the source code to run on the target.

Keep in mind the following information:

- A GD project is saved as a **.graph** file (the `trace_native.graph` is only the backup in case of crash)
- A graph that has been “published” will keep the **.graph** extension which can be used directly by our audio toolchain
- **BUILD** folder is generated when compiling the graph1

We recommend checking the Release [V1.0](#) paragraph!

How to install and run Graph Designer

From SDK

- Our Python version that we use is **3.10**
- If you have downloaded the our `gap_sdk` you can find Graph designer in `gap_sdk/tools/audio-framework/graph_designer`
- **You must install all the requirements from the SDK**
- After that you must also install the requirement for Graph designer using the following commands:

Unset

```
cd tools/audio-framework/graph_designer/  
pip install -r requirements.txt  
python3.10 graph_designer.py
```

- If matlab is installed in you PC then run the following command as well:


Unset

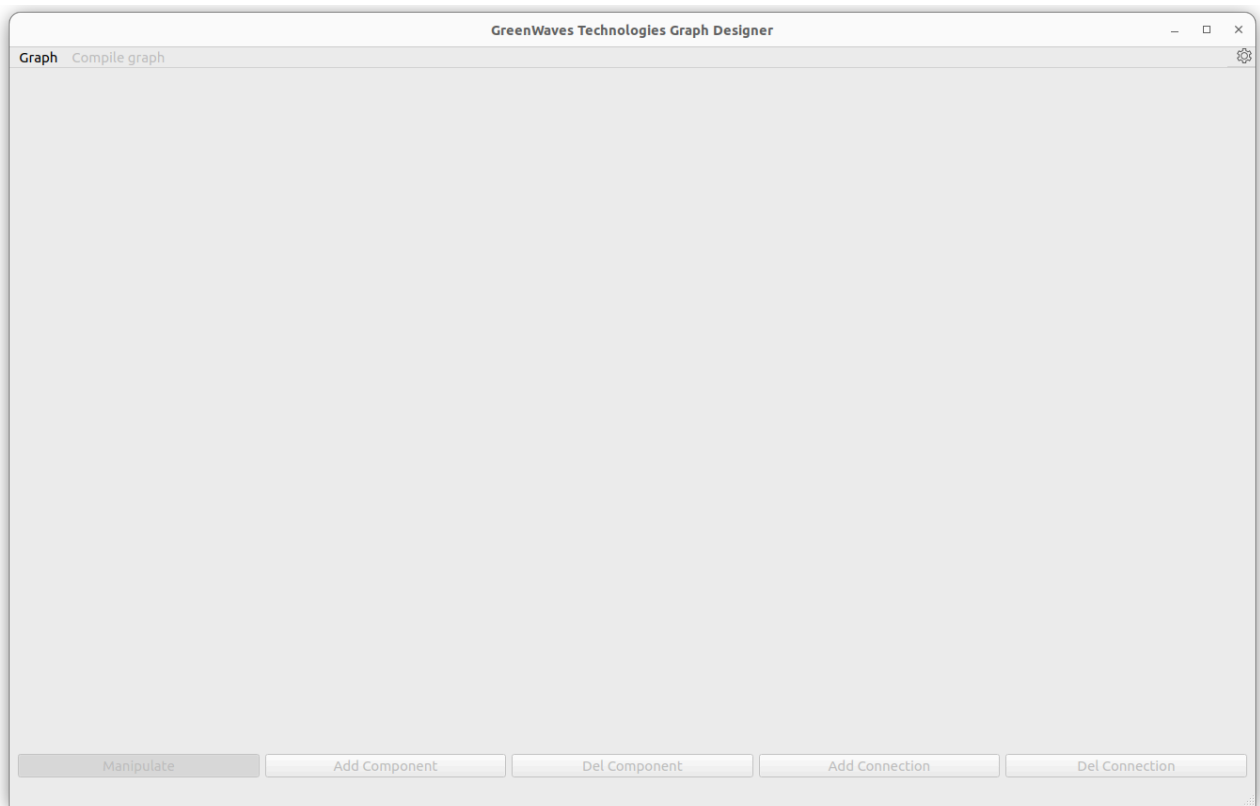
```
pip install matlabengine
```

User interfaces

Main window

This windows contains all the options you need to build a graph, you will find:

- Graph menu,
- Compile graph menu,
- and a General config menu that is display as an icon 
- And All the button needed to create/manipulate a graph



General Configuration of Graph designer

When you click on the top right icon  the following configuration window will appear (see next page).

For each parameter is composed of two lists:

The left one is the path that is saved in the configuration but not applied, and the right one (useful when you need to save paths) is the paths that are set. So, in order to activate a path, you have to move it from the left to the right side. To deactivate a path, you have to do the opposite.

It allows you to set the following three parameters by default to apply in projects:

1. **Project places:** This path is used to save the Graph Designer project in a default folder. It can have multiple paths, but keep in mind that the first link at the top of the list will be the first path parsed.

By default it is set to your ~ path (*home/user*)

2. **Component path:** These path are use to check where the components are located, in order to propose them when trying to add them into the graph

By default it is set to (*../libs/components* and *../libs/bindings*)

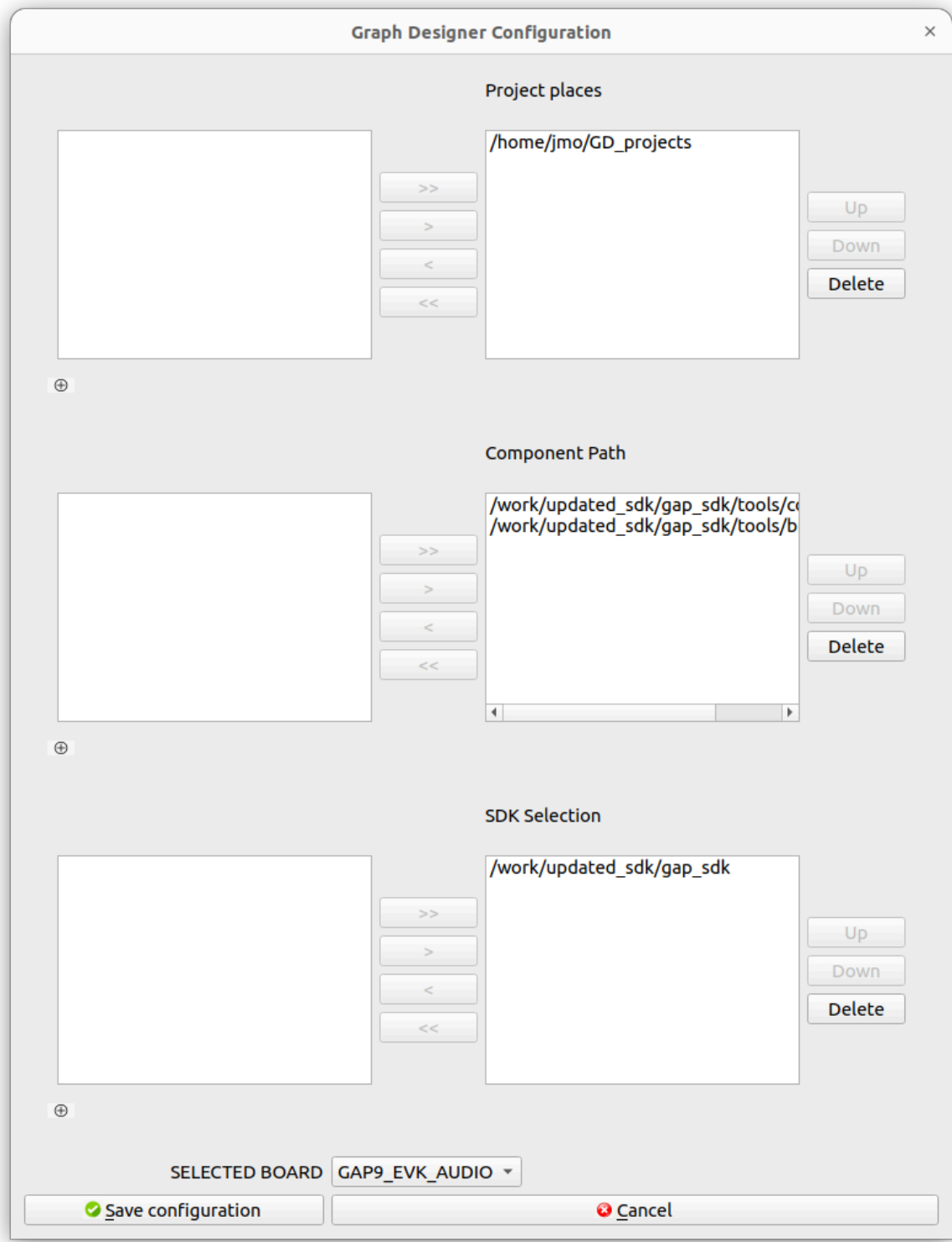
3. **SDK Selection:** These SDK path will be used in order compile the .graph which will generate all the sources for the GAP platform

No default value

4. **SELECTED BOARD:** In our SDK you have multiple board/target that you can select

No default value

When you have saved the configuration by clicking on the save configuration button, all the information is saved into a hidden file located again in *~/ .gdrc*



Graph menu

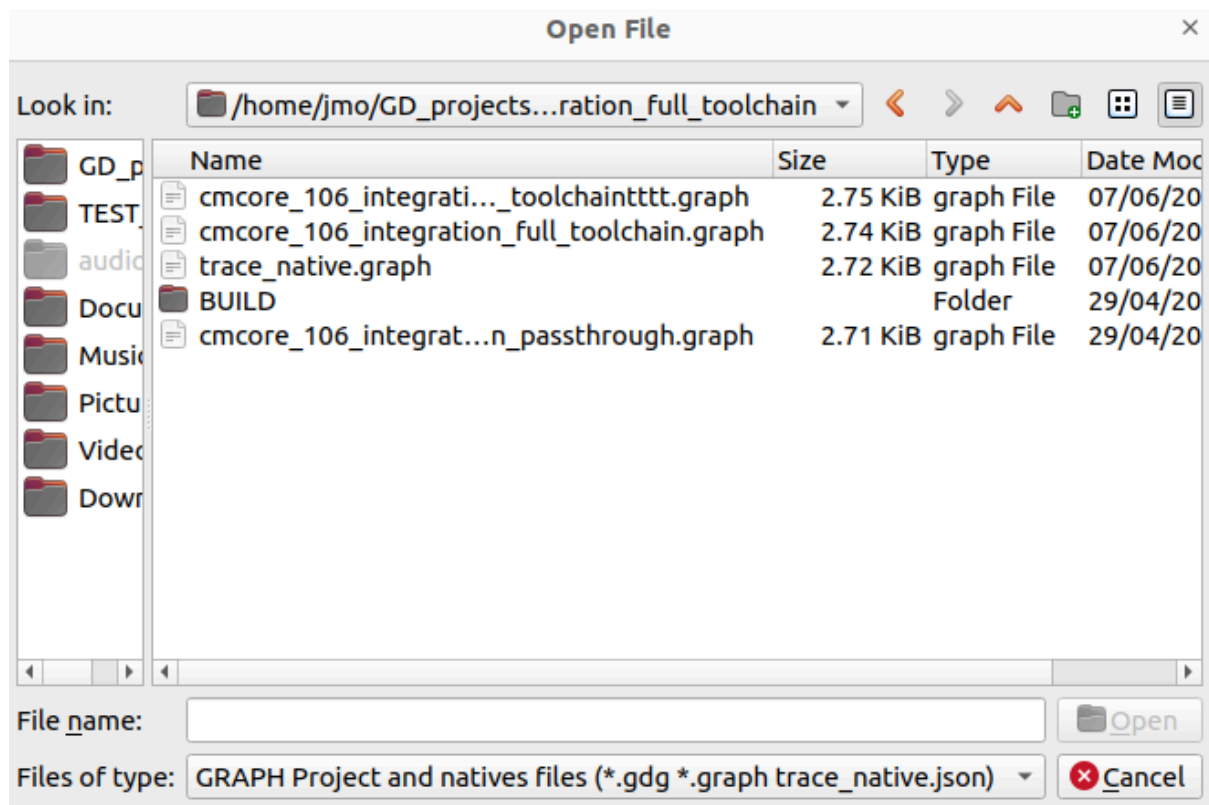
In this menu you will find:

- Open: In order to open a graph designer project (.gdg or trace_native.json)
- New: Create a new project
- Save: Save the graph project (.graph file)
- Publish: Generate a file (.graph) from a GD project that can be readable by the toolchain in order to generate all the sources.(A .graph that has been generated by the audio toolchain can be read by GD)
- Quit: Close Graph designer

When no project is loaded Save and publish options are not available.

Open

When you have clicked on the Open menu the following window will appear:



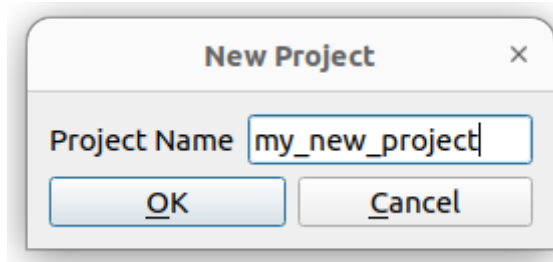
It will open where you have set the **Project places** in the general configuration (see [General Configuration of Graph designer](#))

Click on the desired project and click on the .graph or trace_native.graph file (if a crash has occurred)

New

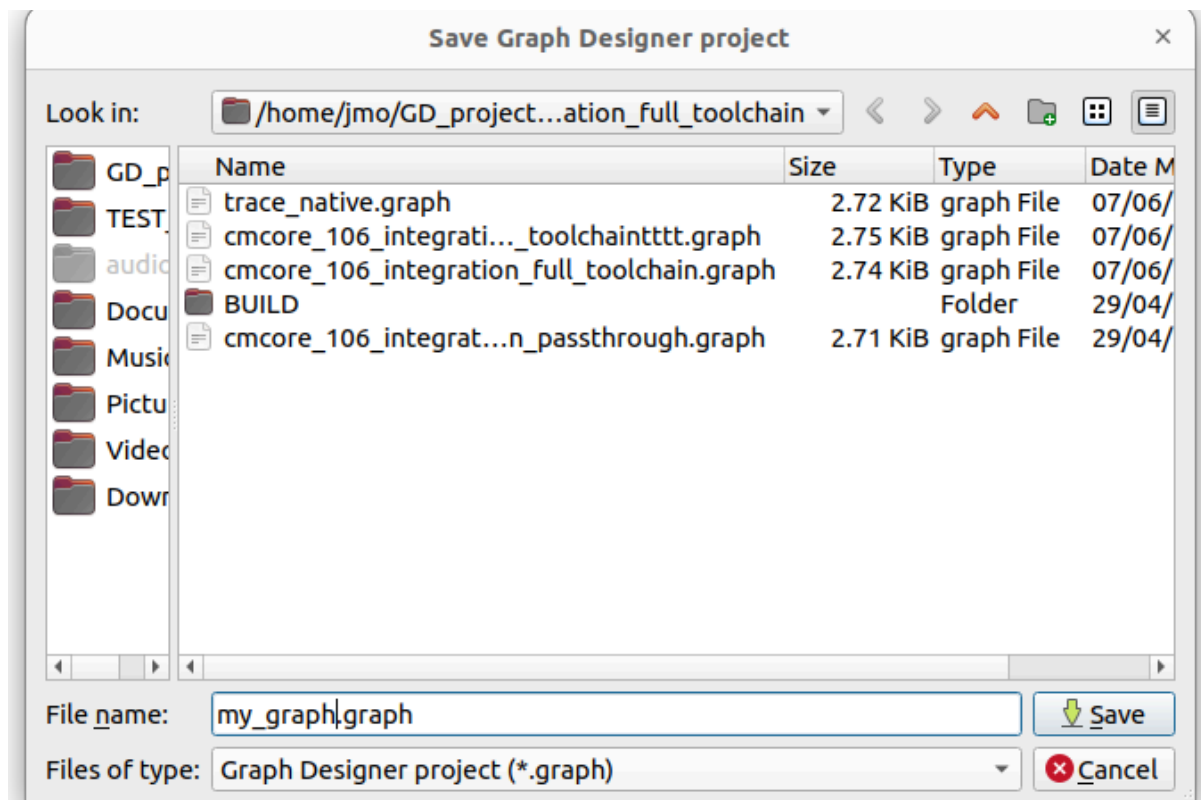
A small dialog will be displayed, you just have to write the name of the project and then click on OK, It will generate the folder of the project in the GD project place (see [General Configuration of Graph designer](#))

If multiple path, keep in mind that if the project is already exist it will display an error and the project will not be created



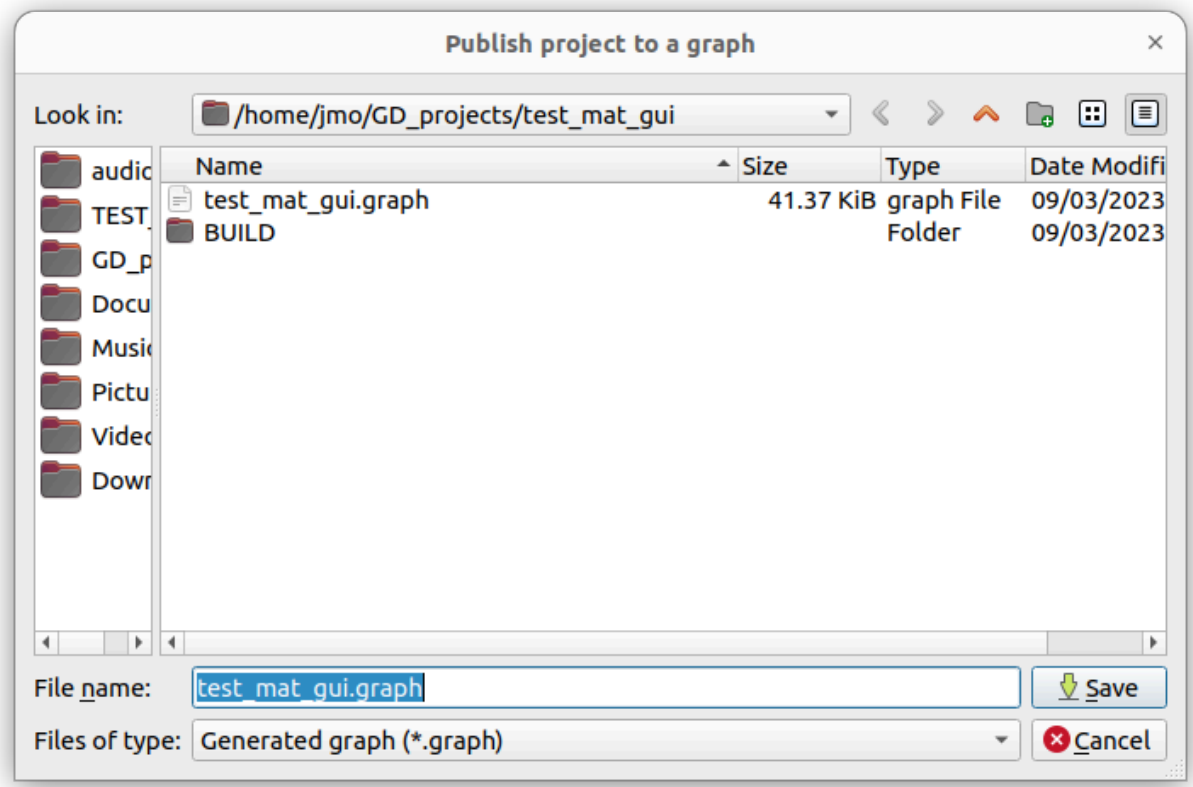
Save

Will save the project as **.graph** file, a dialog will be opened in order to save the project in the project place with the same or a different name:



Publish

Will propose to publish the current project to a **.graph** file, which is mandatory for the compiling toolchain for generating all the required sources for the GAP platform. The following dialog will be displayed:

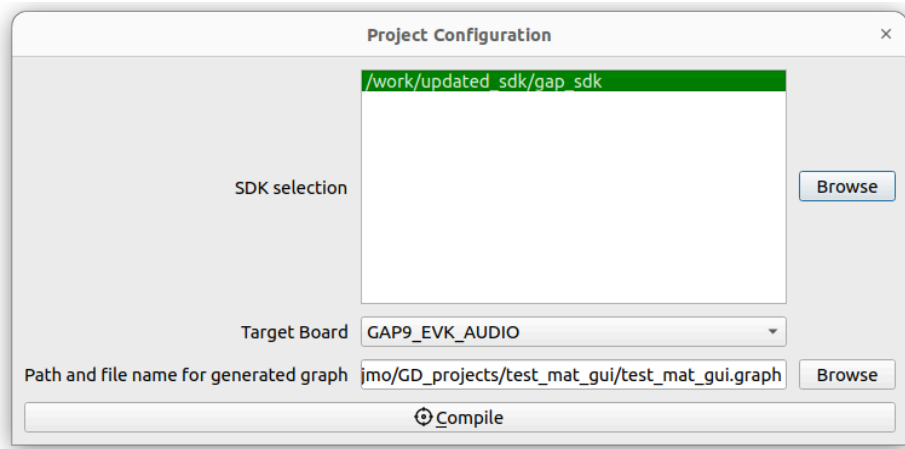


Compile graph menu

This menu is only available when a graph project is opened

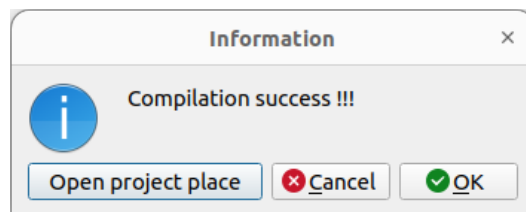
When clicking on its submenu *Compile graph* a window will appear (see below):

1. In order to compile the graph you need first to select a SDK version by selecting the **gap_sdk folder** using the Browse button (A default SDK selecting may appear if you have set-up the General configuration of Graph Designer)
2. You need to select the configuration (target board)
3. And finally you need to put the path and the name file of the generated graph with the .graph extension, a Browse button can be clicked in order to set the place and the name of the generated graph where the project has been saved

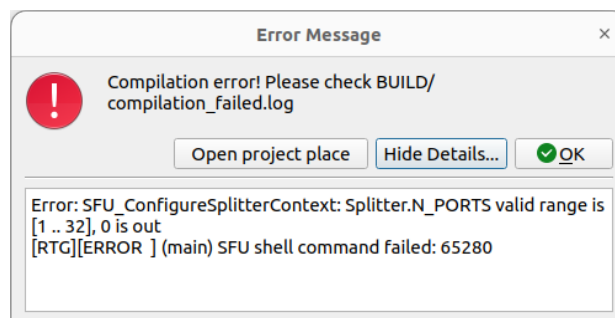


When you click on the Compile button it will try to generate the sources.

If the compilation is successful a dialog will appear, you are able to open the project place by clicking on the “Open project place” button:



Otherwise a dialog will showing the list of errors:



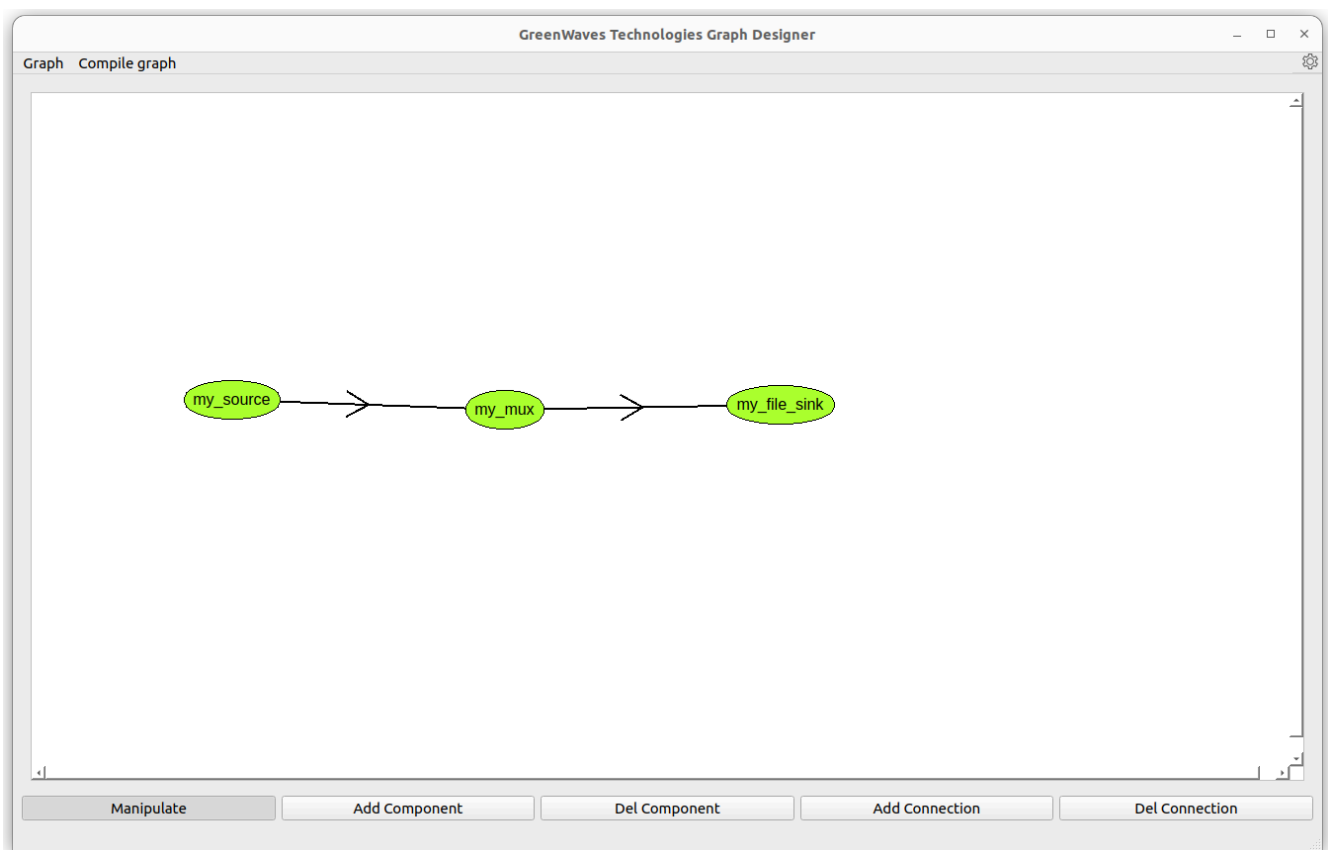
Button layout

These buttons are available only when a graph is opened

You have five buttons for manipulating the graph:

1. **Manipulate:** When it is activated you are able to move the position of a node
2. **Add Component:** Will propose you to add a component with a name and by selecting also the available ones
3. **Del Component:** Delete a component in the graph
4. **Add Connection:** Connect two components
5. **Del Connection:** Delete the connection between two components

Here after is a graph example using a File_source, Mux and a file_sink with the FC implementation each

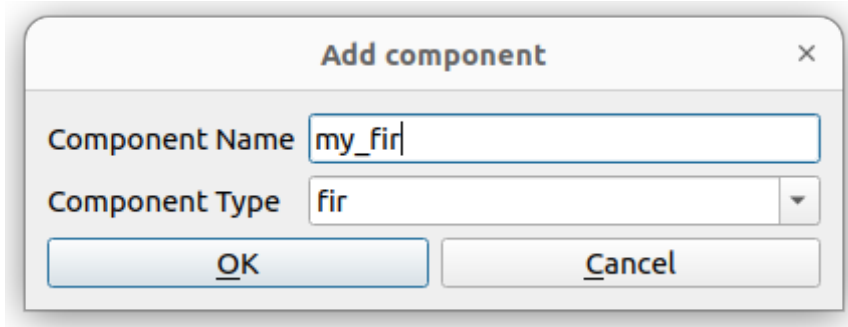


Manipulate

Manipulate mode is the default which means that you are able to move the nodes where there are placed, you can also double click on it in order to set-up the component (see [Property window](#))

Add Component

A dedicated window will be shown as follow, you have to give a name and the type of component that you want to add in the graph:



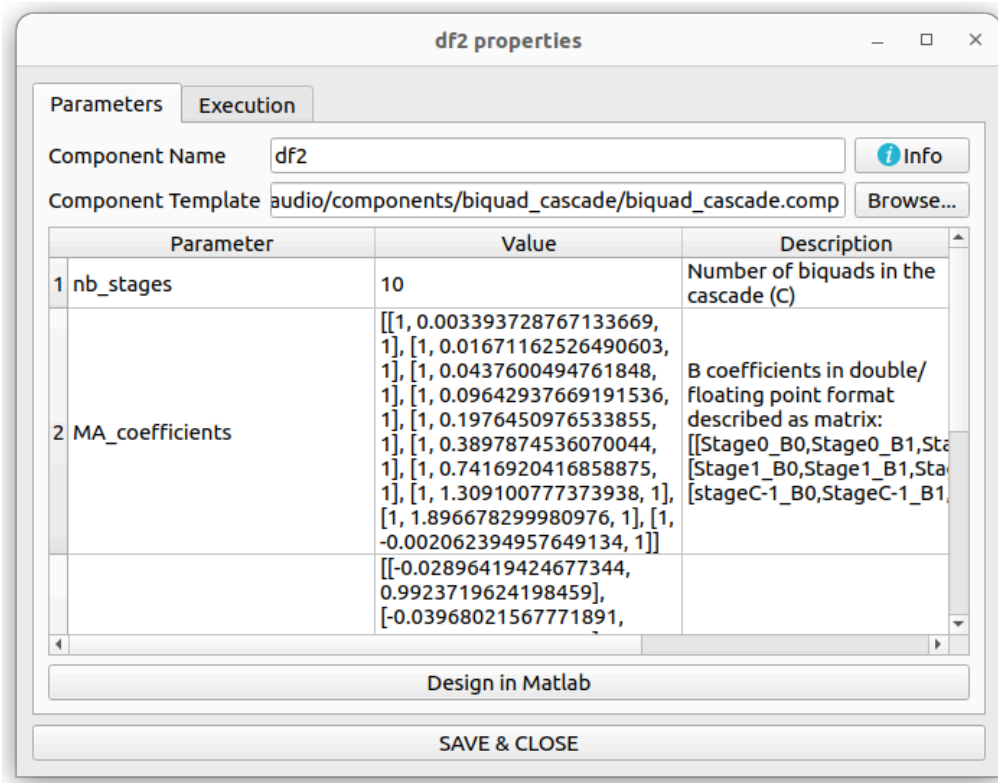
The image shows a dialog box titled "Add component" with a close button (X) in the top right corner. It contains two input fields: "Component Name" with the text "my_fir" and "Component Type" with the text "fir" and a dropdown arrow. At the bottom, there are two buttons: "OK" and "Cancel".

Add connection

For example if you want to connect a component named A to B you have to hold the click from A to B.

Property window

The following window will appear when you click on a placed node:



This windows is composed of two tabs:

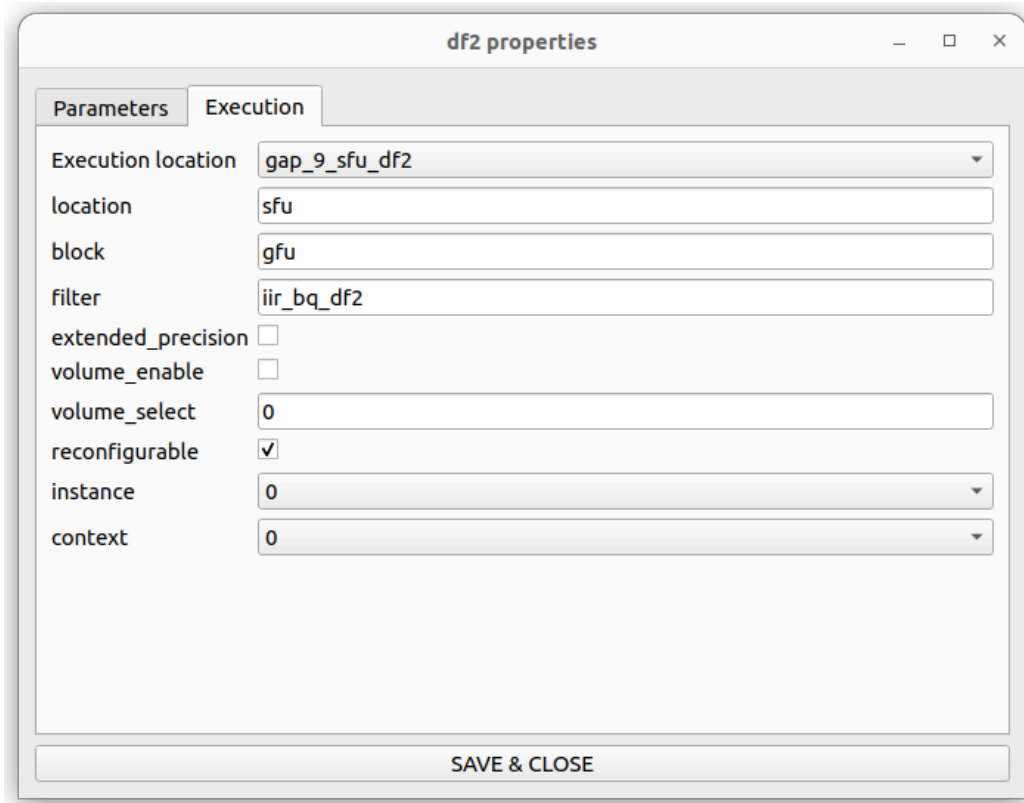
1. Parameters:

You will find:

- the name of the placed node,
- a button “Info” that will show you the parameters and also the image of the implementation
- The list of parameters with their values and also the description for each parameter
- If the matlab implementation as been found in the component then you have the button Design in Matlab available

2. Execution:

If the implementation as been done on the Hardware side you will have the following tab (next page is the Execution location example of a biquad_cascade)



Known issues

- It is possible that it may be difficult to remove the connection between the different components. We recommend moving one of the two nodes in such a way that there is a 45-degree difference in positioning.
- Some of component are still in development
- If there is an exception, you can find the log file in the project where you are currently working on (*debug.log*) that is useful in case of a crash.
- Please select an implementation for each component

Release V1.0

- GD can generate GAP (FC only) sources using the audio toolchain (mapping tool and runtime generator)
- Mixer and splitter are **NOT** handle yet
- Can read .graph file generated from CMCore