

# Writing FrAid Plugins

---

NOTICE: ""

---

## Table of contents

1 FrAid Plugins.....	2
----------------------	---

**Note:**

You need to know some Java to use this feature.

## 1. FrAid Plugins

As already noted in the [Graphics Panel](#) section a FrAid plugin consists of:

- a FrAid library function - the entry point for your algorithm;
- a Behavior Bean - subclass of `org.fraid.plugin.beans.BehaviorBean`
- an Algorithm - subclass of `org.fraid.plugin.algorithm.PaintAlgorithmThread`

The task of writing FrAid plugins may look a bit complicated initially but may be considerably simplified if you use the simple `org.fraid.utils.CreatePlugin` code generator. Follow these steps:

- Download `frd_gen_code.jar` into the bin directory where all other FrAid jars live.
- Run `java -cp frd_gen_code.jar org.fraid.utils.CreatePlugin -source myFun`. Hint: see what files were created and where they are.
- Edit the generated files and make them do whatever you want. If this is the first time you do this you may skip this step and just observe how this works. What the default generated plugin does is just replicate the behavior of the `plot` library function and the associated `FunctionPlotPlugIn`.
- Run `java -cp frd_gen_code.jar org.fraid.utils.CreatePlugin -compile myFun`. If you get any compile errors try the `-printOnly` option and run the compilation manually (sometimes weird errors are reported which go away if you just compile manually).
- Optionally run `java -cp frd_gen_code.jar org.fraid.utils.CreatePlugin -jar myFun` if you want to jar the plugin. Hint: see what goes into the jar.
- Run `fraid.sh` or `fraid.bat`
- Execute your function - for the above example: `myFun( 'sin(x)', 'cos(x) );`
- Run `java -cp frd_gen_code.jar org.fraid.utils.CreatePlugin -delete myFun` if you want to start all over.

Don't forget to check the examples in the FrAid sources

""

""