

[TracNav](#)

- TOC "nodeit" is empty!

jpf-shell

This projects aims to provide a basis for the development of GUI's for JPF. Since part of the JPF ideology is that "*one size does **not** fit all*", it is expected that many different domain specific GUIs (Shells) will be developed for JPF. Jpf-shell serves to be the common framework that these GUIs can build from. Advantages of using jpf-shell include being able to launch the GUI from JPF, minimal need to modify current property configurations for JPF and integration with the Netbeans and Eclipse plugins right out of the box.

Repository

The Mercurial repository for jpf-shell is at <http://babelfish.arc.nasa.gov/hg/jpf/jpf-shell>

Installation

Installing jpf-shell is just like installing any other jpf project. Download the jpf-shell.zip file attached to this page and extract its contents. (Or if you want to be on the bleeding edge clone the project from the mercurial repository and run the 'build' target). After that make sure to add jpf-shell's path to the extensions property in your site.properties file. Usually adding the following lines to your site.properties will work (YMMV):

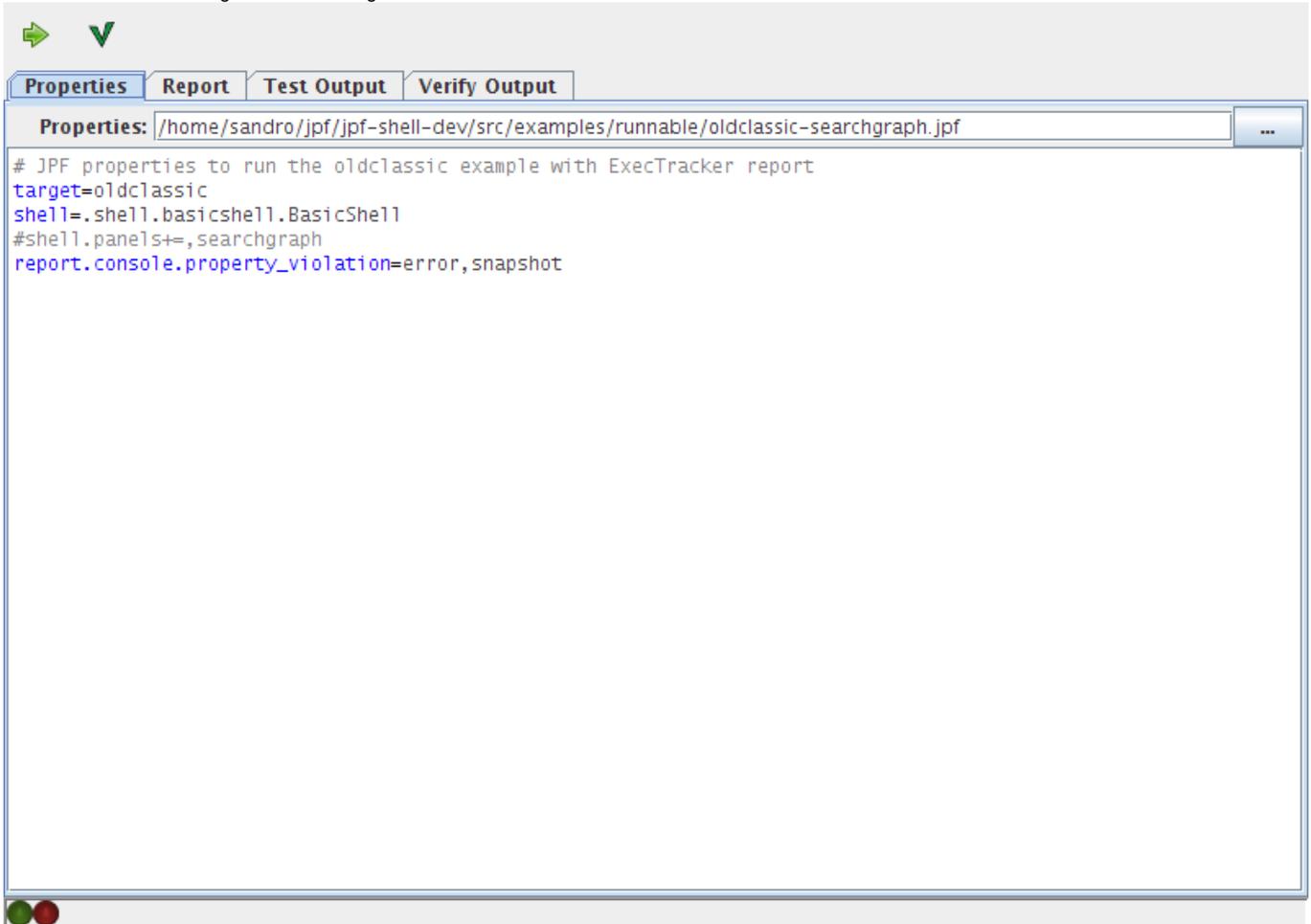
```
#Shell extention
jpf-shell = /path/to/the/shell/project/jpf-shell/
extensions+=, ${jpf-shell}
```

Now all that needs to be done is adding the following the property to your configuration:

```
shell=.shell.basicshell.BasicShell
```

When JPF is started it will now start a shell instead of running the verification.

You should see something like the following:



Choosing the Panels that are used

By default (as seen in the screenshot above) the default panels are the following: Properties, Report, Test and Verify. The reason for this is because of the "jpf.shell" property defined in the jpf.properties file found in the jpf-shell project. To get a better idea of what is going on, look at the following snippet from the file.

```
#Shell Panels
shell.properties = .shell.panels.PropertiesPanel
shell.report = .shell.panels.ReportPanel
shell.test = .shell.panels.TestConsolePanel
shell.verify = .shell.panels.VerifyConsolePanel
shell.searchgraph = .shell.panels.SearchGraphPanel

#Shell configuration properties
shell.available_panels = properties,report,test,verify
shell.panels = ${shell.available_panels}
```

jpf-shell looks to "shell.panels" to define which panels (and in what order) to display in the shell. To modify this list of panels just modify the shell.panels property as needed. For example, if you wanted to add the SearchGraph panel to the shell you can either change the shell.panels property in the jpf.properties file (Note: this will affect ALL usages of the jpf-shell project) or in your own *.jpf file you can add the following:

```
shell.panels+=searchgraph
```

At the time that I am writing this all of the available panels are shown above. For the most up to date listing of the panels available your best bet it to look into the jpf.properties file.

I want the sourcelinks to open in my favorite editor!

Note: If you are running starting jpf-shell from either the [eclipse-jpf?](#) or [netbeans-jpf](#) plugins then the sourcelinks will automatically use the already opened IDE. To stop this behavior go into the Preferences for the plugin and set the 'Shell Port Number' to -1.

To have jpf-shell open sourcelinks in the editor of your choice find the command that will execute the command of your choosing to open the file.

```
#jpf-shell execute will execute the value of shell.editor with #{file} and #{linenumber} replaced by the actual file path

#For netbeans
shell.editor = netbeans --open #{file}:#{linenumber}
#For vi
shell.editor = vi +#{linenumber} #{file}
```

Developing custom shells

[Click here to get more information on how to develop shells](#)