org.lcsim with Jython in Batch Mode

Why would I want to run outside of JAS3 ?

While it is more convenient to do the analysis inside of JAS3 in most cases, there are cases where running without a graphical user interface is desirable. Running a complex analysis on a batch farm would be an example of such a case.
In order to perform the analysis outside of JAS3, a small wrapper is needed, that takes care of loading the file, writing the output and adding all necessary Drivers. When running this wrapper you will have to make sure that your CLASSPATH environment variable is set correctly

If you use JAS3 to download the org.lcsim .jar files, you can be sure to have an up-to-date version. Add the .jar files in the `extensions` and `lib` directories to your CLASSPATH. Be sure to look in the JAS3 installation directory as well as the user directory for these files. See Running JAS Programs in Batch Mode for instructions on how to modify your CLASSPATH variable

We present two examples of wrapper classes here:

- A Java wrapper that can be used to load your existing modules
- A Jython wrapper that can be used to load your existing Java modules as well as modules written in Jython that you may have

Independent of the wrapper you choose, please make sure that all your **Java** analysis modules have been compiled with Javac and that the location of the resulting .class files is part of the CLASSPATH environment variable.

Using the Java wrapper

Please save MainLoop.java, Analysis101.java and psiMuMu.slcio to your harddisk. psiMuMu.slcio is a small datasample, Analysis101.java is our analysis module, which is taken straight from the JAS3 examples, and MainLoop.java is the wrapper that takes care of loading the analysis module correctly.
After compiling the java files with

```
javac Analysis101.java
javac MainLoop.java
```

you can start the program with

```
java MainLoop
```

which will loop over the data sample and create a file `exampleAnalysisJava.aida`

You can add your own analysis modules by replacing

```
loop.add(new Analysis101())
```

with your analysis modules

Don't forget to import your modules into MainLoop.java

Using the Jython wrapper

Please save mainLoop.py, Analysis101.java, Analysis102.py and psiMuMu.slcio to your harddisk. psiMuMu.slcio is a small datasample, Analysis101.java is our analysis module, which is taken straight from the JAS3 examples, Analysis102.py is a slightly modified Jython version of the Java module and mainLoop.py is the wrapper that takes care of loading the analysis module correctly.
After compiling the java file with

```
javac Analysis101.java
```

you can start the program with

```
jython mainLoop.py
```

which will loop over the data sample and create a file `exampleAnalysis.aida`. Please note that this file contains both the output from the Java module and the output from the Jython module.

You can add your own analysis modules by replacing

```
loop.add(Analysis101())
loop.add(Analysis102())
```

with your analysis modules

Don't forget to import your Java modules with

```
import MyJavaAanalysisModule
```

and your Jython modules with

```
from MyJythonModuleName import MyJythonClassName
```

Executing Jython modules in Jas3

Unfortunately, you cannot load the Jython modules in Jas3 as easily as a Java module. However, you can simply load mainLoop.py and add your modules there. Choose 'Run' from the context menu. This will open a Jython console. Type `main()` to start the event loop.

If you have no idea what this section is all about, it is safe to ignore it