# Static Analysis of Critical Sections using Soot

If you need help running the Soot tool, then you can always find the latest command line arguments here.

<http://www.sable.mcgill.ca/soot/tutorial/usage/index.html>

or you can otherwise launch soot by doing the following. The below command line makes the assumption that ‘sootclasses’,’polyglot’,and ‘jasmin’ are located in the current directory of your program.

java -Xmx2g -cp "sootclasses-2.5.0.jar":"polyglotclasses-1.3.5.jar":"jasminclasses-2.5.0.jar":. soot.Main -help

The principle way to have Soot process two classes A and B is just to add them to the command line, which makes them “application classes”:

Soot has it’s own classpath and will load files only from JAR files or directories on that path. (More information on what exactly a JAR file is and why to use it: <http://docs.oracle.com/javase/tutorial/deployment/jar/>) if we keep in mind that a JAR file really is a bunch of .class files packed together then that makes our analysis easier if we can just use .jar files.

## Visualizing Soot Graphs

<http://framepointer.blogspot.com/2010/09/visualizing-soot-graphs.html>

## Debugging common problems

Exception in thread "main" java.lang.RuntimeException: couldn't find class: java.lang.Object (is your soot-class-path set properly?)

Add rt.jar to your Soot classpath.

Exception in thread "main" java.lang.RuntimeException: Attempt to create RefType containing a / -->