User Manual

Product Name: ONI Code Visualization
Team Name: GraalVM UCSC
Date: April 11 2020

Getting Started with GraalVM/Creating JSON Report

1. First you need to install Oracle's mx command

   Link: https://github.com/graalvm/mx

   i. Clone this repository outside of your GraalVM Folder but in the same directory

   ii. File Structure: Project ----> GraalVM, mx

2. Point `JAVA_HOME` to a JVMCI-enabled JDK 8 and export mx as a PATH variable

   i. export PATH=$PWD/mx:$PATH

   ii. ExporJAVA_HOME=/{$YOUR_PATHWAY_TO_JDK}/openjdk1.8.0_242-jvmci-20.0-b02/Contents/Home

   iii. Have these variables initialized in your .bash_profile and then use the command `source .bash_profile` to save the variables

3. Run this command to set up your HelloWorld program

   ```
   echo "public class HelloWorld { public static void main(String[]
   args) { System.out.println(\"Hello World\"); } }" > HelloWorld.java
   ```

4. This will compile the program

   i. `mx build`

   ii. `$JAVA_HOME/bin/javac HelloWorld.java`

   iii. `mx native-image HelloWorld`

5. Then navigate to graalvm/substratevm and run

   i. `mx native-image -H:+PrintAnalysisCallTree HelloWorld`

6. This will generate the JSON static analysis file in graalvm/substratevm/reports.

Intellij Call Trace Plugin

Install IntelliJ. https://www.jetbrains.com/idea/download/#section=mac

Use the folder “calltraceplugin” to run the Intellij Call Trace Plugin. You can test the plugin using gradle. Open the folder in IntelliJ to begin.

Instructions

The json file path must be manually inputted into the program files. In MethodCheck.java and Factory.java, copy your file path to the absolute path line in the code. Such that:

```
String filepath = *Your/file/path*
```

Make sure “calltraceplugin” is open in Intellij. Edit your configurations to run with Gradle. Select this folder in configurations selection, "Gradle Project", and in tasks, input:

```
:runIde
```

Build the project to begin use.