AE Setup Instructions

The Analysis Engine represents and attempts to solve constraint satisfaction problems modeled on top of Java. It can ingest fUML (SysML viewed in activity diagrams) from MagicDraw simulate execution with animation of MagicDraw activity diagrams and data plots.

This page has instructions on how to obtain, build, and run it.

A summary of instructions that may be sufficient:

1. Install the Java JDK 1.6 (or later).
2. Install MagicDraw (MD) if you want to use or develop MD plugins for the AE.
3. If you want to set up for development install Eclipse and the Subclipse plugin (http://subclipse.tigris.org/update_1.8.x).
4. Create/checkout code from SVN: https://sscae-cm.jpl.nasa.gov/svn/cs/trunk
5. If using Eclipse,
    a. Run src/gov.nasa.jpl.ae.tests/TestEventXmlToJava.java as a Java application.
    b. Refresh the src/generated package.
    c. Run src/generated/Main.java as a Java application.
6. Else, if not using Eclipse, enter at a command prompt

```
java -jar AE.jar
javac -cp AE.jar -d bin src/generated/*.java
java -classpath AE.jar:bin generated.Main
```

Detailed Instructions

For non-development

Follow these instructions if you only want to run it and don't care about adding code to the AE or integrating code with the AE. Otherwise, skip this step.

1. Install Java 1.6 or later.
2. Download the AE.jar file
3. Enter at a command prompt

```
4. java -jar AE.jar
   javac -cp AE.jar -d bin src/generated/*.java
   java -classpath AE.jar:bin generated.Main
```

5. Skip to instructions on using with MagicDraw.

Getting Subversion (SVN):

If not using Eclipse download and install Subversion for your platform from http://subversion.apache.org/packages.html.

Steps for getting and running the Analysis Engine (AE) and optionally setting up the development environment:

1. Download and install Java JDK 1.7 (unfortunately 1.6 isn't good enough).
    a. Just do a web search on "jdk"
    b. This link may work JDK: http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1637583.html
2. For participating in Java, python, or jython development, Eclipse is recommended, but this is optional. Skip to step 3 if not using Eclipse.
    a. Download and install eclipse: http://www.eclipse.org/downloads/### I recommend "Eclipse IDE for Java Developers" or "for Java EE Developers"
        i. If on Windows, installing in c:\Program Files\eclipse may avoid classpath changes.
    b. Run Eclipse
    c. Install the Subclipse plugin to access the Subversion (SVN) code repository for the AE.
        i. Go to Help -> Install new software
        ii. click Add
            1. paste http://subclipse.tigris.org/update_1.8.x for Location
            2. maybe put "subclipse" for the Name, and
            3. click Ok
        iii. It should list the package(s) in the window.
        iv. Select them all, choose Next, and complete the installation of the plugin.
        v. As it will tell you, you'll need to restart Eclipse to complete installation, but you might want to install PyDev first in the next step.
    d. If you want to do python or jython development, similarly install the PyDev plugin using the update site:

http://pydev.org/updates

e. Check out the CS project from the SVN repository.
   i. After restarting Eclipse, select from the menu Window -> Open Perspective ?-> Other -> SVN Repository Exploring
   ii. Add a repository by clicking the + icon
   iii. For the server, paste in https://sscae-cm.jpl.nasa.gov/svn/cs/trunk.
   iv. You will be prompted for a username and password.
   v. If it doesn't work, you may need to be given authorization to access the repository. Brad Clement,Thom McVittie, or Sophie Wong may be able to help you.
   vi. If you set up a workspace storage login, do not lose your username/password!
   vii. Accepting the defaults should be fine.
f. Fix project build if you have compile errors (red x markers) on folders or files
   i. If there are errors involving calls to the junit.framework.Assert class, you need to make sure the JUnit4 library is added to the build.
      1. Try right-clicking on an Assert error, choosing Fix project setup, and selecting JUnit4 or the junit.jar for org.junit_4xxx.
      2. If that doesn't work,
         a. try adding it from the Libraries tab of Project-> Properties -> Java Build Path.
         b. It may be located in the Eclipse installation directory in eclipse/plugins/org.junit_4xxx.
      3. If that doesn't work, you could try to do a web search on "junit" and download and install it.
   ii. This shouldn't happen, but if you have errors on calls to japa.parser.* or org.json.*, then the thirdParty source code probably needs to be added to the build.
      1. Select from the menu Project -> Properties
      2. Choose Java Build Path from the left pane.
      3. On the Source tab, choose Add Folder.
      4. Select the folders
         a. thirdParty/douglascrockford-JSON-java-xxx and
         b. thirdParty/javaparser-xxx
g. If editing Java code, in order to keep a consistent style, please import the projectFormatterOptions.xml from your SVN checkout into Eclipse.
   i. You may be able to simply do this as a global import.
      1. From the menu, File -> Import
      2. In the outline, General -> Preferences
      3. Click Browse to find and select the file projectFormatterOptions.xml.
      4. Click Finish.
   ii. If this doesn't work, or if you want to import these setting just for this project and not others,
      1. From the menu: Window -> Preferences
      2. In the outline on the left, Java -> Code Style -> Formatter
      3. On the right, click Import to find and select the file projectFormatterOptions.xml.
      4. Click Apply & Ok.
h. Run the behavior simulator.
   i. Convert event/behavior XML into Java behavior analysis code.
      1. From Eclipse's Package Explorer view of the Java perspective, right-click on the src -> tests -> TestEventXmlToJava.java file
      2. Select Run As -> Java Application
      3. It will read in a default XML file, and generate Java code in src -> generated.
   ii. Simulate the scenario from the generate Java code.
      1. Right-click on the src -> generated -> Main.java file
      2. Select Run As -> Java Application
      3. A simulation of the scenario and any constraint violations will be printed in the console window (maybe after a lot of debug prints).
i. You're done! You can skip step 3.

3. If not using Eclipse, you'll need to build and run it some other way, such as from the command line.
   a. Get Subversion if you don't have it.
   b. Check out from https://sscae-cm.jpl.nasa.gov/svn/cs/trunk
      i. You can do this from the command line; this example checks out into a local directory named CS

```
ii. svn checkout https://sscae-cm.jpl.nasa.gov/svn/cs/trunk CS
```

   c. If you have ant installed, you can run it or compile it from a command line (terminal on MacOS; Command Prompt or cmd.exe on Windows).
      i. In a command prompt/terminal window while in the top-level project directory (e.g., CS_Dev) where the AE.jar file (binary) lives, enter

```
1. ant run-generated
```

      a. to run the default scenario in src/xml/exampleScenario.xml,

```
2. ant run-generated -DxmlFile=myScenario.xml
```

      a. to translate myScenario.xml to Java in src/generated,
      b. to compile the generated Java, and
      c. to run the compiled scenario,

```
3. ant xml2java -DxmlFile=myScenario.xml
```

a. to translate myScenario.xml to Java,

```
4. ant compile-generated -DxmlFile=myScenario.xml
```

a. to translate myScenario.xml to Java, and
b. to compile the generated java

d. If you do not have ant installed,
    i. In a command prompt/terminal window while in the top-level project directory (e.g., CS_Dev) where the AE.jar file lives, enter

```
1. java -jar AE.jar
```

a. to translate the default src/xml/exampleScenario.xml into Java code in src/generated,

```
2. java -jar AE.jar myScenario.xml
```

a. to translate myScenario.xml to Java in src/generated,

```
3. javac -cp AE.jar -d bin src/generated/*.java
```

a. to compile the generated Java,

```
4. java -classpath AE.jar;bin generated.Main
```

a. to run the compiled scenario on Windows,

```
5. java -classpath AE.jar:bin generated.Main
```

a. to run the compiled scenario on Unix platforms (like MacOS and Linux).