How to Develop/Deploy LASAD
Last Updated 7 February 2016

Author: Kevin Loughlin, intern for Dr. Bruce McLaren of Carnegie Mellon University
For questions, issues, and feedback, please contact me via kevinloughlin@college.harvard.edu

IMPORTANT NOTES

- I developed this running Mac OS X El Capitan: 10.11.3
- LASAD can be developed in Eclipse, but I recommend compiling and deploying outside of Eclipse.
- The directory/file names and paths of this project are dependent. It is very important that you use the names and paths exactly listed in the documentation. Do not move or rename any directories/files unless explicitly told to do so.

The LASAD project is available via the GitHub repository at
`https://github.com/kevinloughlin/LASAD-CMU.git`

As a prerequisite, if you do not already have access to the project, open a terminal window, navigate to your home directory, and then enter the following command…
`git clone https://github.com/kevinloughlin/LASAD-CMU.git`
You should now have a directory called `LASAD-CMU` **on your computer.**

Once you have a clone of the repository, please follow these instructions for setting up a development environment and deploying LASAD on a server.

1. Download and unpack/install any of the following required supplemental software that is not already on your computer. These downloads will contain instructions for proper installation. I list the specific versions I used, but if you're using more recent versions, be sure to check backward compatibility.

- Apache Ant (I used 1.9.4)
  https://ant.apache.org/bindownload.cgi
- Java SDK (I used 8u45)*
  http://www.oracle.com/technetwork/java/javase/downloads/index.html
- MySQL Community Server (I used 5.6.24)
  https://dev.mysql.com/downloads/mysql/

**Be mindful that you must compile in the version of Java being run on the server from which you will deploy LASAD. LASAD is confirmed to work for both Java 7 and Java 8 with proper compilation.*

NOTE: Apache Tomcat 8.0.23 and Google Web Toolkit (GWT) 2.5.1 are included in the GitHub repository of LASAD and thus do not need to be downloaded. If you want, you can most likely use a different version of Tomcat, but LASAD needs GWT 2.5.X.

2. Open `Develop/LASAD-Client/build.xml` **and find the property name "**`target.java.version`**". Confirm that its value matches the version of Java being run on your server. "**`1.7`**" corresponds to Java 7, and "**`1.8`**" to Java 8. You can then save and close** `build.xml`**. Check all other file paths (such as Tomcat) for proper configuration as well.**

3. LASAD relies on a dedicated MySQL user that we need to create on the host machine, separate from the root user. We will also need our own database. Assuming MySQL has been successfully installed, start the MySQL server instance, and then open MySQL from the command line as the root MySQL user. If you configured the root user already, use the username and password you chose. If not, root defaults to no password for login.

4. In the MySQL terminal window as root, create the lasad user via

```
CREATE USER 'lasad' IDENTIFIED BY 'lasadpw';
REVOKE ALL PRIVILEGES ON *.* FROM lasad;
```

^This second line might not be necessary, but it doesn't hurt for security purposes.

5. Still in the MySQL terminal window as root, create the lasad database (default password is `lasadpw`**, but this can be changed and must match what is listed in** `Deploy/server.cfg`**) via**

```
CREATE DATABASE lasad;
GRANT ALL PRIVILEGES ON lasad.* TO 'lasad' IDENTIFIED BY
'lasadpw';
FLUSH PRIVILEGES;
```

6. Make sure you can log into and out of MySQL as `lasad` with your password. While logged in as `lasad` (user), show the tables you have access to, and make sure `lasad` (table) is listed with full permissions. You can then exit MySQL.

7. Open `Deploy/lasad-server/server.cfg`

8. Confirm that the database user is set as `lasad`, the database name is set as `lasad`, and the database password is set as your password. If so, the MySQL aspect of LASAD should be good to go, but leave `server.cfg` open, because you still need it for other purposes.

NOTE: You can add to (or delete from) the maps, ontologies, templates, and users you want on the server at startup by accessing `Deploy/lasad-server/conf/default/`. However, make sure that all files have corresponding data in the adjacent directories. For example, don't add a map or ontology without a corresponding template, or you'll get an error when the server starts.

9. Depending on if you want the database to be deleted and recreated each time you start the server (versus simply restored), you can set `Fresh Server` to `true` for deletion and recreation, or `false` for restoration. I recommend setting it as `true` for the first time you run the server (to allow for initialization according to your configuration), but then remember to change it to `false` before future restarts, in order to save your data.

NOTE: Steps 10-13 are ONLY necessary if you need to deploy LASAD on different ports than the defaults. LASAD is currently configured to run on Port 8099, with RMI on Port 1899, and additional Tomcat ports including 8005, 8009, and 8443. Be sure that all ports necessary for LASAD are free, including ports used in `Deploy/apache-tomcat-8.0.23/conf/server.xml`. In steps 10-13, I change the ports, just for the sake of showing you how. For this example, we'll use 8082 for Tomcat and 1399 for RMI.

10. The servlet URL in `server.cfg` should be set to (for example port `8082`) `http://localhost:8082/lasad/lasad_client/PushServlet`

11. Since we're using port `1399` for RMI, we have to change the RMI-Registry Port to `1399`. For the `Servername` property, you can use whatever you want (or leave it as the default), but take note of it. Save and exit `server.cfg`.

12. For RMI, we'll also need to change a line in `Develop/LASAD-Client/src/lasad/gwt/server/LASADGWTServiceBroker.java` that looks like this (line 92 for me).

NOTE: the following is one line of code…

```
server = (ServerInterface)
Naming.lookup("rmi://localhost:1899/LASAD-8099");
```

Change this to our example RMI port, `1399`**. We also need to make sure the** `Servername` **property in** `server.cfg` **matches the end of the RMI URL. Both are** `LASAD-8099` **by default, so we're fine for now. If you change one of the names in the future, you'll need to remember to change the other to match. Save** `LASADGWTServiceBroker.java` **and exit it.**

13. Open `Deploy/apache-tomcat-8.0.23/conf/server.xml` **and change the port away from** `8099` **in**

```
<Connector port="8099"
protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
```

to the port you'll be using. For this example, use `8082`**. You can also change other Tomcat ports in use (such as** `8005, 8009,` **and** `8443`**) from this file. Then, save and close the file.**

Here marks the return to necessary steps.

---

14. If you would like to log to PSLC DataShop, set the appropriate parameters in `Deploy/lasad-server/ds_settings.txt`**, which correspond to lines in** `MapActionProcessor.java`**. By default, logging should be set to false via the first line of** `ds_settings.txt`

15. As a last prerequisite before compilation, verify that your Java version matches the `target.java.version` **from Step 2 (which should match the Java version of your server). If it does, you can proceed to the next step.**

16. Navigate to `Develop/LASAD-Client`**. ALWAYS be sure to shut down any LASAD processes running on your compilation computer before (re)compiling (see note below item 18 on how to stop LASAD if it's running). Then, ALWAYS enter** `ant clean` **on the terminal line before (re)compiling. You should also perform these actions before pushing LASAD code to a repository in order to avoid sharing unnecessary files.**

17. Begin compilation by entering `ant` **on terminal line (3-5 minute process).**

18. Copy `Deploy/` **to your host machine. Configure the** `homeDir`**,** `tomcatDir`**,** `tomcatStartup`**, and** `tomcatShutdown` **variables in startLASAD.sh and/or stopLASAD.sh to match your file paths. Then, launch in the following order…**

(a) Start the MySQL server instance on your host

(b) Run LASAD's startup script(s)

- On Linux/Max, `cd` **to** `Deploy/` **and run** `startLASAD.sh`

- On Windows, run `run.bat` **in** `Deploy/lasad-server` **followed by** `startup.bat` **in** *`Deploy/apache-tomcat-8.0.23/bin/`*

(c) Access the webpage at (assuming you used port `8082`)
`http://[yourhostname]:8082/lasad`

NOTE: You can stop LASAD on Linux/Mac by running `Deploy/stopLASAD.sh`. *Stopping on Windows requires killing the Java process initiated by* `run.bat` *and also running* `Deploy/apache-tomcat-8.0.23/bin/shutdown.bat`.

19. Do not forget to change `Fresh Server` (in `server.cfg`) to false after you've started the server for the first time (that is, if you want to save your data upon server restart).

20. We also believe there is an unfixed memory leak if the file `Deploy/lasad-server/log/debug.log` is created (which only happens if the `Logging` property in `server.cfg` is set to `true`. Thus, for deployment, I recommend setting `Logging` to `off`. Logging to `server.log` will still occur, but `debug.log` will not be created.

TROUBLESHOOTING

- **Check file permissions and file ownership. You MIGHT have to** `chown` **all the files of** `Deploy`**. On our particular remote server, (where our directory for housing** `Deploy` **is named** `lasad-8082`**), I did** `chown -R lasad:tomcat lasad-8082`**, but I didn't have to do this on my local server.**

- **Verify that all required ports are free. Check** `server.log` **and** `catalina.out` **for error messages about ports already in use.**

- **Make sure you are compiling in the same Java version that you are running on your server.**

- **Verify you followed the launch sequence IN ORDER**
  1. **Start MySQL server instance**
  2. **Run startup script (or bat file on Windows)**
  3. **Access web page via browser**

- **Shut down LASAD, run** `ant clean`**, recompile, and restart LASAD.**

- **Confirm that you have not changed any file paths, and check that the file paths in** `build.xml`**,** `startLASAD.sh`**, and** `stopLASAD.sh` **are correct and have the necessary files/directories at the referenced locations.**

- **Reread the directions and start over with a fresh clone of the repository.**

If you still have issues after retrying the directions on a fresh clone, I can be reached via kevinloughlin@college.harvard.edu and am happy to help. Please also contact me if you find errors in these instructions.