Setting up Tomcat on Eclipse

1. **Install Java.** Download for Windows, MacOS, Linux, and Solaris from http://www.oracle.com/technetwork/java/javase/downloads/. JDK 1.7.0_21 and 1.6.0_35 are the latest versions (May 2013), but any Java 6 or 7 version will work. Servlet 3.0 containers (of which Tomcat 7 is one) require Java 6+ and will NOT work with Java 5. You want the full JDK (with compiler), not just the JRE (for running existing apps). Accept all defaults when installing.

2. **Unzip Tomcat.** Download Tomcat from http://mirror.csclub.uwaterloo.ca/apache/tomcat/tomcat-7/v7.0.40/bin/apache-tomcat-7.0.40.zip. Unzip *apache-tom-cat-7.0.40.zip* into the location of your choice. (On Windows, it is recommended to unzip all software tools here cited into a directory in the top level of the C drive, which results, for instance, in "C:\JAVA\apache-tomcat-7.0.40" for Tomcat).

   Change Tomcat Server port. When you download Tomcat from the Apache site, the port is 8080, in case you already have another server running on port 80. If you want to modify the port, so Tomcat runs on 80 instead of 8080, change port number in the file *server.xml* located at the directory "C:\apache-tomcat-7.0.40\conf". (This allows you enter URLs of the form http://localhost/... instead of http://localhost:8080/....).

3. **Install and start Eclipse.** Download from http://www.eclipse.org/downloads/. Choose "Eclipse IDE for Java EE Developers", download, and unzip (it is recommended to unzip at the aforementioned "C:\JAVA\" directory). As of early June/2013, the latest version is Eclipse Juno (4.2). (These instructions also work with Eclipse 3.6 and up. Eclipse 3.5 and earlier do not have adapters for Tomcat 7, so cannot be used.). Double click on *eclipse.exe* in the "eclipse" root folder, then click on "Workbench". It is conviniente to have a shortcut easily accessible, such as on your Desktop or Taskbar.

   Change Eclipse preferences. Go to *Window » Preferences* then *Java » Installed JREs*. Make sure that a JDK (not just JRE) is selected in this option. If not, press *Add*, and then navigate to the base JDK folder. If you develop with servlets, you probably also want to suppress unnecessary warnings about Serializable classes for servlets. Stilll in the Preferences window, go to *Java » Compiler » Errors/Warnings*, and change "Serializable class without serialVersionUID" to *Ignore*.

4. **Add Tomcat Server into Eclipse.** Click on Servers tab at bottom (if you cannot see Servers tab, add the tab via *Window » Show View » Servers*). Right-click and go to *New » Server*. Choose *Apache » Tomcat v7.0 Server*. For *Server runtime environment*, at the bottom, navigate to Tomcat 7 installation folder (e.g., C:\apache-tomcat-7.0.40). Click *Finish*.

5. **Run Tomcat using Eclipse.** Click on Servers tab at bottom. Right-click on Tomcat v7.0 Server, choose "Start". Open http://localhost/ in a browser (or http://localhost:8080/ if you did not change the port from 8080 to 80). A 404 error message that comes from Tomcat will be seen.

   Copy the ROOT (Default) Web App into Eclipse. Eclipse does not copy the default apps (ROOT, examples, etc.) when it creates a Tomcat folder inside the Eclipse's workspace. Therefore, go to "C:\JAVA\apache-tomcat-7.0.34\webapps", Right-click on the ROOT folder, and copy it. After that, go to your Eclipse workspace, go to the .metadata folder, and search for "wtpwebapps". You should find something like "your-eclipse-workspace\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps" (or ".../tmp1/wtpwebapps" if you already had another server registered in Eclipse). Go to the wtpwebapps folder, Right-click, and paste ROOT (say "Yes" if asked if you want to merge/replace folders/files). Then reload http://localhost/ (or http://

localhost:8080/ if using the unmodified version from the Tomcat download site) to see a friendly Tomcat welcome page. If you get a "port 80 is already in use" message, go to the *Windows Control Panel » Services*, and stop the other server (probably IIS).

6. **(Optional) Import and test a sample Web App.** Download test-app.zip file, save it, and import it into Eclipse. Use *File » Import...*; choose *General » Existing Projects into workspace*. Select *Archive File*, then click *Browse* and navigate to *test-app.zip*. Click on Servers tab at bottom. Right-click on Tomcat v7.0 Server, choose *Add and Remove...*, and add *test-app* project to be configured. Start Tomcat, or restart it if it is already running (click on *Servers* tab at bottom. Right-click on "Tomcat v7.0 Server", choose *Start* or *Restart*). Open http://localhost/test-app/ (or http://localhost:8080/test-app/) in a web browser. (Note that this app uses the Servlet 3.0 @WebServlet annotation to provide the URLs for the various servlets.). See the source code for details.

7. **(Optional) Create and test a new Web App.** Select *File » New » Other...* then choose *Web » Dynamic Web Project*. Make sure that the *Target runtime* is "Apache Tomcat v7.0". Create or copy all files into this new project. Deploy and test it as above.

8. **Import and test the SenecaBBB Web App from the Git repository.** Go to *File » Import...*, choose *Git » Project from Git*, and click *Next*. Select *URI* as repository source then click *Next* again. Insert https://github.com/SenecaCDOT-BigBlueButton/SenecaBBB.git for *URI*, and authenticate using your GitHub account by inserting *User* and *Password*. Click *Next*. Make sure that the *master* branch is selected and click *Next*. Choose a Local destination on *Directory* by clicking on *Browse*, (a recommended choise is to select your Eclipse workspace). Click *Next*. Chose *Import an existing project* and click *Next*. Make sure that the "SenecaBBB" project is selected, and click *Finish*. Click on *Servers* tab at bottom, Right-click on "Tomcat v7.0 Server", choose *Add and Remove...*, and add *SenecaBBB* project to be configured. Click *Finish*. Start Tomcat, or restart it if it is already running (click on *Servers* tab at bottom. Right-click on "Tomcat v7.0 Server", choose *Start* or *Restart*). Open http://localhost/SenecaBBB/ (or http://localhost:8080/SenecaBBB/) in a web browser.

Setting up DB on Ubuntu 12.04

1. **Run the following commands to install MySQL.** (Note the password you set when installing mysql server).

   sudo apt-get install mysql-server
   sudo apt-get install mysql-client
   sudo apt-get install libmysql-java

2. **Edit your mysql configuration file.**

   sudo gedit /etc/mysql/my.cnf
   port = 3309 (change all instances of port)
   bind-address = local VM ip

3. **Restart the MySQL service.**

   sudo service mysql restart

4. **Connect to mysql-server.** (Use the password you set when installing MySQL server).

   mysql -u root -p

5. **Create the database 'db' then exit.**

   create database db;

6. **Download the MySQL script files and run them.** (Open another terminal).

   mysql -u root -p db < bbb_db_manual.sql
   mysql -u root -p db < bbb_db_fn_nextval.sql
   mysql -u root -p db < bbb_db_sample_data.sql

7. **Connect to mysql-server again.**

   mysql -u root -p

8. **Create a user for remote access to DB and grant privileges.**

   create user 'senecaBBB'@'%' identified by 'db';
   grant all on db. to 'senecaBBB'@'%' IDENTIFIED by 'db';*

9. **Connect to the MySQL database using Eclipse.** Click on the *Database Source Explorer* tab (if you cannot see the *Database Source Explorer* tab, add it via *Window » Show View » Database Source Explorer*). Right-click on *Database Connections » New...*, and choose MySQL. Name it "SenecaBBB". Click *Next*. Use the information below.

   URL: *jdbc:mysql://localhost:3309/db* /* If it is not working, use your Virtual Machine IP instead of "localhost" */
   Username: *senecaBBB*
   Password: *db*

10. **Connect to the MySQL database using NetBeans.** Click on the *Services* tab then Right-click, go to *Databases » New Connection*, and choose MySQL. Use the information below.

    Host: *127.0.0.1* /* If it is not working, use your Virtual Machine IP */
    Port: *3309*
    Username: *senecaBBB*
    Password: *db*