Development Environment Setup Walkthrough

Information

This is a walkthrough for setting up a Sakai development environment on a laptop or desktop. These instructions will also work for setting up a development server though I suggest slightly beefier settings for memory. This is currently tailored for Sakai 2.9+. Once you have a Sakai development environment setup you should be able to program your own Sakai tools or modify existing tools.

Notes for developers using older versions of Sakai (2.8.x-)

- Developers who are using Sakai 2.8.x through 2.5.x should use the Development Environment Setup Walkthrough (2.5.x-2.8.x)
- Developers who are using Sakai 2.4.x or lower should use the Development Environment Setup Walkthrough (2.2.x-2.4.x)

Notes for windows users

- change the "/" to "\" for all directory paths except those specified in the maven build.properties file
- add the drive letter (C:) to your paths
- use %VARIABLE% instead of $VARIABLE with environment variables
- set your environment variables in Control Panel -> System -> Advanced -> Environment Variables

All of sakai and related programs should be installed in an opt directory which you will need full write access to, put this directory anywhere you want, this document will assume you have placed it in your root directory

1. ## Install/Verify you have Java 1.7
   a. To check: Run **java -version** on the command line
   b. If not correct, download the J2SE SDK (make sure you get the JDK and not JRE) from:
      - http://www.oracle.com/technetwork/java/javase/downloads/index.html
      - Mac OS X users (10.6+) will have Java 1.7 by default
   c. Install it (the SDK) to /opt/java
      - **Note:** Install the JRE to a different directory (probably the default, especially under windows) or you will have problems
   d. Set environment variable: JAVA_HOME=/opt/java
      - Mac users: JAVA_HOME=/Library/Java/Home
      - Windows users default: JAVA_HOME=C:\j2sdkXXX (where "XXX" is the version - for example "j2sdk1.7.0_11")
   e. Add $JAVA_HOME/bin to PATH
   f. Set JAVA_OPTS for tomcat in the tomcat/bin/setenv file (see Install Tomcat 7 / 8)

      ```
      export JAVA_OPTS="-server -Xmx1028m -XX:MaxPermSize=320m -Djava.awt.headless=true -Dcom.sun.
      management.jmxremote -Dsun.lang.ClassLoader.allowArraySyntax=true"
      ```

      - These settings are for developers
      - More info on JVM server settings: Sakai Admin Guide - JVM Tuning
2. ## Install/Verify you have MySQL 5.1 or later
   - MySQL - http://www.mysql.com
     - You should download the **MySQL Community Server**
     - NOTE: MySQL 5.5.x works but there are minor issues (mostly related to creating some profile 2 tables). If you already have 5.5 then you probably do not need to downgrade

   a. To check: Run **mysql --help** on the command line
   b. If not, download MySQL from:
      - http://dev.mysql.com/downloads/mysql/5.1.html
        - Or if you want 5.5 or the current release: http://dev.mysql.com/downloads/mysql/
   c. Custom install to /opt/mysql (options vary slightly based on operating system)
      - Linux users should install MySQL using a package or binaries if possible

      i. Choose standard configuration
      ii. Install as a windows service (Windows only)
      iii. Launch automatically (recommended)
         - If you choose not to do this, make sure that you startup MySQL each time before you try to run Sakai

      iv. Include the bin directory in Path
      v. Don't use an anonymous account
      vi. Set your root password to "mysqlpwd1"
   d. Update the mysql config file
      - For linux/OSX this is going to be: **/etc/my.cnf**
      - Windows: this file will be called **my.ini** and located in the dir you installed mysql into

- MySQL provides a number of preconfigured option files that can be used as a starting point for configuration. Look for my-small.cnf, my-medium.cnf, my-large.cnf and my-huge.cnf.

i. Add the following to the [mysqld] section

```
default-storage-engine = InnoDB
innodb_file_per_table
character-set-server=utf8
collation-server=utf8_general_ci
lower_case_table_names = 1
```

ii. OPTIONAL for 5.5: Add this to enable logging

```
log_output=FILE
log=/tmp/mysql-query.log
slow-query-log=1
long_query_time=1
slow_query_log_file=/tmp/mysql-slow-query.log
expire_logs_days=5
```

3. Setup Sakai DB (schema) and user

a. Create the sakai database and sakai user (password=ironchef)
b. Run the following from a command line prompt:

```
mysql -u root -p
```

- **Note:** You can also do the following if you prefer to be prompted for the password: mysql -uroot -p

c. Then run these commands from the mysql prompt (one command per line):

```
create database sakai default character set utf8;
grant all privileges on sakai.* to 'sakai'@'localhost' identified by 'ironchef';
grant all privileges on sakai.* to 'sakai'@'127.0.0.1' identified by 'ironchef';
flush privileges;
quit
```

d. [Optional] Download and install mysql query browser
   i. http://dev.mysql.com/downloads/query-browser/1.1.html
   ii. You do not have to do anything with query browser now

4. Download and setup Maven 2.2.x+ stable package

- Maven - http://maven.apache.org/

a. Download Maven 2.2.X (minimum of 2.2.1) - http://maven.apache.org/download.html
   - NOTE: some support exists for Maven 3 but there are a few things that do not work as of 6 Sep 2011
b. Extract to /opt (should create apache-maven-2.X.X folder)
c. Set environment variable: MAVEN2_HOME=/opt/apache-maven-2.X.X
d. Add $MAVEN2_HOME/bin to PATH\
e. Set the MAVEN_OPTS environment variable (in linux/OSX this is typically done in ~/.bash_profile)

```
export MAVEN_OPTS='-Xms128m -Xmx796m -XX:PermSize=64m -XX:MaxPermSize=172m'
```

- Adding **-XX:+AggressiveOpts** may speed up your builds depending on your OS, JVM, and system hardware

5. Install/Verify you have Subversion

- Subversion - http://subversion.tigris.org/

a. To check: Run **svn --version** on the command line
   - http://subversion.apache.org/packages.html
   - ***Get the subversion binaries and not the source, if possible***
   - If there are no binaries for your platform, get the source and use the configuration options --with-ssl and --with-libs
b. Extract to /opt (should create subversion-1.2.3)
   - Windows users will want to rename the extracted directory
   - Unix users will probably want to use a package for their flavor
c. Set environment variable: SUBVERSION_HOME=*/opt/subversion-1.2.3*
d. Add $SUBVERSION_HOME/bin to PATH

6. Download and setup tomcat 7.0.21+ (stable only)

- Apache Tomcat - http://tomcat.apache.org/

- **Note:** Always do a fresh install of Tomcat (please note that Tomcat has a bug. Please use Tomcat 7.0.65 or Tomcat 7.0.68, when released. Addendum 30January2017 - appears to work with 7.0.75 as of this writing. Is known not to work with 7.0.72 through 7.0.74
  ~~SAK-31912~~ - FacesContext already released in JSF tools with Tomcat 7.0.72+ RESOLVED)
- **Note:** Windows users should ensure that there are no spaces in the complete tomcat path as this causes errors with JSF tools in Sakai
  GOOD: C:\opt\tomcat\, C:\sakaistuff\installs\tomcat\
  BAD: C:\program files\tomcat\, C:\opt\apache tomcat 7.0.1\

a. Download Tomcat 7 if you're using Sakai 10- http://tomcat.apache.org/download-70.cgi
   Download Tomcat 8 if you're using Sakai 11 - https://tomcat.apache.org/download-80.cgi
   (The directions below are mostly the same for 7 or 8, but the setenv.sh is a little different)

   - Windows users should get the zip file instead of installing a service
     It makes viewing the tomcat logs easier and it is easier to configure

     If you're running the tag of Sakai 2.9.1 or earlier you need to be running Tomcat 7.0.21 or earlier. If you go any higher Webdav and Sitestats will have errors and not work. A better solution is to upgrade to the latest version of Sakai and Tomcat.

     ~~SAK-28531~~ - Sitestats does not load with Tomcat 7.0.35 CLOSED

     ~~SAK-23156~~ - IncompatibleClassChangeError from WebDAV LOCK operations giving 500 response code CLOSED

b. Extract to /opt (symlink the apache-tomcat-7.0.x directory to tomcat after extracting)
   - Example (assuming you have saved the file as /opt/apache-tomcat-7.0.x.tar.gz)

```
cd /opt
tar xzvf apache-tomcat-7.0.x.tar.gz
ln -nsf apache-tomcat-7.0.x tomcat
```

   - Windows users should just rename the directory since they cannot symlink

c. Modify conf/server.xml for international character support
   i. Add URIEncoding="UTF-8" to the Connector element
      - <Connector port="8080" URIEncoding="UTF-8" ...

d. Set environment variable: CATALINA_HOME=/opt/tomcat
e. Add $CATALINA_HOME/bin to PATH
f. Setup the SETENV file in the tomcat/bin directory with JAVA_OPTS (from Install Java 1.7+)

   Populate Database With Demo Data

   Instead of starting with an empty database you can add the flag -Dsakai.demo=true in addition to the others shown here. During your first Tomcat launch it will populate your database with test data (students, courses, etc). Once the data has been created remove this flag for subsequent Tomcat launches. Be advised that this flag will not trigger on a database with data already in it so you must make this choice during the initial setup. It should also be noted that without this flag one will not be able to create any course sites in Sakai unless Academic Term, Subject, Course, and Section data is manually added to the "cm_" tables in the database.

   i. Mac/Linux: Create a file called **setenv.sh** with the following (alternately, you can put this into your **.bashrc** file so they're automatically executed):

   Tomcat 7 Mac/Linux

```
export JAVA_OPTS="-server -Xmx1028m -XX:MaxPermSize=320m -Dorg.apache.jasper.compiler.
Parser.STRICT_QUOTE_ESCAPING=false -Djava.awt.headless=true -Dcom.sun.management.jmxremote"
```

   Tomcat 8 Mac/Linux

```
export JAVA_OPTS="-server -Xmx1028m -XX:MaxMetaspaceSize=512m -Dorg.apache.jasper.compiler.
Parser.STRICT_QUOTE_ESCAPING=false -Djava.awt.headless=true -Dcom.sun.management.jmxremote"
```

   ii. Windows(PC): Create a file called **setenv.bat** with the following:

Tomcat 7 Windows

```
set JAVA_OPTS=-server -Xmx1028m -XX:MaxPermSize=320m  -Dorg.apache.jasper.compiler.Parser.
STRICT_QUOTE_ESCAPING=false -Djava.awt.headless=true -Dcom.sun.management.jmxremote
```

Tomcat 8 Windows

```
set JAVA_OPTS=-server -Xmx1028m -XX:MaxMetaspaceSize=512m -Dorg.apache.jasper.compiler.
Parser.STRICT_QUOTE_ESCAPING=false -Djava.awt.headless=true -Dcom.sun.management.jmxremote
```

g. [OPTIONAL] Delete the default webapps from the webapps dir

```
rm -rf webapps/*
```

Configure tomcat 7/8 to use the old tomcat 5.5 classloader dirs, this is **not** needed for the master branch of Sakai and Sakai 11.

a. Edit **conf/catalina.properties**
   i. Add the following to the line that begins with "common.loader=..."
      - Tomcat 7

        ```
        ,${catalina.base}/common/classes/,${catalina.base}/common/lib/*.jar
        ```

      - Tomcat 8

        ```
        ,"${catalina.base}/common/classes/","${catalina.base}/common/lib/*.jar"
        ```

   ii. Add the following to the line that begins with "server.loader=..."

      ```
      ${catalina.base}/server/classes/,${catalina.base}/server/lib/*.jar
      ```

   iii. Add the following to the line that begins with "shared.loader=..."

      ```
      ${catalina.base}/shared/classes/,${catalina.base}/shared/lib/*.jar
      ```

b. Create the directories

   ```
   mkdir -p shared/classes shared/lib common/classes common/lib server/classes server/lib
   ```

Improve startup speed

You can improve startup speed under both Tomcat 7 and Tomcat 8 significantly.

Tomcat 7

Edit the file conf/catalina.properties and add the property to the bottom

org.apache.catalina.startup.ContextConfig.jarsToSkip=*.jar

Tomcat 8

Edit the file conf/context.xml and add this JarScanFilter block to the context

```
<Context>
    <JarScanner>
        <JarScanFilter defaultPluggabilityScan="false" />
    </JarScanner>
</Context>
```

7. Download and setup MySQL Connector/J stable

- Mysql Connector J - http://www.mysql.com/downloads/connector/j/

a. Download connector for your version of MySQL
    i. If running mySQL 5.0-5.1
       Download version 5.0.8+ from http://dev.mysql.com/downloads/connector/j/5.0.html
    ii. If running MySQL 5.1+
       Download version 5.1+ from http://www.mysql.com/downloads/connector/j/
    - NOTE: the version numbers do not directly align despite the way they appear to here

b. Extract the file to /opt
c. Copy mysql-connector-java-<version>-bin.jar to $CATALINA_HOME/common/lib
d. Delete the extracted folder

8. Use Subversion to download sakai code (Sakai 10 and earlier) or use Git/Github to download sakai code (Sakai 11 and later)

a. Change to the /opt directory
b. Checkout source from trunk (absolute latest stuff) or a release branch or tag:
    i. Trunk: svn checkout https://source.sakaiproject.org/svn/sakai/trunk/ sakai-trunk
    ii. 2.9.x Branch: svn checkout https://source.sakaiproject.org/svn/sakai/branches/sakai-2.9.x/

Alternative Sakai 2.9.x repository

Should you run into errors related to missing dependencies when building source from the above repository with Maven (Step 11 below), attempt to checkout from https://source.sakaiproject.org/svn/sakai/branches/sakai-2.9.x-all instead

    iii. If you want a version other than these, browse the tags and branches at https://source.sakaiproject.org/svn/sakai/

c. Checkout will take about 5-10 minutes
d. The directory created by subversion will be referred to as <sakai-src> directory (this could be whatever you want; above examples use 'sakai-trunk' and 'sakai-2.9.x')

9. Setup sakai.properties file

a. Create **sakai**directory in $CATALINA_HOME
    - $CATALINA_HOME should be /opt/tomcat if you have been following these instructions

b. Copy the default config from default.sakai.properties to $CATALINA_HOME/sakai/sakai.properties
    - The default configuration template is at https://source.sakaiproject.org/svn/config/trunk/configuration/bundles/src/bundle/org/sakaiproject/config/bundle/default.sakai.properties.

c. Edit the sakai.properties file for a MySQL dev environment. Starting at the section marked **# DATABASE**:
    i. Set BaseDataSource username
        1. Set username@javax.sql.BaseDataSource=sakai
    ii. Set BaseDataSource password
        1. Set password@javax.sql.BaseDataSource=ironchef
    iii. Find the section: **# HSQLDB settings** and verify following lines are commented out

```
# HSQLDB settings
(DEFAULT)

#vendor@org.sakaiproject.db.api.SqlService=hsqldb
#driverClassName@javax.sql.BaseDataSource=org.hsqldb.jdbcDriver
#hibernate.dialect=org.hibernate.dialect.HSQLDialect
#validationQuery@javax.sql.BaseDataSource=select 1 from INFORMATION_SCHEMA.SYSTEM_USERS
# Two hsqldb storage options: first for in-memory (no persistence between runs), second for
disk based.
#url@javax.sql.BaseDataSource=jdbc:hsqldb:mem:sakai
#url@javax.sql.BaseDataSource=jdbc:hsqldb:file:${sakai.home}db/sakai.db
```

    iv. Find the section: **# MySQL settings** and uncomment the 6+ lines in it

```
# MySQL settings
vendor@org.sakaiproject.db.api.SqlService=mysql
driverClassName@javax.sql.BaseDataSource=com.mysql.jdbc.Driver
hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
url@javax.sql.BaseDataSource=jdbc:mysql://127.0.0.1:3306/sakai?
useUnicode=true&characterEncoding=UTF-8
validationQuery@javax.sql.BaseDataSource=select 1 from DUAL
defaultTransactionIsolationString@javax.sql.BaseDataSource=TRANSACTION_READ_COMMITTED
```

d. Save the changes to the sakai.properties file

10. Create maven settings.xml file

a. Create a new xml file in your user home directory in the **.m2** directory called **settings.xml**
   - **Note:** This is probably c:\documents and settings\<username> in Windows
   - Download a sample settings.xml file (setup for windows users)

b. Add the following lines:

```
<settings xmlns="http://maven.apache.org/POM/4.0.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                       http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <profiles>
    <profile>
      <id>tomcat5x</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <properties>
        <appserver.id>tomcat5x</appserver.id>
        <appserver.home>/opt/tomcat</appserver.home>
        <maven.tomcat.home>/opt/tomcat</maven.tomcat.home>
        <sakai.appserver.home>/opt/tomcat</sakai.appserver.home>
        <surefire.reportFormat>plain</surefire.reportFormat>
        <surefire.useFile>false</surefire.useFile>
      </properties>
    </profile>
  </profiles>
</settings>
```

   - **Note:** Unfortunately, Sakai does not use the standard appserver.home so you have to specify a sakai.appserver.home
   - **Note:** The sakai.appserver.home must be *C:\opt\tomcat* for windows users
   - **Note:** Do not include trailing / or \ slashes in the directory paths

- (optional) You can specify the tomcat home to be an environment variable like so:

```
      <maven.tomcat.home>${env.CATALINA_HOME}</maven.tomcat.home>
```

- (optional) Users who use a network proxy need to add the following to their file

```
...
  <proxies>
     <proxy>
        <active>true</active>
        <protocol>http</protocol>
        <host>www.your.proxy.host</host>
        <port>80</port>
        <username>your_username</username>
        <password>your_password</password>
        <nonProxyHosts>localhost</nonProxyHosts>
     </proxy>
  </proxies>
</settings>
```

   - If you do not use a username or password for your proxy just leave out those 2 lines

- You only need the nonProxyHosts option if you have a local maven repo that does not require the proxy to be accessed
- **Note:** Maven 2 doesn't support Microsoft's NTLN authentification scheme. If you connect to a proxy like ISA you will need to use a tool, such as http://ntlmaps.sourceforge.net/ to proxy your traffic

11. Use Maven to build Sakai

a. Open a command line shell
b. Change directory to /opt/<sakai-src>
c. Execute **mvn clean install** to build Sakai using maven
    - Note: The build will take an extra 5-10 minutes to download dependencies the first time
d. Execute **mvn sakai:deploy** to deploy Sakai to your tomcat using maven
e. Partial builds are supported by the maven2 build system
   You can now do a "mvn clean install sakai:deploy" from any subdirectory and build just that code
f. Once you have downloaded the jars you can run maven off-line with **mvn -o clean install sakai:deploy**

12. Start Tomcat and check to make sure Sakai runs

a. Start tomcat using $CATALINA_HOME/bin/startup

> **Viewing an active launch log in Linux**
>
> If you are in Linux you can view the activity log as its generated by using "# ./catalina.sh run" from Tomcat's bin directory instead of startup.sh. This is helpful for viewing potential errors as they happen or to know if Tomcat has locked-up during a launch. A launch will have completed when "INFO: Server startup in xxxx ms" is displayed in the terminal.

b. Allow 1 minute+ for tomcat to start up
c. Open http://localhost:8080/ to verify tomcat is running
d. Open http://localhost:8080/portal to verify sakai is running
e. Login to sakai as username:admin, password:admin
f. Shutdown Tomcat using $CATALINA_HOME/bin/shutdown

13. Install Eclipse 3.2+ stable

- Eclipse - http://www.eclipse.org/
    - Note: If you already have eclipse 3.1.x installed, your best best is to start over with a fresh install rather than attempting to upgrade
    - Note: If you have not installed eclipse already, you should download the WebTools Platform: All-in-one package for Eclipse 3.3+ which includes Eclipse and all of the necessary webtools packages, this is much faster and easier than downloading them seperately, you can skip the WebTools step if you do this, here are instructions to install the Eclipse WebTools

a. Download the newest stable version http://www.eclipse.org/
b. Extract the downloaded file to /opt
    - Windows users should extract to c:\
    - Mac users should use the installer and put it in Applications
c. Run eclipse to verify it works (**/opt/eclipse/eclipse**)
    - Windows users: c:\eclipse\eclipse.exe to run eclipse
    - Note: If it does not work, there is probably a problem with your java install
d. Set the memory settings for Eclipse
    - The default memory settings for Eclipse are much too low to handle the number of webapps in a full Sakai installation

    i. Shutdown eclipse if it is running
    ii. Edit the eclipse.ini file in the directory you extracted/installed eclipse
    iii. Change the settings from something like this:

    ```
    -vmargs
    -Xms40m
    -Xmx256m
    ```

    to something like this (leave any params that do not match these alone):

    ```
    --launcher.XXMaxPermSize
    256M
    -vmargs
    -Xms128m
    -Xmx1024m
    -XX:+UseParallelGC
    ```

    - Windows users should not use notepad to edit this file, use wordpad or edit (command line)
    - Mac users will need to take additional steps to edit the eclipse.ini file

    1. Control-click on the Eclipse Application icon and select **Show Package Contents**
    2. Double-click on the **Contents** folder
    3. Double-click on the **MacOS** folder, the eclipse.ini file should be here
    4. Full path: eclipse/Eclipse.app/Contents/MacOS/eclipse.ini
e. Set the JVM
    - The default JVM will be the one in the JRE, things will work better if you change this to the one in the J2SE

            i. Startup eclipse if it is not running
            ii. Select **Window -> Preferences -> Java -> Installed JREs**
            iii. Select the current installed JRE
            iv. Click the **Edit** button
            v. Click **Browse** and navigate to the home directory of your JAVA installation (e.g. /opt/java)
            vi. Click **OK** to save and then OK to Finish
                - Running Eclipse with a different JVM from the default
                  Add `-vm /opt/java/bin/java` after the eclipse executable
                - You must include the full path to the java executable file
    - More Eclipse tips here: Eclipse notes

14. Add subclipse to Eclipse

- Subclipse - http://subclipse.tigris.org/

a. Startup eclipse
b. Select **Help -> Software Updates**
c. Select **Available Software**, click Next
d. Click **Add Site**
e. Enter http://subclipse.tigris.org/update_1.6.x for the URL
f. Click **Finish**
g. Check the Subclipse box and click Next
    - Ignore the integrations unless you have a need for them, they can stop the install from working
h. Accept terms, click Next, click **Finish**, click **Install All**
i. Restart the workbench when asked

- Add bin and target to global svn ignore in Eclipse (Optional)
    - This will keep you from committing bin and target directory files when you add projects to svn
    a. Click on **Window -> Preferences**
    b. Select **Team -> Ignored Resources**
    c. Click on **Add Pattern** and enter "bin"
    d. Click on **Add Pattern** and enter "target"
    e. Click on **Add Pattern** and enter "m2-target"
    f. Click on **Apply** and then **OK**

15. Add Maven Eclipse Plugin to Eclipse

- Maven Plugin for Eclipse - http://m2eclipse.codehaus.org/

a. Startup eclipse
b. Select **Help -> Software Updates**
c. Select **Available Software**, click Next
d. Click **Add Site**
e. Enter http://m2eclipse.sonatype.org/sites/m2e for the URL
f. Click **Finish**
g. Expand the **Maven Integration for Eclipse** update site
h. Check the **Maven Integration** box and click Next
    - Ignore the optional parts unless you have a need for them
i. Accept terms, click Next, click **Finish**, click **Install All**
j. Restart the workbench when asked

16. Add Lombok plugin to Eclipse [OPTIONAL]

- Lombok Library - http://projectlombok.org/download.html

a. Shutdown eclipse
b. Follow the steps on the page linked above
c. Restart eclipse

- NOTE: this is only required for certain parts of the code that use the lombok library (a limited set)

17. Import Sakai source code into Eclipse

a. [OPTIONAL] Run the following maven commands in your sakai-src root folder (e.g. /opt/cafe-2.5.x)
   This will clean out any eclipse files that are in there already:

```
mvn eclipse:clean
```

b. [OPTIONAL] Generate the eclipse files using maven (this is not required and not recommended)
   This will generate valid eclipse .project and .classpath files from the maven build

```
mvn eclipse:eclipse
```

   - Special Case: If your workspace is in a different location, I've found that this option is needed passed to maven, otherwise the dependencies get incorrect. *-Declipse.workspace=..*
        - Passing the full path to the local workspace or current directory might also work
   - If you are using the Maven Plugin for Eclipse then you should run this instead:

```
mvn eclipse:m2eclipse
```

c. Startup eclipse if it is not running
d. Create a new workspace for Sakai
    i. Click on File -> Switch Workspace
    ii. Enter "WS-Sakai" in place of the default "workspace" directory
    iii. Click **OK** (eclipse will restart)
        - The following steps should be done in the **WS-Sakai** workspace
e. Add Maven Repository libraries to classpath
    - *NOTE:* This is for backwards compatibility with projects using the M2_REPO variable so it can be skipped
    i. Select Window -> Preferences -> Java -> Build Path -> Classpath Variables
    ii. Add MAVEN_REPO classpath variable with the path to the local maven repository (Sakai 2.4 or lower)
        - The path should be: $USER_HOME/.maven/repository
    iii. Add M2_REPO classpath variable with the path to the local maven repository (Sakai 2.5 or higher)
        - The path should be: $USER_HOME/.m2/repository
f. Switch to the Java perspective (Window -> Open Perspective -> Java)
    - Make sure you are in the package explorer
g. Turn off Automatic builds
    - Select Project and uncheck **Build automatically**
h. Select File -> Import -> Existing Projects into Workspace
    i. You can either import the entire sakai source tree (e.g. /opt/sakai-2.9.x) or selectively choose specific sub-projects you'll be working on (e.g. /opt/sakai-2.9.x/announcement)
    ii. Click Finish to import the selected projects
        - This will take awhile, probably 5+ minutes
    iii. Clean all Sakai projects
        1. Select Project -> Clean
        2. Select **Clean all projects**
        3. Check **Start a build immediately**
        4. Click **OK**
    iv. Build All (Project -> Build All)
        - There will be hundreds of warnings, do not worry about these
i. Turn back on Automatic builds
    - Select Project and check **Build automatically**
    - Select project and then use the **down triangle** menu to add Filters (target, m2-target)