Sakai 19 Install Guide (Source)

1.0 Install Prerequisites

To begin with the following items must be installed (you may skip any you all ready have installed.

1. Java 1.8
2. Git
3. Maven 3.0 or later
4. Tomcat 8

Sakai 12.0 was QA tested with Java 1.8.0_111, maven-3-2.5 (though maven-3.3.9 should also work), Apache Tomcat 8.5.28

1.1 Verify/Install Java 1.8

Oracle's Sun Java SE 8, a.k.a Java 1.8, is the required version to use with the Sakai. Certain files, such as *.jsp and *.jws, require compilation so downloading and attempting to use only the run time environment (JRE 8.0) will not suffice.

To confirm that Java is both installed on your system and is the correct version for Sakai, run `java -version` from the command line:

```
java -version
```

If Java is installed, basic version and build information will be displayed. Example output:

```
java version "1.8.0_111"
Java(TM) SE Runtime Environment (build 1.8.0_31-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)
```

1.1.1 Set Java environment variables

Several environment variables and related properties must be set for Java. For UNIX operating systems one typically modifies a startup file like `~/.bash_login` to set and export shell variables while Mac users typically set and export environment variables in `.bash_profile`. For Windows, go to `Start -> Control Panel -> System -> Advanced -> Environment Variables` and set JAVA_HOMI via the GUI.

Set the JAVA_HOME environment variable to point to the base directory of your Java installation and add Java's `/bin` directory to the PATH environment variable.

If the variable JRE_HOME is already set or if you want to use a particular JRE if you have more than one JRE installed on your machine then you'll want to set the JRE_HOME variable as well. JRE_HOME is what Apache Tomcat uses when it starts up, but it defaults to use JAVA_HOME if JRE_HOME is not set. In most cases, setting JAVA_HOME should cover both cases sufficiently.

Variable	Unix	Mac	Windows
JAVA_HOME	`export JAVA_HOME=/usr/java/java-current`	`export JAVA_HOME=/Library/Java/Home`	`JAVA_HOME=C:\jdk1.8.0_31`
PATH	`export PATH=$PATH:$JAVA_HOME/bin`	`export PATH=$PATH:$JAVA_HOME/bin`	`;C:\jdk1.8.0_31\bin`

Windows: append the string to the end of the `Path` system variable

Set JAVA_OPTS

The default Java virtual machine (JVM) settings are insufficient for an application of Sakai's size. As a result several JVM parameters must be increased for Sakai to run, while others may need to be adjusted for optimal performance.

We recommend that you define these settings in Tomcat's `/bin` directory in a file named `setenv.sh` (Unix/Mac) or `setenv.bat` (Windows). See the "Install Tomcat 8" section below for more details.

Specify a Language and Locale (optional)

You can define the default language/locale when starting Sakai by setting the system properties `-Duser.language` and `-Duser.region`. For information on supported languages see the release notes or visit the i18N Work Group space.

```
-Duser.language=pt
-Duser.region=PT
```

In the case your locale is not fully supported in Java (as it happens with Basque or Mongolian languages) you should read this information:
Endorsed I18n Project

Specify an HTTP Proxy (optional)

In environments where local network policy or firewalls require use of an upstream HTTP proxy/cache, Sakai needs to be configured accordingly. Otherwise components or services which use HTTP requests, such as the `BasicNewsService` for RSS feeds in the News tool, cannot retrieve data from the target URLs. This can be fixed with the following `JAVA_OPTS` arguments:

```
-Dhttp.proxyHost=cache.some.domain
-Dhttp.proxyPort=8080
```

1.2 Install Git and/or local Github client

The Sakai Community uses Github as its source control management (SCM) system. Use a reasonably current version of Git to get the source. Alternatively, you can obtain binary packages for Mac/Linux and Windows (Linux is more popular and gets the preponderance of QA testing, by far). Extract the distribution archive into your installation directory.

https://github.com/sakaiproject/sakai

1.3 Install Maven 3

The Apache Maven project management framework provides Sakai with "a set of build standards, artifact repository model and a software engine that manages and describes projects" (Better Builds, p. 22). As part of the installation process you will use Maven as a build tool in order to compile, test and deploy Sakai to a servlet container such as Apache Tomcat.

It is recommended to use Maven 3.2.5 through 3.3.9.

You can download Maven at

http://maven.apache.org/download.html

Extract the distribution archive into your installation directory of choice, e.g. `/usr/local/apache-maven-3.2.2`. Confirm that you have installed the correct version of Maven and can start it by issuing `mvn --version` from the terminal. At this point your environment is prepared to build and deploy the Sakai source code.

```
mvn --version
```

```
Apache Maven 3.2.5 (45f7c06d68e745d05611f7fd14efb6594181933e; 2014-06-17T09:51:42-04:00)
Maven home: /usr/local/apache-maven-3.2.5
Java version: 1.8.0_111, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-8-openjdk-amd64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "3.13.0-30-generic", arch: "amd64", family: "unix"
```

1.3.1 Set Maven environment variables

A number of environment variables must be set for optimal Maven performance. For UNIX operating systems one typically modifies a startup file like ~/.bash_login to set and export shell variables while Mac users typically set and export environment variables in `.bash_profile`. For Windows, go to `Start -> Control Panel -> System -> Advanced -> Environment Variables` and set your Maven environment variables via the GUI.

Set the `MAVEN_HOME` environment variable to point to the base directory of your Maven installation and add the Maven `/bin` directory to your `PATH` variable:

Variable	Unix/Mac	Windows
MAVEN_HOME	`export MAVEN_HOME=/usr/local/apache-maven-3.2.2`	`set MAVEN_HOME=C:\apache-maven-3.2.2`
PATH	`export PATH=$PATH:$MAVEN_HOME/bin`	`;C:\apache-maven-3.2.2\bin`

⚠ Windows: append string to the end of the `Path` system variable

MAVEN_OPTS

Maven does not read `JAVA_OPTS` on start up, resulting occasionally in "Out of Memory" errors when building Sakai. To assure sufficient memory allocation during builds, you should add a `MAVEN_OPTS` environment variable as defined below. For UNIX operating systems one typically modifies a startup file like ~/.bash_login to set and export shell variables while Mac users typically set and export environment variables in `.bash_profile`. For Windows, go to `Start -> Control Panel -> System -> Advanced -> Environment Variables` and set `MAVEN_OPTS` via the GUI.

```
export MAVEN_OPTS='-Xms512m -Xmx1024m'
```

1.4 Install Tomcat 8

- Apache Tomcat - http://tomcat.apache.org/
- **Note:** Always do a fresh install of Tomcat 8
- **Note**: For Sakai versions prior to 11.4 please use Tomcat 8.0.34 or earlier. Later versions of Tomcat may cause problems.
  example: SAK-31608 - Getting issue details... STATUS
- **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 8.0.43\

1. Download Tomcat 8 - https://tomcat.apache.org/download-80.cgi
    - 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
2. Extract to /opt (symlink the apache-tomcat-8.0.x directory to tomcat after extracting)
    - Example (assuming you have saved the file as /opt/apache-tomcat-8.0.x.tar.gz)

    ```
    cd /opt
    tar xzvf apache-tomcat-8.0.x.tar.gz
    ln -nsf apache-tomcat-8.0.x tomcat
    ```

    - Windows users should either rename the directory or, if comfortable, create a directory junction using an elevated cmd prompt:

```
mklink /J C:\apache-tomcat-8.0.x C:\tomcat
```

3. Modify conf/server.xml for international character support
   a. Add URIEncoding="UTF-8" to the Connector element
      - <Connector port="8080" URIEncoding="UTF-8" ...
4. Set environment variable: CATALINA_HOME=/opt/tomcat
5. Add $CATALINA_HOME/bin to PATH
6. Setup the SETENV file in the tomcat/bin directory with JAVA_OPTS (from Install Java 1.8).

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.

We recommend using -Djava.util.Arrays.useLegacyMergeSort=true until this issue can be resolved - SAK-31707 - Getting issue details... STATUS

   a. 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 8 Mac/Linux

```
export JAVA_OPTS="-server -d64 -Xms1g -Xmx2g -Djava.awt.headless=true -XX:+UseCompressedOops -XX:
+UseConcMarkSweepGC -XX:+DisableExplicitGC"
JAVA_OPTS="$JAVA_OPTS -Dhttp.agent=Sakai"
JAVA_OPTS="$JAVA_OPTS -Dorg.apache.jasper.compiler.Parser.STRICT_QUOTE_ESCAPING=false"
JAVA_OPTS="$JAVA_OPTS -Dsakai.security=$CATALINA_HOME/sakai/"
JAVA_OPTS="$JAVA_OPTS -Duser.timezone=US/Eastern"
JAVA_OPTS="$JAVA_OPTS -Dsakai.cookieName=SAKAI2SESSIONID"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=8089 -
Dcom.sun.management.jmxremote.local.only=false -Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false"
```

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

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 -Dhttp.
agent=Sakai -Djava.util.Arrays.useLegacyMergeSort=true -Dfile.encoding=UTF8
```

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

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

Improve startup speed

You can improve startup speed under Tomcat 8 significantly.

For Sakai 12 and earlier: Edit the file conf/context.xml and add this JarScanFilter block to the <Context>

```
       <JarScanner>
        <!-- This is to speedup startup so that tomcat doesn't scan as much -->
        <JarScanFilter defaultPluggabilityScan="false"
defaultTldScan="false"
tldScan="jsf-impl-*.jar,jsf-widgets-*.jar,myfaces-impl-*.jar,pluto-taglib-*.jar,sakai-sections-app-util-*.jar,
spring-webmvc-*.jar,standard-*.jar,tomahawk*.jar,tomahawk-*.jar"/>
    </JarScanner>
```

For Sakai 19 and later: After the JSF 2.3 migration, users may update the tomcat configuration to this one:

```
       <JarScanner>
        <!-- This is to speedup startup so that tomcat doesn't scan as much -->
        <JarScanFilter defaultPluggabilityScan="false"
defaultTldScan="false"
tldScan="jsf2-widgets-*.jar,javax.faces-*.jar,jsf-impl-*.jar,jsf-widgets-*.jar,myfaces-impl-*.jar,pluto-taglib-
*.jar,sakai-sections-app-util-*.jar,spring-webmvc-*.jar,standard-*.jar,tomahawk*.jar,tomahawk-*.jar"/>
    </JarScanner>
```

2.0 Get the Sakai source code

There are two ways to acquire Sakai source code. You can choose to download a packaged *.zip or *tar.gz file from Sakai's release page or check out the code directly from our code repository using Git's source code control system and Github as the repository hosting service.

Sakai source code can be checked forked from our Github repository. The latest development work is located in `the master branch`; stable releases can be found in `tags` while maintenance and other work is performed in `a 19.x branch`.

19 release tag

To checkout a stable release tag, clone the git repo and then checkout the tag (in this case 19.0):

```
git clone https://github.com/sakaiproject/sakai.git

cd sakai && git checkout 19.0
```

Release Infomation

The latest stable release may be newer than the version listed above. Please see https://github.com/sakaiproject/sakai/releases.

19.x maintenance branch

The latest bug fixes for a particular release can be found in our maintenance branches. Please note that certain maintenance branch fixes require database schema changes. You can check out the maintenance branch by issuing the following command from the terminal:

```
git clone https://github.com/sakaiproject/sakai.git

cd sakai && git checkout 19.x
```

If you all ready have cloned the repository, skip step the clone step above.

3.0 Configure Sakai

1.0 Create a `sakai.properties` file

The `sakai.properties` file is a central configuration file that is typically stored in a `/sakai` subdirectory relative to the Tomcat home directory (`$CATALINA_HOME`). It is a non-XML text file containing a series of key/value pairs that is read using the load method of java.util.properties. Settings in `sakai.properties` govern everything from setting your institution's name to configuring your database. All settings in `sakai.properties` are read on startup; any changes you make subsequently will only take effect when you restart web application server. You may want to create a local.properties file in the same directory as sakai.properties. Properties listed in local.properties override sakai.properties.

For a source installation the default `default.sakai.properties` file is located in the `config` module:

```
sakai-src/config/configuration/bundles/src/bundle/org/sakaiproject/config/bundle/default.sakai.properties
```

⚠ The bin package does not include a `sakai.properties` file. This is a deliberate exclusion; it eliminates the possibility of overwriting a local `sakai.properties` file if a bin package is opened over an existing Sakai installation.

If you need to override the default settings you must create your own `sakai.properties` file either from scratch or from a known working copy adding new key/value settings in order to customize your installation. We recommend that you review the `default.sakai.properties` file included in the source installation or in the appropriate maintenance branch.

Version	default
11	default.sakai.properties
10	`default.sakai.properties`
2.9	`default.sakai.properties`

The default location for your local `sakai.properties` file is `$CATALINA_HOME/sakai`. This folder is not created by Maven during the build and deployment process, so you will have to create it manually or via a script. You can also store Sakai's configuration files outside of your web application server's file hierarchy. For example, in a development environment you may find yourself frequently reinstalling Tomcat and unless you create a build script to automate the Tomcat installation and configuration process avoiding having to recreate `$CATALINA_HOME/sakai` and `sakai.properties` each time has its advantages.

To locate your properties file outside of your web application server environment modify the Java startup command or the `JAVA_OPTS` environment variable and set a system property named `sakai.home`. Make sure your external location is readable and writable by your web application server.

```
-Dsakai.home=/path/to/desired/sakai/home/
```

ⓘ For list of sakai.properties settings see the Sakai Properties Reference; for detailed documentation on the full variety of possible `sakai.properties` settings, see the `sakai_properties.doc` in `/reference/docs/architecture/sakai_properties.doc`.

2.0 Configure home page tool set per site

Tool collections for the home page can be configured for each site type. However, if the `wsetup.home.toolids.*` property is not set, the Worksite setup tool will default to the following set of tools for the home page: `sakai.iframe.site`, `sakai.summary.calendar`, `sakai.synoptic.announcement`, `sakai.synoptic.chat`, `sakai.synoptic.messagecenter`. Synoptic tools will be added or dropped from home page depending on whether their linked tool exists in the site or not. See SAK-15504 - Getting issue details... STATUS and SAK-16747 - Getting issue details... STATUS for more details.

```
wsetup.home.toolids.count=5
wsetup.home.toolids.1=sakai.privacy
wsetup.home.toolids.2=sakai.iframe
wsetup.home.toolids.3=sakai.synoptic.announcement
wsetup.home.toolids.4=sakai.synoptic.chat
wsetup.home.toolids.5=sakai.synoptic.messagecenter
```

3.0 Work site setup group helper

A new group helper is enabled by default (see SAK-13413 - Getting issue details... STATUS for more details). Site maintainers can now create groups based on sections and roles. To switch back to the old 2.5 style of group helper (ad-hoc only), one needs to add following setting in `sakai.properties`:

```
wsetup.group.helper.name = sakai-site-manage-group-helper
```

4.0 Session timeout warning

Sakai includes a property called inactiveInterval@org.sakaiproject.tool.api.SessionManager, which dictates the length of inactive time before a users session times out and allows for the enabling of a session timeout warning. Session status is now checked by the Sakai portal. If the remaining session time is less than the warning time (say 10 minutes). When a session expires, the any page requests are redirected to the URL indicated by the `loggedOutUrl` sakai property. See SAK-13987 - Getting issue details... STATUS SAK-8152 - Getting issue details... STATUS for more details.

To enable the session timeout warning, set the following properties in your local `sakai.properties` with a time interval of your choosing:

```
timeoutDialogEnabled=true
timeoutDialogWarningSeconds=600
```

5.0 Configure email

Enabling Sakai to both send and receive email requires setting a number of properties in `sakai.properties`. In order to send mail Sakai requires the address (name or IP) of an external SMTP server that will accept mail from Sakai:

```
smtp@org.sakaiproject.email.api.EmailService=some.smtp.org
```

Sakai's SMTP server is Apache James for Sakai 10 and earlier and SubEthaSMTP from 11. Most sys admins prefer running a standard mailer like Postfix on port 25 and configuring it to forward requests to Sakai. You may also currently have a mailer service running on port 25 (Linux usually has it running by default). So consider setting Sakai's SMTP service to run on a different port (e.g., 8025) in order to avoid conflicts.

To enable Sakai to receive mail you'll need to set the following properties:

```
# flag to enable or disable SMTP service for incoming email (true | false)
#Default=false.
smtp.enabled=true

# dns addresses used by SMTP service for incoming email. (only supported on versions <= Sakai 10)
smtp.dns.1=255.255.255.1
smtp.dns.2=255.255.255.2

# SMTP port on which incoming SMTP service listens.
# Recommend running on 8025, and using a standard mailer on 25 to forward mail to Sakai.
# Default=25.
smtp.port=8025
```

Additional settings can be enabled to add support emails for a variety of tasks.

```
# Email support address used in incoming email rejection messages.
mail.support=address@somedomain
# A variation on this that's used in some places instead of the one above! Best to set both of them
support.email

#To change the postmaster address in general
smtpFrom@org.sakaiproject.email.api.EmailService
# Email address to send errors caught by the portal, and user bug reports in response.
portal.error.email=address@somedomain

# Email address used as the "from" address for any email sent by Worksite Setup tool or Site Info tool.
setup.request=address@somedomain

# Send an email to the user when the user is added.
notifyNewUserEmail=true


#For msgcntr notifications
msgcntr.notification.from.address

#Whether or not to send the real address as msgcntr notifications
msgcntr.notification.user.real.from=true (false)
```

6.0 Configure logging

Once you have Sakai installed, configured and started, you can monitor Sakai by watching the logs. The log level for the standard Sakai source code and the demo is set to show info and warnings only. Watch for the `WARN:` messages. There are going to be some "normal" ones at startup, and some will likely slip by at runtime, but any warning is potentially something you might want to check out.

Logging levels can be specified in `sakai.properties`. This augments and overrides the levels set in the default config file. Example:

```
log.config.count=3
log.config.1 = ALL.org.sakaiproject.log.impl
log.config.2 = OFF.org.sakaiproject
log.config.3 = DEBUG.org.sakaiproject.db.impl
```

This uses the established (if awkward) method of having a `name.count` followed by `name.1`, `name.2` etc. to form an array of strings for the value "name". In this case, the name is "log.config". The values are of the form `LEVEL.logger`, and the possible levels are: `OFF TRACE DEBUG INFO WARN ERROR FATAL ALL`.

Sakai uses log4j for logging. See the official log4j documentation for more information about how to configure it if you have questions, but a few notes are collected here below.

To change the logging for Sakai in the source modify `kernel/kernel-common/src/main/config/log4j.properties` the following property:

```
log4j.logger.org.sakaiproject=INFO
```

To turn on debug logging for all of Sakai, change the value from INFO to DEBUG. In order to enable debug logging for a single Sakai components, add a line such as in the following example, which will leave most of Sakai at INFO, but generate DEBUG level messages for the SQL service:

```
log4j.logger.org.sakaiproject=INFO
log4j.logger.org.sakaiproject.component.framework.sql.BasicSqlService=DEBUG
```

The logging controls are part of the LogConfigurationManager, implemented as a component in the Kernel. It can be disabled, if that's desired, with an entry in `sakai.properties`:

```
enabled@org.sakaiproject.log.api.LogConfigurationManager = false
```

For Mac and *nix systems, the most important log is found in Tomcat's `logs/catalina.out`. It can be instructive to watch this log as Tomcat is starting up, by using a startup command like the following:

```
bin/startup.sh; tail -f logs/catalina.out
```

Tomcat on Windows tends to be a little more puzzling about its logs, and it includes more of them, but its default behavior is to open `catalina.out` in a new window as soon as you start Tomcat. If you need more information about the logs in Windows, we'll refer you to the official Tomcat documentation.

The SMTP server logs from Sakai will be written to the $CATALINA_HOME/sakai/logs directory.

7.0 Managing temporary files

Depending on usage, Sakai may create many large temporary files in the system temporary file storage location (e.g. /tmp). It is a good practice to routinely remove temporary files older than a day, especially if you have a lot of users or have heavy usage of mail sending with attachments. Note that you should not simply remove all temp files on a schedule as some of them may be in active use (also be careful about removing temp files which are used by other processes).

3.1 Configure database

Sakai 12 database support details

Make sure to include a MySql connector jar in your CATALINA_HOME/lib.

Create a Sakai database with a default character set of UTF-8. Create a user account to which you will assign all permissions.

Use any database name and username you wish and then update your sakai.properties file (or even better, create a local.properties file, which has precedence).

In the example below the database is named "sakaidatabase" and the user account is "sakaiuser".

Excerpt from sakai.properties

```
username@javax.sql.BaseDataSource=sakaiuser
password@javax.sql.BaseDataSource=sakaipassword

## 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/sakaidatabase?useUnicode=true&characterEncoding=UTF-8
validationQuery@javax.sql.BaseDataSource=
defaultTransactionIsolationString@javax.sql.BaseDataSource=TRANSACTION_READ_COMMITTED
```

```
mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 272

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> create database sakai  default character set utf8;
Query OK, 1 row affected (0.00 sec)

mysql> grant all on sakai.* to sakaiuser@'localhost' identified by 'sakaipassword';
Query OK, 0 rows affected (0.00 sec

mysql> grant all on sakai.* to sakaiuser@'127.0.0.1' identified by 'sakaipassword';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> quit
```

4.0 Build, deploy and start up Sakai in Tomcat

Install the Sakai master project

Sakai makes use of Maven's Project Object Model to provide an XML representation of basic project information covering dependency management, build targets, external repositories, issue tracking, mailing lists, reporting plugins, developer bios, etc. A top-level master pom.xml file located in the `/master` project acts as a parent for other Sakai pom.xml files. If you are building Sakai for the first time you should install the master project first by issuing the `clean` and `install` goals from the `/master` project folder.

```
cd master
mvn clean install
cd ..
```

Install and deploy Sakai

Once you have built the master pom, issue the following Maven goals from the top-level Sakai source directory:

`-Dmaven.tomcat.home` specifies Tomcat's location. -Dsakai.home specifies the location of the sakai properties file(s) within Tomcat. -Djava.net.preferIPv4Stack can help resolve issues with Maven hanging while attempting to pull from certain repositories. These can be omitted if specified in Maven's `settings.xml` file (version 2.x) or in MAVEN_OPTS (version 3.x).

```
mvn clean install sakai:deploy -Dmaven.tomcat.home=/pathto/tomcathome -Dsakai.home=/pathto/tomcathome/sakai -
Djava.net.preferIPv4Stack=true
```

Your first Sakai build will take some time as Maven downloads and stocks your local `.m2/repository` with missing dependencies while performing the build and deploying **`.war` and** `.jar` files to Tomcat's `$CATALINA_HOME/webapps`, `$CATALINA_HOME/components` and `$CATALINA_HOME/shared/lib` folders. If during this process Maven reports that the build failed read the accompanying error message carefully to troubleshoot the issue (see the Troubleshooting section).

You can also issue `mvn clean install sakai:deploy` from any sakai project module top-level folder in order to build and deploy portions of Sakai such as individual tools.

Maven Goal Options

There are a number of other ways to build and deploy Sakai using Maven:

Print debug output

```
mvn -X clean install sakai:deploy
```

Build and Deploy in offline mode

If your local repository contains all Sakai project dependencies, you can run Maven "offline" by adding the `-o` option:

```
mvn -o clean install sakai:deploy
```

Skip unit tests

There may be occasions when you want to build and deploy Sakai without executing the set of unit tests that accompany many of the Sakai modules. If so add `-Dmaven.test.skip=true` to the goals you issue:

```
mvn clean install -Dmaven.test.skip=true sakai:deploy
```

Perform a "framework" build

The framework build profile triggers a Sakai build minus it's tool set.

The -Pframework build profile is no longer supported in Sakai 10.x versions.

```
mvn -Pframework clean install sakai:deploy
```

Perform a "cafe" build

The programmers' cafe build profile is favored by many developers new to Sakai as well as those developing or testing new capabilities.

⚠ The -Pcafe build profile is no longer supported in Sakai 10.x versions.

```
mvn -Pcafe clean install sakai:deploy
```

4.1 Start/Stop Tomcat

Start/stop Tomcat from the terminal by running the appropriate startup/shutdown script located in `$CATALINA_HOME/bin`:

Unix/Mac

```
sh startup.sh
sh shutdown.sh
```

Windows

```
startup.bat
shutdown.bat
```