

# Sakai 19 Configuration

## 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	<a href="#">default.sakai.properties</a>
10	<a href="#">default.sakai.properties</a>
2.9	<a href="#">default.sakai.properties</a>

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/
```

ℹ 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](#) - Upgraded site: editing My workspace leads to duplicate Home CLOSED and

🔴 [SAK-16747](#) - tools are not auto-added into Home tool page CLOSED 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 - Group creation enhancement](#) CLOSED 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 - Session Timeout Popup Utilizing Polling data from entity broker](#) CLOSED

[✓ SAK-8152 - Sakai should warn a user before the user's session times out and data is inadvertently lost](#) CLOSED 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).