

## ***Creating the Sakai 2.1.2 EAR File for WebSphere Community Edition***

### Overview

The WebSphere Community Edition (WASCE) is a FREE J2EE container and web application server based on Apache Geronimo and Tomcat. There is also a free Eclipse plug-in for WASCE. The plug-in allows someone to run and code using WASCE without leaving the Eclipse IDE.

The good news is that anyone can download, install, and use WASCE and the Eclipse plug-in for free. Even better is that WASCE is a supported IBM product. This means that everyone has the option to purchase support for WASCE from IBM.

This document describes how you can use Eclipse to deploy Sakai 2.1.2 on a WASCE server. It does not describe how to create a full Sakai development environment; hopefully we will address this in the near future.

### In General

WASCE is built on top of Apache Geronimo and Tomcat. This means that only a little work needs to be done in order to get up and running. To net it out, you will need to:

- Setup the Java and Eclipse environments
- Create an enterprise application (ear file) using Eclipse
- Import all Sakai war files into Eclipse
- Configure WASCE to load the common jar files
- Fix xml validation errors in web.xml files
- Edit the Geronimo deployment descriptor file for the war files to load local libraries first
- Identify and properly name all of Sakai's common jar files for WASCE recognition
- Create a Geronimo deployment descriptor xml file for the EAR file
- Configure WASCE to recognize Sakai startup parameters (sakai.home, sakai.components)
- Import and startup the EAR file

## ***Setting up the Environment (this is not necessary if you are using the Java Kickstart CD)***

### Optional Step 1: Install the SUN JDK 1.4.2\_11

I put mine in c:\j2sdk1.4.2\_11

### Optional Step 2: Download and Install Eclipse

I used version 1.0.1 of the 'All in one' WebTools Platform. Newer versions should work just fine:

<http://www.eclipse.org/webtools>,

### Optional Step 3: Download and Install WAS CE

A) Download WASCE 1.0.1.2:

[http://www.ibm.com/developerworks/downloads/ws/wasce/?S\\_TACT=105AGX10&S\\_CMP=WESDL](http://www.ibm.com/developerworks/downloads/ws/wasce/?S_TACT=105AGX10&S_CMP=WESDL)

B) Install WASCE 1.0.1.2, using the command line and -is parameter:

wasce\_setup-1.0.1.2-win.exe -is:javahome *javaDir*

Note: Substitute *javaDir* with your java home directory (e.g. mine is C:\j2sdk1.4.2\_11)

### Download and Install the WASCE 1.0.1.2 Plugin for Eclipse

Use the Eclipse Update Manager to install the WTP server adapter for WebSphere Application Server Community Edition

## ***Obtain Sakai and MySQL***

### Obtain the Sakai runtime code

There are two ways you can acquire the Sakai War files and common libraries used for running Sakai :

- From the Sakai demo. Simply download and extract the demo :  
<http://cvs.sakaiproject.org/release/2.1.2/>
- From source code. You will need to follow the instructions to download and build Sakai code : <http://cvs.sakaiproject.org/release/2.1.2/>

### Download, Install, and configure MySQL

Follow the Sakai Installation notes to properly configure the Sakai database:

<http://cvs.sakaiproject.org/release/2.1.2/>

## ***Prepare Sakai with Eclipse***

### Create the Eclipse Enterprise Application

Create a new J2EE Enterprise Application project

The Target Runtime should be WebSphere Application Server Community Edition

### Import all Sakai war Files into EAR

Import all 41 of the Sakai war files. Using the Sakai demo, these war files can be found in the Tomcat/webapps folder:

- Use Eclipse to Import each war file
- Target Runtime should be WebSphere Application Server Community Edition
- Make sure your new EAR application is selected, and the “add module to an ear application” checkbox is selected
- Click next. Deselect all web libraries. This will make the libraries appear in the applications WEB-INF/lib directory

### Fix all web.xml errors

WASCE is very careful to validate xml files before running a web application.

Unfortunately, there are quite a few validation errors that you need to fix. I’ve generalized the types of errors you might find, and how you should fix them. For the most part, you can get through by commenting out the tags that are no longer a part of the specification.

- Errors due to using version 2\_3 of the web-app schema. Simply replace with a reference to the new version, 2\_4:

```
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee  
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
```

- Invalid tags in the xml, which you need to simply comment out.
  - *Invalid content was found starting with element 'description'*
  - *Invalid content was found starting with element 'display-name'*.

### Deployment descriptors for the web applications – Geronimo-application.xml

WASCE likes to have a special deployment descriptor for its web applications. The descriptor is called Geronimo-web.xml. By default, Eclipse assumes that these web applications should load classes from the Geronimo server first.

For the **sakai-portlets** and **su-tool** applications, make sure the ‘Search the Geronimo Server First’ is selected. In the xml file, this equates to:

```
<context-priority-classloader>>false</context-priority-classloader>
```

I am doing this because these two web applications have their own commons-logging and/or log4j jar files, and I would like the sakai application to load the same logging classes as server. This may not be necessary, further testing is required.

For the remaining 39 web applications, modify the Geronimo-web.xml file to ‘Search the web app first’. In the xml file, this equates to:

```
<context-priority-classloader>>true</context-priority-classloader>
```

## Getting Sakai 2.1.2 on WebSphere Application Server Community Edition (WASCE)

Save your changes to the Geronimo-web.xml files

### Modify the Geronimo-application.xml

The EAR file itself has its own deployment descriptor, called Geronimo-application.xml. This file is used to identify any dependent jar files, and the order in which they should be loaded.

The first thing we need to do is add the dependencies for the common jar files. Check out the table below to see the libraries that I believe are common. We need to add each of these files to the Geronimo-application.xml file, but WASCE has problems with the way that some of these jar files are named. To fix this, you will need to make sure that the jar files are named properly by making sure that there is a hyphen between the jar files name and version number.

**Here is a table of how I renamed the jar files. Not all of them required renaming:**

Original file	New name
activation-1.0.2.jar	activation-1.0.2.jar
aopalliance-1.0.jar	aopalliance-1.0.jar
cglib-full-2.0.2.jar	cglib-full-2.0.2.jar
commons-beanutils-1.7.0.jar	commons-beanutils-1.7.0.jar
commons-collections-3.1.jar	commons-collections-3.1.jar
commons-dbc-1.2.1.jar	commons-dbc-1.2.1.jar
commons-fileupload-1.0.jar	commons-fileupload-1.0.jar
commons-lang-2.0.jar	commons-lang-2.0.jar
commons-pool-1.2.jar	commons-pool-1.2.jar
dom4j-1.4.jar	dom4j-1.4.jar
ehcache-1.1.jar	ehcache-1.1.jar
hibernate-2.1.8.jar	hibernate-2.1.8.jar
hsqldb-1.7.3.3.jar	hsqldb-1.7.3.3.jar
jdbc2_0-stdext-h2.1.8.jar	jdbc2_0-stdext-h2.1.8.jar
jdom-1.0.jar	jdom-1.0.jar
jstl-1.1.2.jar	jstl-1.1.2.jar
jta-h2.1.8.jar	jta-h2.1.8.jar
log4j-1.2.8.jar	log4j-1.2.8.jar
mail-1.3.1.jar	mail-1.3.1.jar
mysql-connector-java-3.1.12-bin.jar	mysql-connector-java-3.1.12.jar
odmg-3.0.jar	odmg-3.0.jar
OkiOSID-2.0.jar	OkiOSID-2.0.jar
quartz-1.5.1.jar	quartz-1.5.1.jar
sakai-active-tool-sakai_2-1-2.jar	sakai-active-tool-sakai-2.1.2.jar
sakai-authentication-sakai_2-1-2.jar	sakai-authentication-sakai-2.1.2.jar
sakai-common-agent-api-sakai_2-1-2.jar	sakai-common-agent-api-sakai-2.1.2.jar
sakai-common-authorization-api-sakai_2-1-2.jar	sakai-common-authorization-api-sakai-2.1.2.jar
sakai-common-composite-component-data-sakai_2-1-2.jar	sakai-common-composite-component-data-sakai-2.1.2.jar
sakai-common-edu-person-api-sakai_2-1-2.jar	sakai-common-edu-person-api-sakai-2.1.2.jar
sakai-common-manager-api-sakai_2-1-2.jar	sakai-common-manager-api-sakai-2.1.2.jar
sakai-common-registry-api-sakai_2-1-2.jar	sakai-common-registry-api-sakai-2.1.2.jar
sakai-common-superstructure-api-sakai_2-1-2.jar	sakai-common-superstructure-api-sakai-2.1.2.jar
sakai-common-type-api-sakai_2-1-2.jar	sakai-common-type-api-sakai-2.1.2.jar
sakai-common-uuid-api-sakai_2-1-2.jar	sakai-common-uuid-api-sakai-2.1.2.jar
sakai-comp-shared-help-sakai_2-1-2.jar	sakai-comp-shared-help-sakai-2.1.2.jar
sakai-comp-shared-scheduler-sakai_2-1-2.jar	sakai-comp-shared-scheduler-sakai-2.1.2.jar
sakai-comp-test-api-sakai_2-1-2.jar	sakai-comp-test-api-sakai-2.1.2.jar
sakai-component-sakai_2-1-2.jar	sakai-component-sakai-2.1.2.jar
sakai-components-loader-sakai_2-1-2.jar	sakai-components-loader-sakai-2.1.2.jar

## Getting Sakai 2.1.2 on WebSphere Application Server Community Edition (WASCE)

sakai-db-sakai_2-1-2.jar	sakai-db-sakai-2.1.2.jar
sakai-edu-coursemanagement-api-sakai_2-1-2.jar	sakai-edu-coursemanagement-api-sakai-2.1.2.jar
sakai-edu-coursemanagement-component-sakai_2-1-2.jar	sakai-edu-coursemanagement-component-sakai-2.1.2.jar
sakai-function-sakai_2-1-2.jar	sakai-function-sakai-2.1.2.jar
sakai-gradebook-help-sakai_2-1-2.jar	sakai-gradebook-help-sakai-2.1.2.jar
sakai-gradebook-service-sakai_2-1-2.jar	sakai-gradebook-service-sakai-2.1.2.jar
sakai-help-api-sakai_2-1-2.jar	sakai-help-api-sakai-2.1.2.jar
sakai-help-content-sakai_2-1-2.jar	sakai-help-content-sakai-2.1.2.jar
sakai-id-sakai_2-1-2.jar	sakai-id-sakai-2.1.2.jar
sakai-jgroups-sakai_2-1-2.jar	sakai-jgroups-sakai-2.1.2.jar
sakai-legacy-bundle-sakai_2-1-2.jar	sakai-legacy-bundle-sakai-2.1.2.jar
sakai-legacy-service-sakai_2-1-2.jar	sakai-legacy-service-sakai-2.1.2.jar
sakai-messageforums-api-sakai_2-1-2.jar	sakai-messageforums-api-sakai-2.1.2.jar
sakai-messageforums-component-shared-sakai_2-1-2.jar	sakai-messageforums-component-shared-sakai-2.1.2.jar
sakai-metaobj-api-sakai_2-1-2.jar	sakai-metaobj-api-sakai-2.1.2.jar
sakai-present-api-sakai_2-1-2.jar	sakai-present-api-sakai-2.1.2.jar
sakai-presentation-help-sakai_2-1-2.jar	sakai-presentation-help-sakai-2.1.2.jar
sakai-profile-api-sakai_2-1-2.jar	sakai-profile-api-sakai-2.1.2.jar
sakai-profile-component-shared-sakai_2-1-2.jar	sakai-profile-component-shared-sakai-2.1.2.jar
sakai-request-sakai_2-1-2.jar	sakai-request-sakai-2.1.2.jar
sakai-roster-api-sakai_2-1-2.jar	sakai-roster-api-sakai-2.1.2.jar
sakai-roster-component-shared-sakai_2-1-2.jar	sakai-roster-component-shared-sakai-2.1.2.jar
sakai-rwiki-help-sakai_2-1-2.jar	sakai-rwiki-help-sakai-2.1.2.jar
sakai-rwiki-model-sakai_2-1-2.jar	sakai-rwiki-model-sakai-2.1.2.jar
sakai-rwiki-service-sakai_2-1-2.jar	sakai-rwiki-service-sakai-2.1.2.jar
sakai-samigo-api-sakai_2-1-2.jar	sakai-samigo-api-sakai-2.1.2.jar
sakai-samigo-audio-sakai_2-1-2.jar	sakai-samigo-audio-sakai-2.1.2.jar
sakai-samigo-component-sakai_2-1-2.jar	sakai-samigo-component-sakai-2.1.2.jar
sakai-scheduler-api-sakai_2-1-2.jar	sakai-scheduler-api-sakai-2.1.2.jar
sakai-search-model-sakai_2-1-2.jar	sakai-search-model-sakai-2.1.2.jar
sakai-search-service-sakai_2-1-2.jar	sakai-search-service-sakai-2.1.2.jar
sakai-sections-api-sakai_2-1-2.jar	sakai-sections-api-sakai-2.1.2.jar
sakai-sections-comp-shared-data-sakai21-sakai_2-1-2.jar	sakai-sections-comp-shared-data-sakai21-sakai-2.1.2.jar
sakai-sections-help-sakai_2-1-2.jar	sakai-sections-help-sakai-2.1.2.jar
sakai-session-sakai_2-1-2.jar	sakai-session-sakai-2.1.2.jar
sakai-syllabus-api-sakai_2-1-2.jar	sakai-syllabus-api-sakai-2.1.2.jar
sakai-syllabus-component-shared-sakai_2-1-2.jar	sakai-syllabus-component-shared-sakai-2.1.2.jar
sakai-syllabus-help-sakai_2-1-2.jar	sakai-syllabus-help-sakai-2.1.2.jar
sakai-thread-local-sakai_2-1-2.jar	sakai-thread-local-sakai-2.1.2.jar
sakai-tool-registration-sakai_2-1-2.jar	sakai-tool-registration-sakai-2.1.2.jar
sakai-tool-sakai_2-1-2.jar	sakai-tool-sakai-2.1.2.jar
spring-1.2.6.jar	spring-1.2.6.jar
xalan-2.6.0.jar	xalan-2.6.0.jar

WASCE loads its own spring classes. We also should ‘hide’ these spring classes, and force our enterprise application to use the ones that work with Sakai.

**Here is a sample of my working Geronimo-application.xml file.** Note how we are using the newly-named jar files and hiding those dastardly spring classes.

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://geronimo.apache.org/xml/ns/j2ee/application-1.0"
xmlns:sec="http://geronimo.apache.org/xml/ns/security-1.1"
xmlns:sys="http://geronimo.apache.org/xml/ns/deployment-1.0" application-
name="Sakai212EAR" configId="Sakai212EAR/Sakai212EAR">

<dependency>
<uri>sakai/activation/1.0.2/jar</uri>
```

## Getting Sakai 2.1.2 on WebSphere Application Server Community Edition (WASCE)

```
</dependency>
<dependency>
<uri>sakai/aopalliance/1.0/jar</uri>
</dependency>
<dependency>
<uri>sakai/cglib-full/2.0.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/commons-beanutils/1.7.0/jar</uri>
</dependency>
<dependency>
<uri>sakai/commons-collections/3.1/jar</uri>
</dependency>
<dependency>
<uri>sakai/commons-dbc/1.2.1/jar</uri>
</dependency>
<dependency>
<uri>sakai/commons-lang/2.0/jar</uri>
</dependency>
<dependency>
<uri>sakai/commons-pool/1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/dom4j/1.4/jar</uri>
</dependency>
<dependency>
<uri>sakai/ehcache/1.1/jar</uri>
</dependency>
<dependency>
<uri>sakai/hsqldb/1.7.3.3/jar</uri>
</dependency>
<dependency>
<uri>sakai/jdbc2_0-stdext-h/2.1.8/jar</uri>
</dependency>
<dependency>
<uri>sakai/jdom/1.0/jar</uri>
</dependency>
<dependency>
<uri>sakai/jstl/1.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/jta-h/2.1.8/jar</uri>
</dependency>
<dependency>
<uri>sakai/mail/1.3.1/jar</uri>
</dependency>
<dependency>
<uri>sakai/mysql-connector-java/3.1.12/jar</uri>
</dependency>
<dependency>
<uri>sakai/odmg/3.0/jar</uri>
</dependency>
<dependency>
<uri>sakai/quartz/1.5.1/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-request-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-tool-registration-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-common-agent-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-common-authorization-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-common-composite-component-data-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
```

## Getting Sakai 2.1.2 on WebSphere Application Server Community Edition (WASCE)

```
<uri>sakai/sakai-common-edu-person-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-common-manager-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-common-registry-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-common-superstructure-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-common-type-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-common-uuid-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-components-loader-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-comp-shared-help-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-comp-shared-scheduler-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-edu-coursemanagement-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-edu-coursemanagement-component-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-function-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-gradebook-help-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-help-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-present-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-presentation-help-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-profile-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-profile-component-shared-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-roster-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-roster-component-shared-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-rwiki-help-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-samigo-audio-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-samigo-component-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-scheduler-api-sakai/2.1.2/jar</uri>
</dependency>
```

## Getting Sakai 2.1.2 on WebSphere Application Server Community Edition (WASCE)

```
<dependency>
<uri>sakai/sakai-search-model-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-search-service-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-sections-comp-shared-data-sakai21-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-sections-help-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-syllabus-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-syllabus-component-shared-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-syllabus-help-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-thread_local-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/xalan/2.6.0/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-help-content-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-id-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-jgroups-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-legacy-bundle-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-messageforums-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-messageforums-component-shared-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-component-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-active-tool-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-rwiki-service-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/commons-fileupload/1.0/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-comp-test-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-legacy-service-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-tool-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-metaobj-api-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-sections-api-sakai/2.1.2/jar</uri>
```



## Getting Sakai 2.1.2 on WebSphere Application Server Community Edition (WASCE)

```
</dependency>
<dependency>
<uri>sakai/sakai-rwiki-model-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/hibernate/2.1.8/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-db-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-authentication-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-session-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-gradebook-service-sakai/2.1.2/jar</uri>
</dependency>
<dependency>
<uri>sakai/OkioSID/2.0/jar</uri>
</dependency>
<dependency>
<uri>sakai/sakai-samigo-api-sakai/2.1.2/jar</uri>
</dependency>

<dependency>
<uri>org.apache.pluto/pluto/1.0.1/jar</uri>
</dependency>

<dependency>
<uri>sakai/spring/1.2.6/jar</uri>
</dependency>

    <hidden-classes>
        <filter>org.springframework</filter>
    </hidden-classes>

</application>
```

### Export the Ear file

At this point you can export your EAR application to a file on your local disk

## **Configure WASCE**

### Load those common library files

In the WASCE\_HOME/repository, create a new directory called /sakai/jars. Copy all of the above mentioned common jar files into this directory

### Modify Geronimo Environment to pass JVM Arguments.

In the WASCE\_HOME/bin directory, modify the set\_env.bat (Windows) or set\_env.sh (Linux) file. Near the bottom of the file, add the required Sakai JVM parameters for sakai.components.root and sakai.home.

### **Here is a sample of my set\_env.bat file**

```
@REM -----
@REM WASCE customizations for geronimo.bat.
```

## Getting Sakai 2.1.2 on WebSphere Application Server Community Edition (WASCE)

```
@REM
@REM This file is invoked prior to setjavaenv.bat and therefore selectively
@REM overrides default behavior.
@REM -----

@REM First instance of our WASCE variable is initialized by the installer..
set WASCE_JAVA_HOME=c:\j2sdk1.4.2_11
@REM Only use our WASCE pointer to Java if it is actually set to something
if "%WASCE_JAVA_HOME%" == "" goto noinstaller
set JRE_HOME=%WASCE_JAVA_HOME%
set JAVA_HOME=%WASCE_JAVA_HOME%
:noinstaller
if "%JRE_HOME%" == "" set JRE_HOME=%JAVA_HOME%
if "%JAVA_HOME%" == "" set JAVA_HOME=%JRE_HOME%
if not "%JAVA_HOME%" == "" goto setenvok
@echo Error: Neither WASCE_JAVA_HOME, JAVA_HOME nor JRE_HOME is defined.
@echo At least one of these environment variables needs to be set.
set ERRORLEVEL=1
goto end
:setenvok

@REM Setup our WASCE_HOME for the samples to use
if not "%GERONIMO_HOME%" == "" goto checkBase
@REM %~dp0 is expanded pathname of the current script
set GERONIMO_HOME=%~dp0..
@REM resolve .. and remove any trailing slashes
set CURRENT_DIR=%cd%
cd %GERONIMO_HOME%
set GERONIMO_HOME=%cd%
cd %CURRENT_DIR%

:checkBase
if "%GERONIMO_BASE%" == "" set GERONIMO_BASE=%GERONIMO_HOME%
if "%WASCE_HOME%" == "" set WASCE_HOME=%GERONIMO_HOME%
:gotBase
set JAVA_OPTS=%JAVA_OPTS% "-Djava.endorsed.dirs=%GERONIMO_BASE%/lib/endorsed" "-
Dsakai.home=C:/sakai212/sakai/" "-Dsakai.components.root=C:/sakai212/components/"

:end
@rem Done.
```

### ***Finish Configuring Sakai***

#### Copy the Sakai properties file

Make sure you put the properties file in your sakai.home directory (above).

#### Modify the Sakai properties file

Following the Sakai installation instructions, modify the sakai.properties file to communicate with the proper database.

#### Copy Sakai components directory

Make sure you copy the components directory to the location specified in the sakai.components.root parameter (above).

## **Testing**

### Start the server

When you first start the server, look for errors loading those common jar files. If you encounter an error, chances are that the jar file is improperly named or you did not properly include it in the Geronimo-application.xml file.

### Install and start the EAR

We do get some notifications in the console.

### **Here is what I see upon a proper startup:**

If you see many spring errors, make sure you properly hid the spring files from your application

### Checking the log

Other useful information can be found in: WASCE\_HOME/var/log/server.log

## **Current Issues**

### Cannot Log on as Administrator

I have not had enough time to look into this one properly, but somehow the administrator's password gets mangled when Sakai first sets itself up. I've seen this problem both with hypersonic and with MySQL.

The workaround is to launch the Sakai portal page and create a new user. Give this new user a password of *admin*.

Then you need to modify the sakai\_user table, and manually replace the pw for admin with the PW for the new user you have just created. With MySQL you can use the Query Browser tool to easily accomplish this.

If you are using the hypersonic database, you will need to modify the hypersonic SAKAI\_HOME/db/sakai.db.log or sakai.db.script file to make the change.

Then you should be able to log on as admin/admin

### “Out of Memory” Errors

Read the Sakai Installation guide to tweak the JAVA\_OPTS environment variable. Then either set a system environment or add information to the set\_env.bat file.

## **Other Issues?**

You should check out IBM's education industry landing page at:

More information about porting applications to WASCE can be found at IBM DeveloperWorks.

Thanks!

Chris Davia

[davia@us.ibm.com](mailto:davia@us.ibm.com)