

JBOSS Tomcat Dependencies

While efforts have been made to write LAMS using standard J2EE features it is configured to suit the JBOSS and Tomcat classloader and security implementations. It was not feasible to avoid these dependencies.

As we know about dependencies, they will be added to the list. This will make it easier to port to another application server if required - we will at least have a list of things we know we need to check/modify.

In some cases they are Tomcat dependencies, so it may be possible to avoid some problems in porting if the new application server also uses Tomcat.

General Configuration

Database configuration is done using the JBOSS configuration file mysql-ds.xml. These settings will need to be adapted to however the new application server configures JNDI database sources.

Security

JBOSS Cache

We use the built in JBOSS Cache, configured as an MBean. Currently, we include jboss-cache.jar and jboss-jmx.jar in our shared libraries for development/building.

The JBOSS Cache should work in other appservers but will need to be configured differently - we may not be able to use the MBean interface.

If this is not possible, then Hibernate will need to be reconfigured in the application content files to use another cache (e.g. ehcache) and any programmatically controlled object caching will need to be replaced with another cache strategy.

The package that needs to be modified for a new cache strategy are ???.

(Note: this work is not completed so it may change.)

EAR File Class Path

The application.xml has been coded to list all the [common jars](#) (Hibernate, Spring, Struts) as modules. This puts the modules on the EAR class path. Any appserver to which LAMS is ported must have an equivalent way of defining jars to be shared across all the web-apps in an EAR.