

# JBOSS Class Loader Configuration

## Unified Class Loader

Due to the structure of our EAR, we need to use the unified class loader. So the setting "UseJBossWebLoader" has been set to true. Our structure is may not be fully J2EE 1.4 compliant so it may cause problems on another application server.

File affected:

\\jboss-4.0.2\server\default\deploy\jbossweb-tomcat55.sar\META-INF\jboss-service.xml

For more information on this option, see <http://jira.jboss.com/jira/browse/JBAS-1691>

## Why?

If we don't use the unified class loader, then our tool jar files cannot find the jar files in the tool's war.

For example, consider the IMSCP tool uses the Reload project's jars to handle the IMS Content Package files.

The tool's jar is lams-tool-laicp10.jar and it sits directly in lams.ear. This is needed so that the progress engine can call the tool to create a new tool session, and to support other core -> tool function calls.

The tool's war MANIFEST references lams-tool-laicp10.jar. The Reload jars are in WEB-INF/lib.

Without the unified class loader, when the IMSCP authoring screens call the service bean to unpack an IMS content package, the code in lams-tool-laicp10.jar throws an exception as it cannot access the Reload jars.

In most cases, the third party jar files are shared across web-apps and so we have placed them directly in lams.ear. This appears to be the easiest way of sharing jar files across an EAR. But there will be a few cases, like the IMSCP tool that does have its own jar files.

## Potential Issues

Using the unified class loader gives the potential for a new tool to stop some or all of LAMS working. If the tool includes a jar file that conflicts with an existing jar file then it may cause errors - either to the new tool or to existing parts of the LAMS installation. This will not be an issue for certified tools (as the certification will check for these issues) but a non-certified tool may cause problems.