

# Troubleshooting

## Troubleshooting

These pages are the "gotchas" of programming LAMS. Problems we encountered using Spring, Hibernate, JBOSS, etc and how we worked around the problems.

They aren't necessarily the best solutions, but they are solutions, or at least warnings of problems you may encounter.

### Table of Content

- [Firewalls](#)
- [Flash Player](#)
- [Windows Stuck At Back in Firefox](#)
- [LAMS won't start due to "No ClassLoaders found for: org.jboss.cache.TreeCache"](#)
- [Flash Clients Not Updating or Mysteriously Crashing When Loading: Flash Getting Stuck](#)
- [Flash Clients Mysteriously Crashing: WDDX Exceptions](#)
- [Wrong URL](#)
- [Working Offline](#)
  - [Tool Deploy Utility](#)
  - [Tiles](#)
  - [Hibernate 2.0 DTD](#)
- [Turn on Access Log in Tomcat](#)
- [Initialising Spring Beans Within Transactions](#)
- [Invalid Mapping on Deployment](#)
- [javax.servlet.ServletException: PermGen space](#)
- [java.lang.OutOfMemoryError: PermGen space](#)
- [Can't save to database: org.hibernate.exception.GenericJDBCException: could not execute query](#)
- [HTTP Status 408 in Internet Explorer when logging in](#)
- [FCKEditor Problems \(Rich Text/HTML Editor\)](#)
- [Wildfire Problems](#)
- [LMS Integration Problems](#)
- [Caching errors: write lock for ... could not be acquired after 15000 ms](#)
- [java.sql.SQLException: Binary logging not possible. Message: Transaction level 'READ-COMMITTED' in InnoDB is not safe for binlog mode 'STATEMENT'](#)

## Firewalls

If you are having problems getting the login page to show, or "can't connect" errors to the MySQL or Wildfire (the chat server) then double check the firewalls on the server. For example, JBoss normally runs on port 8080 and your firewall may be set up to only allow traffic on port 80. If the MySQL database or the Wildfire server is on another server to LAMS, then the LAMS server must be able to access the MySQL/Wildfire server (note: the client PCs do not have to be able to access MySQL/Wildfire directly unless you want to access them via their own administration screens).

## Flash Player

We have encountered a number of problems in LAMS using the Flash Player 9,0,16,0. If you have this version, please upgrade to a newer version of the Flash player.

In authoring, the preview window and the tool authoring windows won't open on a Macintosh, running Firefox and Flash Player 8,22 and 8,34. If you are running on a Mac using Firefox, it would be safest to upgrade to Flash 9 at this point, although we are going to try to get it working with Flash 8. It does not seem to be a problem with Flash 8 on Windows XP.

## Windows Stuck At Back in Firefox

In LAMS 2.1 onwards you can have multiple tool windows opening in authoring. For example, if I have a Noticeboard window open and I open a Forum window, the Noticeboard window stays open. (This was different to 2.0). If you click again on the icon for a window that is open (so you double click on the Noticeboard activity again) then that window should be brought to the front.

But in Firefox it may not come to the front, due to Firefox's default settings. Go to the Options dialogue in the Tools menu (Windows) or the Preferences dialog (Mac OSX) and select Content. Click the 'Advanced' button and turn on 'Raise or lower Windows'.

## LAMS won't start due to "No ClassLoaders found for: org.jboss.cache.TreeCache"

When JBoss starts, the following error appears in the log and LAMS won't run:

```
Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'
```

```
12:20:59,734 INFO \[lams:org.jboss.web.tomcat.tc5.TomcatDeployer\] deploy, ctxPath=/web-console, warUrl=file:/D:/jboss-4.0.2/server/default/deploy/management/console-mgr.sar/web-console.war/ 12:21:00,484 ERROR \[lams:org.jboss.deployment.MainDeployer\] could not create deployment: file:/D:/jboss-4.0.2/server/default/deploy/local-service.xml org.jboss.deployment.DeploymentException: No ClassLoaders found for: org.jboss.cache.TreeCache; - nested throwable: (java.lang.ClassNotFoundException: No ClassLoaders found for: org.jboss.cache.TreeCache) at org.jboss.system.ServiceConfigurator.install(ServiceConfigurator.java:143) at org.jboss.system.ServiceController.install(ServiceController.java:202) at sun.reflect.GeneratedMethodAccessor48.invoke(Unknown Source)
```

You have forgotten to copy jgroups.jar and jboss-cache.jar from server/all/lib to server/default/lib.

## Flash Clients Not Updating or Mysteriously Crashing When Loading: Flash Getting Stuck

The Flash clients occasionally get "stuck" somehow. Either:

- the client itself is updated on the server and it appears from the logs that the browser is requesting the new client but the user seems to be stuck with an old copy of the client, or
- one of the datafiles get corrupted and the Flash client crashes during loading.

We have only seen it in Windows. We normally fix it by clearing the Flash Shared Objects area:

- Go to Windows Explorer and find "Documents and Settings\<your windows username>\Application Data\Macromedia\Flash Player\SharedObjects\<alphanumeric string>\<LAMS servername>". For example, on my PC currently:
  - the folder matching LAMS running on my local PC is "C:\Documents and Settings\FionaM\Application Data\Macromedia\Flash Player\SharedObjects\Z7JE3FN2#localhost" and
  - the folder for LAMS running on Shaun (our build server) is "C:\Documents and Settings\FionaM\Application Data\Macromedia\Flash Player\SharedObjects\Z7JE3FN2\shaun.melcoe.mq.edu.au"
- Delete any files found in the folder.
- In your browser, clear the browser cache and close the browser.
- Restart your PC.

## Flash Clients Mysteriously Crashing: WDDX Exceptions

Normally when responding to a call from Flash, the server tries to catch any exceptions and return an error message. But if the exception is thrown while creating the actual Flash packet (ie in the WDDX serialiser) then the Flash client will appear to "hang" and do nothing.

If this happens, have a look for an entry like this in lams.log:

```
Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'
```

```
FATAL org.lamsfoundation.lams.web.util.CustomStrutsExceptionHandler - fatal System exception: \[lams:java.lang.NullPointerException\] :null java.lang.NullPointerException at com.allaire.wddx.BeanSerializer.writeObject(BeanSerializer.java:108) at com.allaire.wddx.WddxOutputStream.writeObject(WddxOutputStream.java:303)
```

This will normally be caused by WDDX being unable to serialise something in the message. If this occurs, check the following:

- Make sure your collections are Vectors and Hashtables. Anything else may cause a problem.
- Do the methods and the variable names in the DTO match? For example, if you are returning a BlahDTO, a variable named "blahblah" but a method call of "getBlahBlah()" could cause a problem - the variable should be named blahBlah.
- All the getter methods will be called by the serialiser. Sometimes we forget and we create a getter method that we don't want called by the serialiser. For example, assume you have a BlahDTO, which is the DTO version of the Blah object. Do not have a method getBlah(), which converts the BlahDTO to the Blah object, as the serialiser will call getBlah() and try to serialise the Blah object. If you want to have a method like this, call it something like createBlah().
- Does the DTO contain a complex object that WDDX may not be able to cope with? Look for objects that from the core Java classes or another class library, rather than classes written for LAMS. For example, we have a DTO that contains a TimeZone object and we tried to serialise the DTO. It worked for a server running in the Brisbane timezone but not in the Sydney timezone. Turned out the Sydney TimeZone object contains a SimpleTimeZone and WDDX doesn't seem to be able to serialise SimpleTimeZone. The Brisbane timezone object doesn't contain a SimpleTimeZone object. We had to change our code so that we weren't trying to serialise a TimeZone object.

## Wrong URL

Symptoms:

- You get the error "The browser you are using doesn't support Rich Text Editor, Please use a supported browser".
- Pages look like the stylesheets or common images are missing.

If you encounter any "general wierdness" such as the symptoms listed above then check the URL that you are using to access LAMS. In System Administration screen (logged in as sysadmin), go to the Edit Configuration page and look for the entry ServerURL. The URL you use to access LAMS must be this URL.

For example, the default Windows url is <http://localhost:8080/lams/>, as most developers run it on their local PC. If you then log into LAMS using [http://127.0.0.1:8080/lams](http://127.0.0.1:8080/lams/), then the HTML pages will end up with some of the links specifying <http://127.0.0.1:8080/lams> and others <http://localhost:8080/lams/>, which the browser will see as two different servers.

The problem is compounded by the Firefox setting "Load Images for the originating site only". If you have this turned on and the URLs are different, then Firefox will not load up the images from the common area, hence the tabs don't look right and the HTMLEditor buttons don't appear. If you turn this setting off, then the buttons will display okay but you may run into other issues.

## Working Offline

When your PC is not connected to the Internet, all sorts of problems can occur due to the lack of access to dtd files.

If you find any instances where LAMS doesn't run (as opposed to a development task not running), please let [lams:Fiona](#) know or raise a JIRA entry for it. This needs to be fixed as LAMS must be able to run on without an Internet connection.

## Tool Deploy Utility

Tool Deployment fails trying to update the application.xml file. This is due to the deployment tool not being able to access application\_1\_3.dtd.

- Do the normal assemble-ear and deploy.ear.
- Remove the DOCTYPE line from the application.xml file in server/default/deploy/lams.ear/META-INF (I suggest cutting it to the paste buffer as you will need it again soon).
- Run the tool deployment
- Put the DOCTYPE line back into the application.xml.
- Start JBOSS

## Tiles

This must be fixed as it stops LAMS running.

The tiles dtd, tiles-config\_1\_1.dtd, cannot be found locally. It isn't in the struts jar, and isn't supplied as part of JBOSS. The only solution we have found so far is to remove the dtd entry in the tiles-defs.xml file. They should have been removed from existing files - please draw our attention to any tiles-defs.xml files that you may find with the dtd entry.

## Hibernate 2.0 DTD

This must be fixed as it stops LAMS running.

If you use Middlegen to generate your initial hbm.xml files, then it creates the file with a 2.0 DTD reference. You must change this to the 3.0 reference as the 2.0 DTD is not available locally.

## Turn on Access Log in Tomcat

If LAMS is run with Apache HTTP, then Apache would be used for the access logging. But what if an issue appears to be "after" Apache or if Apache isn't being used? Then we have to turn on the logging in Tomcat.

In the directory <jboss-directory>\server\default\deploy\jbossweb-tomcat55.sar, edit the file server.xml. Uncomment out the valve entry:

```
<!--
<Valve className="org.apache.catalina.valves.FastCommonAccessLogValve"
  prefix="localhost_access_log." suffix=".log"
  pattern="common" directory="{jboss.server.home.dir}/log"
  resolveHosts="false" />
-->
```

This will enable the logging valve that is designed for a production server. You can change the class to org.apache.catalina.valves.AccessLogValve or org.apache.catalina.valves.ExtendedAccessLogValve (W3c Extended Log File Format).

## Initialising Spring Beans Within Transactions

Initialising the normal Spring bean (ie singleton style) within transactions can lead to problems later. This "gotcha" usually appears in unit testing.

Generally Spring based beans are created when the Spring application context file is read. The beans are then available whenever you use them (in or out of a transaction). So, the beans are often created when JBOSS is started or when the war file is deployed to JBOSS.

But when running in junit, the application contexts are loaded when your code requests the context. If you are calling the content repository, then the contexts may be set up at two different times. (See Unit Testing With Content Repository).

Assume you have derived your test class from AbstractLamsTestCase. Your test code's context is created during setup(). The context for the content repository is set up when your code calls RepositoryProxy.getLocalRepositoryService().

If you call `getLocalRepositoryService()` and cache the value outside of a transaction, then the problem does not seem to occur. For example, in the IMS Content Package tool (`lams_tool_ims_cp`) I call `getLocalRepositoryService()` as part of the `ImscpServicePOJO` constructor. So when the context is configured in `setup()`, the content repository's context is also set up.

If your code happens to call `getLocalRepositoryService()` inside one of your transactions, then you may get the following exception when your code calls a content repository method.

Note: the formatting of this text has been changed to include extra carriage returns - otherwise all the text on the screen becomes too wide.

```
[lams:junit] Could not create Hibernate transaction; nested exception is java.lang.IllegalStateException:
Already value
    [lams:org.springframework.jdbc.datasource.ConnectionHolder@1758cd1] for key
    [lams:org.springframework.jdbc.datasource.DriverManagerDataSource@1873eb2] bound to thread [lams:main]
[lams:junit] org.springframework.transaction.CannotCreateTransactionException: Could not create Hibernate
transaction;
    nested exception is java.lang.IllegalStateException: Already value
    [lams:org.springframework.jdbc.datasource.ConnectionHolder@1758cd1] for key
    [lams:org.springframework.jdbc.datasource.DriverManagerDataSource@1873eb2] bound to thread [lams:
main]
[lams:junit] java.lang.IllegalStateException: Already value [lams:org.springframework.jdbc.datasource.
ConnectionHolder@1758cd1] for
    key [lams:org.springframework.jdbc.datasource.DriverManagerDataSource@1873eb2] bound to thread
[lams:main]
    [lams:junit] at org.springframework.transaction.support.TransactionSynchronizationManager.bindResource
(TransactionSynchronizationManager.java:137)
    [lams:junit] at org.springframework.orm.hibernate.HibernateTransactionManager.doBegin
(HibernateTransactionManager.java:415)
    [lams:junit] at org.springframework.transaction.support.AbstractPlatformTransactionManager.getTransaction
(AbstractPlatformTransactionManager.java:269)
    [lams:junit] at org.springframework.transaction.interceptor.TransactionAspectSupport.
createTransactionIfNecessary
(TransactionAspectSupport.java:201)
    [lams:junit] at org.springframework.transaction.interceptor.TransactionInterceptor.invoke
(TransactionInterceptor.java:49)
    [lams:junit] at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed
(ReflectiveMethodInvocation.java:138)
    [lams:junit] at org.springframework.aop.framework.adapter.MethodBeforeAdviceInterceptor.invoke
(MethodBeforeAdviceInterceptor.java:53)
    [lams:junit] at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed
(ReflectiveMethodInvocation.java:138)
    [lams:junit] at org.springframework.aop.framework.JdkDynamicAopProxy.invoke(JdkDynamicAopProxy.java:152)
    [lams:junit] at $Proxy1.createCredentials(Unknown Source)
    [lams:junit] at org.lamsfoundation.lams.tool.ims_cp.service.ImscpServicePOJO.configureContentRepository
(ImscpServicePOJO.java:241)
    [lams:junit] at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    [lams:junit] at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
    [lams:junit] at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
    [lams:junit] at org.springframework.aop.framework.AopProxyUtils.invokeJoinpointUsingReflection
(AopProxyUtils.java:61)
    [lams:junit] at org.springframework.aop.framework.ReflectiveMethodInvocation.invokeJoinpoint
(ReflectiveMethodInvocation.java:149)
    [lams:junit] at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed
(ReflectiveMethodInvocation.java:116)
    [lams:junit] at org.springframework.transaction.interceptor.TransactionInterceptor.invoke
(TransactionInterceptor.java:56)
    [lams:junit] at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed
(ReflectiveMethodInvocation.java:138)
    [lams:junit] at org.springframework.aop.framework.JdkDynamicAopProxy.invoke(JdkDynamicAopProxy.java:152)
    [lams:junit] at $Proxy0.configureContentRepository(Unknown Source)
    [lams:junit] at org.lamsfoundation.lams.tool.ims_cp.service.TestImscpService.testConfigureContentRepository
(Unknown Source)
    [lams:junit] at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    [lams:junit] at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
    [lams:junit] at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
```

Now, I'm not sure if this "fix" is really a fix or just a hack to avoid an underlying problem. I suspect that it is the latter (a hack) but it works for the moment. I have seen <http://forum.springframework.org/viewtopic.php?t=2259&highlight=key+already+bound> other references to people getting this error when initialising objects during a transaction.

## Invalid Mapping on Deployment

If you get the error listed below when you try to start JBOSS, check your server/default/lib directory. Remove any cglib or hibernate2 jar files in the lib directory.

The "correct" version of cglib and hibernate (hibernate3) are in the lams.ear file, but there are conflicting jar files in the lib directory then these will override the ear file values.

```
2005-06-16 17:05:19,141 ERROR [lams:org.springframework.web.context.ContextLoader] Context initialization failed

org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'coreSessionFactory'
defined in class
path resource [lams:org/lamsfoundation/lams/applicationContext.xml]: Initialization of bean failed; nested
exception is
net.sf.hibernate.MappingException: invalid mapping
net.sf.hibernate.MappingException: invalid mapping
at net.sf.hibernate.cfg.Configuration.addInputStream(Configuration.java:284)
at org.springframework.orm.hibernate.LocalSessionFactoryBean.afterPropertiesSet(LocalSessionFactoryBean.java:
383)
```

## javax.servlet.ServletException: PermGen space

same as below

## java.lang.OutOfMemoryError: PermGen space

This error occurs when the JVM runs out of space in the permanent generation heap. Normally happens when you have more than a handful of users using a server. The solution in this case is to set the MaxPermSize when LAMS starts.

By default the MaxPermSize is set to 64m (need to verify this is true for 1.5), so increase this as necessary.

On Unix systems, edit \$JBOSS\_HOME/bin/run.conf and set -XX:MaxPermSize in the JAVA\_OPTS section, as:

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
\# # Specify options to pass to the Java VM. # if \[lams: "x$JAVA_OPTS" = "x" \]; then    JAVA_OPTS="-server \-
Xms256m \-Xmx512m \-XX:MaxPermSize=128m" fi
```

On Windows systems, edit \$JBOSS\_HOME/bin/run.bat and set as:

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
rem Sun JVM memory allocation pool parameters. Modify as appropriate. set JAVA_OPTS=%JAVA_OPTS% \-Xms256m \-
Xmx512m \-XX:MaxPermSize=256m
```

The memory numbers above are examples only.

## Can't save to database: org.hibernate.exception.GenericJDBCException: could not execute query

For example, when clicking 'save' to save a tool's content, you get the above exception, and the following in the logs:

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
java.sql.SQLException: Illegal mix of collations (latin1_swedish_ci,IMPLICIT) and (utf8_general_ci,COERCIBLE) for
operation '='
at com.mysql.jdbc.MySQLIO.checkErrorPacket(MySQLIO.java:2928)
```

Means one or more of your tables aren't using default charset utf8 - at a mysql prompt, you can do a

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
select table_schema, table_name, table_collation where table_schema='lams';
```

to check, where table\_collation should all be utf8\_general\_ci.

## HTTP Status 408 in Internet Explorer when logging in

When trying to login to LAMS using Internet Explorer, you get this message text:

```
HTTP Status 408 - The time allowed for the login process has been exceeded.
If you wish to continue you must either click back twice and re-click the link you requested or close and re-
open your browser
```

The solution is to adjust your privacy settings to allow the LAMS server's cookie through. To do this, either:

- Go to Tools->Options, and select the 'privacy' tab. Lower the privacy slider to a less restrictive setting.
- Or go to Tools->Options, select the 'privacy' tab, and then click the 'sites' button. Add your LAMS server domain (e.g. <http://example.com>) to the list of allowed sites.

## FCKEditor Problems (Rich Text/HTML Editor)

See the [FCKeditor](#) page.

## Wildfire Problems

See the [Wildfire](#) page.

## LMS Integration Problems

See the [Integration](#) page.

## Caching errors: write lock for ... could not be acquired after 15000 ms

JBoss 4.0.2 is using JBossCache libraries that have some Hibernate - Spring integration bugs. Those bugs were fixed in later releases. If you repeatedly encounter locking errors:

- download [JBossCache](#) version 1.4.x (but not version 2 or higher, since it will not work with JBoss 4)
- from this distribution copy jgroups.jar, jboss-cache.jar and jboss-serialization.jar to \$JBoss\_HOME/server/default/lib/ on your JBoss server
- restart the server

Starting with LAMS 2.2 these libraries will be a part of LAMS distribution and will be deployed along with the project.

## java.sql.SQLException: Binary logging not possible. Message: Transaction level 'READ-COMMITTED' in InnoDB is not safe for binlog mode 'STATEMENT'

This is related to a MySQL [bug](#). Workaround is to set binlog\_format=MIXED in your my.cnf file.