EL4J

TroubleshootingGuideEl4j

Generated from Wiki page:
http://wiki.elca.ch/twiki/el4j/bin/view/EL4J/TroubleshootingGuideEl4j

Version	Date	Author(s)	Visa
1.14	15 Aug 2008 - 08:17	DavidBernhard, Main.DavidBernhard, Main.FlorianKeusch, Main.PhilippHOser, Main.StefanWismer	

©ELCA Informatique SA, Switzerland 2009

Table of Contents

1 TroubleshootingGuideEl4j....1
1.1 Maven troubleshooting....1
1.1.1 Maven does not build....1
1.1.1.1 Tomcat issues....3
1.1.1.2 Jar hell detection....3
1.1.1.3 Smaller issues with Maven....4
1.1.2 JSF troubleshooting....5
1.1.3 Troubleshooting other issues with EL4J....5
1.1.3.1 A Spring Bean is not found or is duplicated....5
1.1.3.2 Class not found exception during application runtime....6
1.1.3.3 What interceptors are active on a Spring bean?....6
1.1.3.4 My interceptor is not added to the interceptor chain as I expect it. What shall I do?....6
1.1.3.5 Data that you send to the DAO is not written to the database....6
1.1.3.6 A service call takes too long....7
1.1.3.7 The application is blocked....7
1.1.3.8 Hessian / Burlap and overloading....7
1.1.4 Resources with further help....7

1 TroubleshootingGuideEl4j

- Maven troubleshooting
  - Maven does not build
  - Tomcat issues
  - Jar hell detection
  - Smaller issues with Maven
- JSF troubleshooting
- Troubleshooting other issues with EL4J
  - A Spring Bean is not found or is duplicated
  - Class not found exception during application runtime
  - What interceptors are active on a Spring bean?
  - My interceptor is not added to the interceptor chain as I expect it. What shall I do?
  - Data that you send to the DAO is not written to the database
  - A service call takes too long
  - The application is blocked
  - Hessian / Burlap and overloading
- Resources with further help

1.1 Maven troubleshooting

1.1.1 Maven does not build

- Often maven tells you what the issue is. Maybe calling maven with `-e` or `-X` helps to figure out more.
- If an artifact is missing:
  - Is the corresponding artifact correctly named (group id, artefact id, version)?
  - Try putting the artefact in your local repository
  - Verify that maven looks in the correct repositories (maven shows you what repositories it knows). You can browse the repositories with a browser.
  - You may need to build and install `mvn install` some other maven projects in your local repository first (so that they can be found)/ Try rebuilding with `mvn clean install` in the global repository
  - Try using the option "-o" to have mvn work offline.
  - Try using the option "-U" to look for new available snapshots
  - Rename your local repository (to hide it temporarily from maven)
  - Detect what maven does with Wireshark. Sometimes the network is just too slow for maven.
  - Sometimes an artifact is not found in proximity. Try relaunching proximity.
- If there is a mistake in your pom.xml
  - Ensure the XML is syntactically correct
  - Try `mvn -N help:effective-pom` and `mvn -N help:effective-settings`
  - Compare your pom.xml to a working sample pom.xml
  - Does it help if you remove certain parts of your pom (e.g. definition of some plugins, some dependencies, ...)?
- There is a duplicated jar-file in the generated war or ear files or in the class path. A related question is : how do I check the sanity of all my dependencies (e.g. before a release)?
  - A useful tool to check the sanity of your dependencies is to run `mvn site` on the global level of your project. (we do this e.g., to check the sanity of EL4J). You can then go through all dependencies in the html file under `target/site/dependency-convergence.html` (under `Project Information -> Dependency Convergence`). If you need more info on the dependency, you can find it under the section `Project Information -> Dependencies` of each of your modules.
  - In the directory of the `pom.xml` that generates the file with the duplicated jar files, execute `mvn project-info-reports:dependencies` to get a report on the dependencies (the report is then under `target/site/dependencies.html`). In the dependency tree of this report, you will find the conflicting definition that lead to the duplicated jar files. Usually it occurs because a dependency to a same jar file is named differently. Ideas to resolve it: unify

the naming or make an exclusion statement when adding the the dependency. Here is a sample exclusion statement:

```
<dependency>
   <groupId>org.codehaus.xfire</groupId>
   <artifactId>xfire-core</artifactId>
   <version>1.2.4</version>
   <exclusions>
       <exclusion>
           <groupId>commons-logging</groupId>
           <artifactId>commons-logging</artifactId>
       </exclusion>
       <exclusion>
           <groupId>jdom</groupId>
           <artifactId>jdom</artifactId>
       </exclusion>
   </exclusions>
</dependency>
```

- ♦ To do this on a whole project, try the following:

```
cd $EL4J_ROOT/external
mvn depgraph:fullgraph -Ddepgraph.filterEmptyArtifacts=true -Ddepgraph.dotFile=el4j.dot
grep label el4j.dot | sort > DuplicateCandidates.txt
# resolve candidates manually (find in DuplicateCandidates.txt)
# determine candidates graphically (in el4j.png) or in text (in el4j.dot)
```

- ♦ If in a generated war file, you have 2 times the same file under different names, e.g. `WEB-INF\lib\module-core-1.9-20080123.163316-1.jar` and `WEB-INF\lib\module-core-1.9-SNAPSHOT.jar`. This can be a bug of the war plugin that only occurs with snapshots. The file with ending "-SNAPSHOT.jar" comes from "module-web-war" but in "module-web-war" there should be "module-core-1.9-20080123.163316-1.jar" so merging of wars would work properly. To solve this issue add the following pom-config in the pom of your war module:

```
<pluginManagement>
   <plugins>
      <plugin>
         <!--
            Ignore the SNAPSHOT libs delivered by war
            and take the "fresh" ones.
          -->
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-war-plugin</artifactId>
            <configuration>
               <dependentWarExcludes>WEB-INF/lib/*-SNAPSHOT.jar</dependentWarExcludes>
            </configuration>
         </plugin>
   </plugins>
</pluginManagement>
```

- There is another strange maven error
  - ♦ Take the latest mvn version from the EL4J page (from the convenience zip).
  - ♦ Have you followed all steps of the GettingStarted guide?
  - ♦ When you interrupt the downloading of the tomcat zip file, the unzip process does not work. Delete the incomplete tomcat*.zip file and try again.
  - ♦ Try uncommenting the mirror setting in your `settings.xml` file => does the problem go away?
  - ♦ Under cygwin, the ~ is sometimes set to something like `c:/cygwin/home/` (via the environment variable HOME). But in general under Windows, ~ is located under `C:/Documents and Settings/`. Mvn and svn take the default ~ setting of Windows.
  - ♦ Do you use backslashes '\' instead of slashes '/' in your settings.xml file.

- Under Windows only: Sometimes there are issues with paths that are interpreted by Windows or Unix (Cygwin). When a script is launched in cygwin, it uses the Unix conventions. But as soon as a call is made to Java, the (Windows) executable uses the Windows conventions. Unix conventions contain e.g. '/' as directory separator and ':' as a path separator. Windows conventions contain e.g. '\' as directory separator and ';' as a path separator.
- It's possible that you have 2 different JVM versions in eclipse and in mvn. Use the same version (both for execution and build time).

1.1.1.1 Tomcat issues

- Try rebuilding with `mvn clean install`
- Look into the log files of tomcat (under `logs/`).
- Try copying the file `etc/tomcat/context.xml` to the tomcat `conf` directory.
- Remove the deployed directories and the deployed war files in the `webapps` directory of tomcat and try again
- Sometimes if only a `xml` file is changed, the application is not reloaded. Try making a `touch` (command in cygwin) on another file to force a reload.
- Add a user with the role "admin" in the file "conf/tomcat-users.xml". This allows that you can manage the application via the tomcat manager (web user interface).
- Try deleting the zip that is used to download tomcat and try again
- You may want to make a try with Jetty

1.1.1.2 Jar hell detection

Jar hell is when the same jar or class is included more than once on the classpath. One possible symptom is that assembly/singlejar builds do not run whereas executing with maven (exec:java) does; this is due to different classpath search orders.

If maven detects two different versions of the same dependency like cglib:cglib:1.0 and cglib:cglib:1.1 on the classpath, it will use its own strategies to pick which one to use. This is not jar hell. However, if they are not identically named like cglib:**cglib**:1.0 and cglib:**cglib-full**:1.0 maven considers them different artifacts and adds them all.

Add the maven duplicate finder plugin to your pom with

```
<plugin>
   <groupId>ch.elca.el4j.maven.plugins</groupId>
   <artifactId>maven-duplicatefinder-plugin</artifactId>
   <executions>
      <execution>
         <id>dup</id>
         <goals>
            <goal>inspect</goal>
         </goals>
         <phase>verify</phase>
      </execution>
   </executions>
</plugin>
```

Unfortunately you cannot just do `mvn duplicatefinder:inspect` yet as that would not pick up dependencies.

Your build will pause and a window appear showing all classes on the classpath and their location (click on them for more information). If the root entry `(root)` is black, all is well. If not, follow the tree down to a red .class file and click it to see its locations in the bottom of the window. As the full path is displayed, you can find the group/artifact/version information from the path.

Run `mvn dependency:tree` to get the full dependency tree of your build. Find the two or more conflicting artifacts from the last step and see which of your direct dependencies included them.

In your dependencies, add a manual exclusion to prevent an artifact included somewhere else from being wrongly excluded. For example,

```
<dependency>
    <groupId>joda-time</groupId>
    <artifactId>joda-time-hibernate</artifactId>
    <version>1.0</version>
    <exclusions>
   <exclusion>
       <groupId>cglib</groupId>
       <artifactId>cglib-full</artifactId>
   </exclusion>
   <exclusion>
       <groupId>commons-logging</groupId>
       <artifactId>commons-logging-api</artifactId>
   </exclusion>
    </exclusions>
</dependency>
```

This excludes any version of cglib-full or commons-logging that jodatime has declared as a dependency. The proper cglib will then be included from somewhere else.

1.1.1.3 Smaller issues with Maven

- **Q** Since we have moved to Tomcat 6, it seems that Tomcat logs information intended to the standard output only in its file localhost..log (typically what your application writes to System.out or exceptions not handled by the application and caught by Tomcat). How can I configure again the old behavior?

- **A** If you want to see it on the console as well, which makes it easier to quickly identify a problem, add the following red part in your `$CATALINA_HOME/conf/logging.properties` file:

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].handlers =
2localhost.org.apache.juli.FileHandler**, java.util.logging.ConsoleHandler**

- **Q** I installed EL4J as described. When using eclipse, if I put my cursor on a class name, I often get the error "A Runtime Error has occurred. Do you wish to debug?". When I debug it, it points me to a line of javascript (method printFileStatus) inside the javadoc. What can I do?

- **A** This happens because the javascript function inside the javadoc can not be found. Normally this will be ignored but the Office-Suite installs a debugger which will be launched. To turn this "feature" off just disable the JIT debugging of this debugger.

- **Q** Why does JBoss AS always stop working after a few (re)deployment cycles?

- **A** JBoss supports hot re-deployment of WARs and EARs. Unfortunately, due to bugs in the JVM, repeated redeployment of an EAR eventually causes the JVM to run out of perm gen space. For this reason, we recommend running JBoss in a JVM with a large perm gen space at development time (configure this in bin/run.conf): `-Xms512m -Xmx1024m -XX:PermSize=256m -XX:MaxPermSize=512`

- **Q** The build of EL4J in the external directory fails due to jax-ws (I am using JDK 6)?

- **A** This is a known issue of jax-ws. If you want the jaxws support working, set up jaxws 2.1 like described here: http://weblogs.java.net/blog/ramapulavarthi/archive/2007/01/problems_using.html

- **Q** `mvn site` does not work in the external directory. What to do?

- **A** `mvn site` does only work in the external/site directory

- **Q** During a build, I receive the following error: "java.io.EOFException: Unexpected end of ZLIB input stream" during the extraction of the tomcat zip file.

- **A** Try deleting the tomcat `.zip` file that was incompletely downloaded and start again. (It is in the `tomcat5x.basedir` as defined in the m2 `settings.xml`).

- **Q** I get a NullPointerException in FtpWagon.openConnection when executing mvn deploy?

- **A** Try setting the correct user name and password in your settings.xml file (under `servers/server/username` and `servers/server/password`).

- **Q** A DLL file is missing. What shall I do?

- **A** There is a DLL file (only for Windows) in the internal file tool module. Put it e.g. into your jre/bin directory.

- **Q** During db:create or another database operation, the database cannot be cleared and/or created.

- **A** Remove the disturbing parts of the database tables and constraints. If you don't need to keep the database (e.g. during development) you can clear the database(s) by removing the derby database file (e.g. under `D:\Projects\EL4J\tools\derby\derby-databases`) or by calling `db:destroy` (e.g. for oracle). An oracle destroy script can be found under `external\framework\modules\database\src\main\resources\etc\sql\oracle`.

1.1.2 JSF troubleshooting

Sometimes it's not easy to see what's going on. Look at the exception stack trace in the output in your shell. During the development with JSF there are 3 main sources of errors:

- Mismatch between the **database** model and your **bean entities**.
  - Check the SQL scripts if the fields match the fields of the entities.
  - Examine your DAO or hibernate configuration.
  - Check the annotations in your Java beans (e.g. @NotNull, @Length(max = 32) matches your SQL definitions)
  - Don't use database keywords in your SQL scripts for names (e.g. don't create a table with the name 'user'). Some databases can't cope with that.

- Your **'.xhtml' faces configuration files** contain some errors.
  - Check for misspellings (this happens often).
  - A method (e.g. you call as an action when clicking on a button) does not exist or exists with other parameters.
  - Check the page navigation.

- Your **Java classes** contain bugs.
  - Check the work flow.
  - Check for null values.
  - Use the debugger to see where it fails. See DebuggingHowTo

1.1.3 Troubleshooting other issues with EL4J

1.1.3.1 A Spring Bean is not found or is duplicated

- What spring configuration files are loaded (look at the spring output (to enable more output: refer to the TracingStackEl4j))?
- Ensure that you use the ModuleApplicationContext or the ModuleWebApplicationContext
- What is the classpath? => please refer to the next point

- What is the list of inclusive/ exclusive locations where you launch the Module*ApplicationContext?
- Is it possible that a jar file is twice in the classpath, maybe of a different version? Could it be that a configuration file exists once from a jar and once via a classed directory?

1.1.3.2 Class not found exception during application runtime

- What is your classpath? In web applications: you can easily check the jars in the exploded-war directory. In general: when you run a "mvn install" of your project you see the full list of runtime dependencies (= all modules that are included in the class path). Another alternative is to look into the dependency graph (refer to dep graph maven plugin). When launching applications in Maven, the normal classpath (System.getProperty("java.class.path")) only contains classworlds. => maybe it's easier to figure out when launching the application via eclipse?
- Often it is useful to find out in what jar the class is located (via Google or via the prefix of your classes).
- Looking at what classes are loaded and from where they are loaded gives also some indication. This can be done by adding the switch `-verbose` when launching the JVM.

1.1.3.3 What interceptors are active on a Spring bean?

- Have a look at the stack trace when a method on the bean is executed. This can be done via the debugger, via an exception (e.g. via `System.out.println(new Exception().printStackTrace())` or a thread dump.
- The EL4J JMX support allows to see all the interceptors on a spring bean.
- The following blog entry describes simple code to get all interceptors on a spring bean: http://swik.net/Spring/Ben+Hale's+Blog/Another+Reason+to+Love+Spring+2.0:+Interceptor+Combining/die8

1.1.3.4 My interceptor is not added to the interceptor chain as I expect it. What shall I do?

- Check the above question to ensure that your interceptor is really not added.
- A known issue is that you may have multiple proxies around the spring bean that confuses the code that adds the interceptor. E.g. when configuring the trace-interceptor and the transaction-interceptor (e.g. directly or indirectly via annotations), the transaction-interceptor may not work!
  We plan to provide some improvement for this "feature" (in order not to speak of a bug) of spring.
- Another issue we have encountered is that due to a circular dependency, a spring bean was not proxied (and therefore also had no interceptor around it).
- It is also possible that the support to combine JDK 1.5 annotations or commons attributes with interceptors is not turned on. Please check in the documentation of your annotation/ attribute whether the corresponding config file is loaded (`framework\modules\core\src\main\resources\optional\interception\transactionJava5A`
- Sometimes the commons attribute files are not generated correctly (there is 1 class file per annotated java class). Try clean/ rebuild, and verify that commons attribute is activated in the build system.

1.1.3.5 Data that you send to the DAO is not written to the database

- Is a transaction active? Does it commit?
  - Is your transaction interceptor turned on as you expect it (check the earlier point about this topic)
  - Is the @Transactional annotation set (either on the DAO or on the service layer)?
  - Is a transaction manager bean defined?
  - Enable tracing in the Spring Transaction manager to see whether a start transaction/ commit is sent to the db (try to set the logging level of the classes in the package "org.springframework.jdbc." to "DEBUG". This shows you in the log what the spring transaction management does. BTW: this only works if you have the EL4J-default, non-JTA transaction manager. Otherwise you need to adapt the package above to include your particular transaction manager.)
- Is there an exception during your call? => do you loose some exception on the way?

- Does the db receive your SQL? => enable logging of db output (in hibernate, ibatis, in the db, or via P6Spy)
- Look into the database (e.g. via Squirrel) => does everything look ok?
- Compare your transaction code and configuration to those of the EL4J standard example (refdb and keyword demos). What is different in the configuration/ code/ annotations?

1.1.3.6 A service call takes too long

- What is the state of your system ressources (memory, cpu, file-system, network database, other servers, ...)? See TracingStackEl4j for tools to look at each of these resources.
- What happens globally in your call? Are there many remote calls?
- Use a profiler
- Use the simpleStatistics/ detailed statistics module to track down execution times.
- See also next point
- Are all your CPU cores used? => TracingStackEl4j

1.1.3.7 The application is blocked

- Try a thread dump (Crtl-Break, see TracingStackEl4j) => what threads are blocked? Does a `jstack -l` help?
- Try calling the deadlock detection routine of the JDK 6 (java.lang.management.ThreadMXBean)
- Swing applications: check in the EDT ideas of the TracingStackEl4j document.
- Can you localize what sections (threads/ processes) of the application are still running?

1.1.3.8 Hessian / Burlap and overloading

- Hessian / Burlap does not enable method overloading by default. If you have the methods m(A a) and m(A a, B b) then it knows only 1 method. To solve this problem you can choose different names or enable overloading. If overloading is enabled this might slow your application?? because now every method is mangled. To enable overloading you can use the property `overloadEnabled` in your protocol configuration file:

```
<property name="serviceProperties">
    <map>
        <entry key="overloadEnabled" value="true" />
    </map>
</property>
```

1.1.4 Resources with further help

- http://wiki.elca.ch/twiki/el4j/bin/view/EL4J/FrequentlyAskedQuestions
- http://wiki.elca.ch/twiki/el4j/bin/view/EL4J/TracingStackEl4j
- Issues that occurred in the short term: http://el4j.svn.sourceforge.net/viewvc/*checkout*/el4j/trunk/el4j/etc/KnownIssues.txt
- The documentation of the features you are using (mainly under ModuleDocumentations).