Problem: Conflicting libraries

Author:
ObjectPlanet, Inc.

Created On: 11 Sep 2008 09:50 AM

Opinio uses a number of third-party libraries. These are included in the [opinio-home]/WEB-INF/lib directory. Sometimes these libraries might conflict with libraries used by other applications or the application server itself. If this situation occurs, you need to tell the Java runtime which library to use. How to configure this correctly depends on what application server is used. It might also depend on the Java distribution/vendor used.

General work-around, if you know which .jar files that are conflicting, and Opinio is the only application on the application server:
Rename the xxx.jar file (the one that is NOT in the [opinio-home]/WEB-INF/lib directory) to "xxx.jar.unused", and restart the application server.

Another work-around is sometimes to tell the Java runtime in what folders (and in what order) to look for libraries. This is specified in the Java classpath. Refer to your documentation for the application server on how to set the classpath. For example, in Tomcat 5.5 for Windows this is configured in the file "setclasspath.bat". Look for the line starting with
"set CLASSPATH=%JAVA_HOME%libtools.jar". If you have a conflicting version of "mail.jar" for example, change this line to: "set CLASSPATH=[opinio-home]/WEB-INF/lib/mail.jar;%JAVA_HOME%libtools.jar"

See below for example of error messages, and how it might be solved:

EXAMPLE 1:

ERROR -- EmailManager.sendEmail(): Can not find provider. javax.mail.NoSuchProviderException: smtp
ERROR -- Unable to send email with exported invitees file to 'Invitees from 'Everyone else'': com.objectplanet.survey.util.S: Email Error

Possible solution: Rename or delete the conflicting library.

EXAMPLE 2:

-- ERROR -- User admin:
javax.xml.stream.FactoryConfigurationError: Provider null could not be instantiated: java.lang.NullPointerException
at javax.xml.stream.FactoryFinder.newInstance(FactoryFinder.java:75)
at javax.xml.stream.FactoryFinder.find(FactoryFinder.java:136)
at javax.xml.stream.FactoryFinder.find(FactoryFinder.java:92)
at javax.xml.stream.XMLInputFactory.newInstance(XMLInputFactory.java:136)

Possible solution:

In the tomcat startup.sh, we need to add the following java option
-Djavax.xml.stream.XMLInputFactory=com.bea.xml.stream.MXParserFactory
The XMLInputFactory is an abstract class that is part of stAX (Stream APIs for XML) APIs. It is used for XML reading process. To instantiate the class, the JVM needs to find an implementation class for it. There are implantations from Sun, BEA, Oracle, etc.
The JVM uses the system variable to identify an implementation class. For Java5, the stAX is part of core package so when this value is not specified, it would use the default value, com.bea.xml.stream.MXParserFactory. However, for Java 1.4, there is no default value.
For Opinio, it uses the implementation from BEA. It is resides in jsr173_1.0_ri.jar. Because of the missing system property, it cannot find the value for the implementation class. By adding the proper java option in the startup.sh should fix this problem.

Another possible solution:
If there is a file called "jaxp.properties" in your JDK/jre/lib folder, and you are running WebLogic, add this line to the file:
javax.xml.stream.XMLInputFactory=com.bea.xml.stream.MXParserFactory