

Configuring a Different Xalan.jar in ColdFusion MX 6.1

Posted At : October 14, 2003 2:06 PM | Posted By : Steven Erat

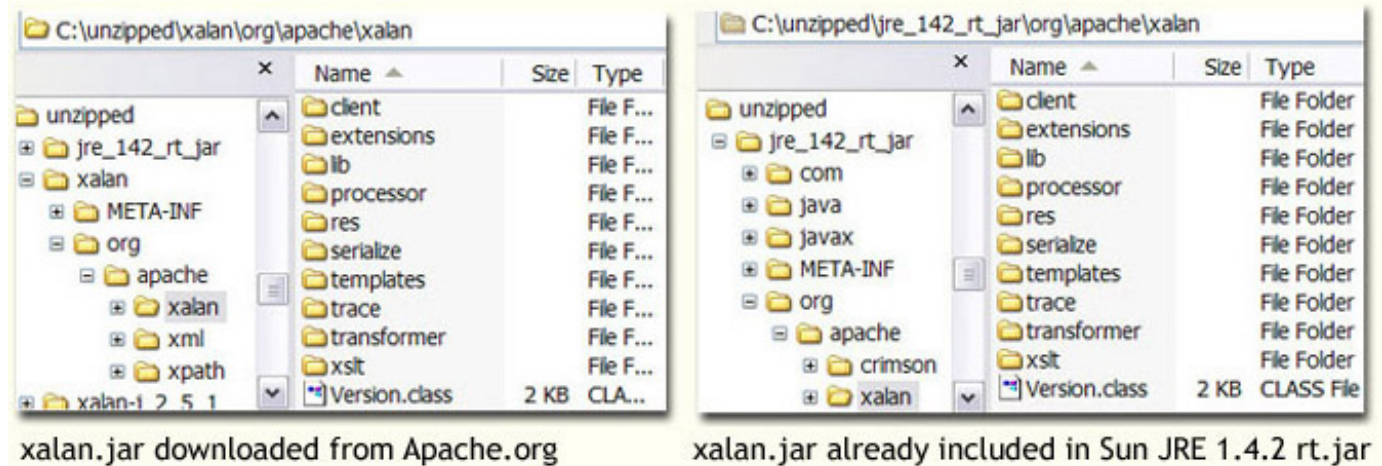
Related Categories: Java, Linux, ColdFusion, Travel, Macromedia

ColdFusion MX's functionality for parsing and transforming XML documents, such as `XMLParse()` and `XMLTransform()`, is handled under the hood by Apache's **Crimson** and **Xalan** implementations, respectively.

Since Crimson is generally retired in favor of Xerces as the choice of XML Parsers, you may wish to download Xerces jar into the CF classpath (WEB-INF/lib) for programmatic access as a java object in your CFML.

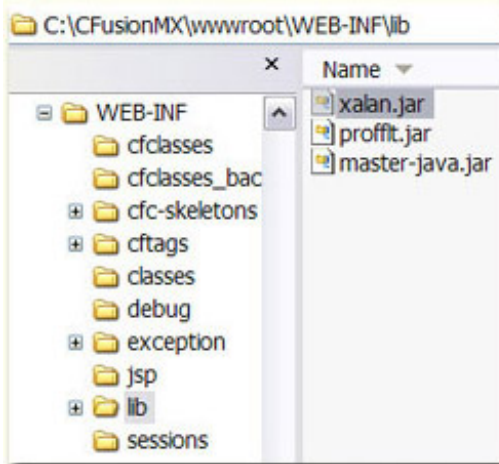
However, ColdFusion MX 6.1 ships with Sun JRE 1.4.2 on most platforms, and that presents a problem if you wish to configure CF for another version of Crimson or Xalan. I would expect that of the two, folks would only be interested in employing a newer Xalan.jar. Simply copying the xalan.jar into {cfroot}/wwwroot/WEB-INF/lib would not be sufficient to access it from CFML because the Sun JRE 1.4.2 already has xalan.jar built into rt.jar (the jvm's runtime classes). Your new xalan.jar would always be loaded after rt.jar, unless configured otherwise, and so you'd always be using the JRE's xalan.jar regardless (Figure 1).

Figure 1



To put a new xalan.jar in front of rt.jar, you have to explicitly put the new xalan.jar in the jvm option `Xbootclasspath`. ColdFusion and the jvm will then "see" the new xalan.jar first when the jvm starts up. To do this, first make a backup copy of {cfroot}/runtime/bin/jvm.config in case something goes wrong, then copy xalan.jar to {cfroot}/wwwroot/WEB-INF/lib (Figure 2), and then make the appropriate modification to the "JVM Args" field in the ColdFusion Administrator's Java and JVM page.

Figure 2



xalan.jar added to ColdFusion's default jar path in WEB-INF/lib

Specifically, add the path to xalan.jar to a new jvm arg "Xbootclasspath/p" which shouldn't be confused with the "Xbootclasspath/a" option already there (Figure 3). The Xbootclasspath/p option is described as "*Applications ... use this option for the purpose of overriding a class in rt.jar*", but also consider the **warning** from Sun regarding this practice, "*Applications [doing this] should not be deployed as doing so would contravene the Java 2 Runtime Environment binary code license*". So use the option at your discretion. You can use the {application.home} syntax when adding to the paths.

Figure 3

JVM Arguments

```
-server -Dsun.io.useCanonCaches=false -Xbootclasspath/p:"{application.home}/wwwroot/WEB-INF/lib/xalan.jar" -Xbootclasspath/a:"{application.home}/../lib/webchartsJava2D.jar" -
```

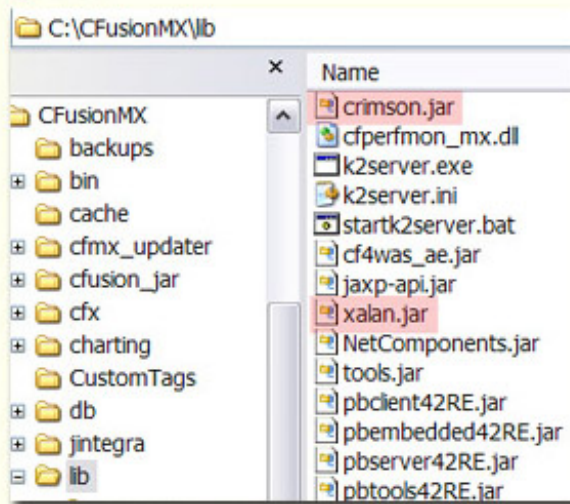
Specify any specific JVM initialization options separated by space(s).

Non-standard option -Xbootclasspath/p set with path to new xalan.jar

Once this is done, restart ColdFusion. If it won't start then there maybe something wrong with the JVM Arg modification, so restore jvm.config from backup and try again. If successful, then ColdFusion should be using the new xalan.jar for XMLTransformations, and will also be accessible programmatically directly through Java in your code.

In the event that you change CFMX to use another JRE that does not have xalan.jar or crimson.jar inside the rt.jar file, ColdFusion comes with both of them already in {cfroot}/lib/. Since CF won't have xalan.jar in the JRE already, it will fall back and use the one in the lib directory (Figure 4).

Figure 4



xalan.jar and crimson.jar provided
in case CF is configured for other jvm

The reason why you would want to configure a different Xalan version in the first place is likely a very specific one, such as a recent mailing list thread I've seen regarding minute differences in performance in one way or another. Add a comment if you know of a justifiable reason to do so. Thanks.