

Adding TLD support to the Jetty container

Symptoms

The embedded jetty container starting up for integration tests in a separate module fails to successfully run web apps which use tag libraries. The app server starts successfully however when the test case attempts to hit a jsp which imports the tag libraries the server fails with the following exception. This behaviour is not seen when deploying the webapp to an installed tomcat instance or whe using the jetty maven plugin to start the container.

```
org.apache.jasper.JasperException: Unable to initialize
TldLocationsCache: null
    at
org.apache.jasper.compiler.TldLocationsCache.init(TldLocationsCache.java:253)
    at
org.apache.jasper.compiler.TldLocationsCache.getLocation(TldLocationsCache.java:224)
    at
org.apache.jasper.JspCompilationContext.getTldLocation(JspCompilationContext.java:526)
    at
org.apache.jasper.compiler.Parser.parseTaglibDirective(Parser.java:422)
    at org.apache.jasper.compiler.Parser.parseDirective(Parser.java:492)
    at org.apache.jasper.compiler.Parser.parseElements(Parser.java:1552)
    at org.apache.jasper.compiler.Parser.parse(Parser.java:126)
    at
org.apache.jasper.compiler.ParserController.doParse(ParserController.java:211)
    at
org.apache.jasper.compiler.ParserController.parse(ParserController.java:100)
    at org.apache.jasper.compiler.Compiler.generateJava(Compiler.java:155)
    at org.apache.jasper.compiler.Compiler.compile(Compiler.java:295)
    at org.apache.jasper.compiler.Compiler.compile(Compiler.java:276)
    at org.apache.jasper.compiler.Compiler.compile(Compiler.java:264)
    at
org.apache.jasper.JspCompilationContext.compile(JspCompilationContext.java:563)
    at
org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:303)
    at
org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:314)
    at org.apache.jasper.servlet.JspServlet.service(JspServlet.java:264)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:820)
```

Reason

For whatever reason, Jetty's thread's context class loader is set to something other than the webapp classloader when the TldLocationsCache is first initialized. The code will try to scan the context classloader for all jars in WEB-INF/lib, but if the context classloader is set to something else, the jars won't be found.

Workaround

The workaround is to place the tld files anywhere in WEB-INF. The secondary scan, after the above one, will look for any tld files in WEB-INF and automatically register them.

For more details...

See: <https://jira.codehaus.org/browse/CARGO-651>