

## Infrastructure - Task #3769

Story # 3770 (Closed): member nodes want to be able to use libclient without having to rely on maven

### Jibx-bind dependencies in d1\_common\_java are showing up in d1\_libclient\_java as dependencies

2013-05-17 20:59 - Rob Nahf

<b>Status:</b>	Closed	<b>Start date:</b>	2013-05-17
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Rob Nahf	<b>% Done:</b>	100%
<b>Category:</b>	d1_common_java	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2013.22-Block.3.3	<b>Story Points:</b>	
<b>Milestone:</b>	None		
<b>Product Version:</b>	*		

**Description**

Certain jibx jars are showing up in the dependency tree of d1\_libclient\_java, when these are only used by d1\_common\_java to create the dataone DataTypes from the d1\_schemas. This functionality is never called from d1\_libclient, so creates false-dependencies.

wrt discussion on packaging libclient\_java and it's dependencies as a releases, we want an automatic way to exclude them from the dependency tree, preferably from within d1\_common\_java itself.

JibX tools and its dependencies on Eclipse plugin arch are not needed for the execution of the jibx generated classes.

Maven seems to have mechanisms for excluding dependencies from the transitive dependencies "graph"  
see: <http://maven.apache.org/guides/introduction/introduction-to-optional-and-excludes-dependencies.html>

#### History

##### #1 - 2013-05-17 21:12 - Rob Nahf

- Parent task set to #3770

##### #2 - 2013-05-20 20:16 - Rob Nahf

- Status changed from New to In Progress

- % Done changed from 0 to 70

setting the scope of the jibx dependencies to 'provided', in both d1\_common\_java, and d1\_jibx\_extensions reclassifies most of the unneeded dependencies from libclient's dependency:tree, however, the filter in mvn dependency:tree -Dscope=compile,runtime doesn't work, so they do show in the tree as 'provided' dependencies. Not sure what that does for any other maven plugins that copy them.

Also, the maven-jibx-plugin introduces dependencies on jibx jars, unbelievably...

Will try to filter out these dependencies at the packaging stage.

##### #3 - 2013-05-23 18:13 - Rob Nahf

- Status changed from In Progress to Testing

- % Done changed from 70 to 90

to get rid of the transitive dependencies introduced by maven-jibx-plugin and other plugins added exclusions to the d1\_common\_java dependency declaration in d1\_libclient\_java pom.

```
org.dataone
d1_common_java
${d1_common_java_version}
```

jar

org.jibx  
maven-jibx-plugin

org.apache.maven.plugins  
maven-compiler-plugin

This removes most of the unwanted dependencies.  
Also added miscellaneous exclusions to the copy-dependencies command:

org.apache.maven,org.codehaus.plexus,classworlds

**#4 - 2013-05-23 18:17 - Rob Nahf**

- *Status changed from Testing to In Review*

**#5 - 2013-05-31 16:26 - Rob Nahf**

- *Target version changed from 2013.20-Block.3.2 to 2013.22-Block.3.3*

**#6 - 2013-06-05 22:11 - Rob Nahf**

- *Status changed from In Review to Closed*

- *% Done changed from 90 to 100*

- *translation missing: en.field\_remaining\_hours set to 0.0*

Decided to forego fine-tuning which jars to exclude from the dependency hierarchy and exclusions.