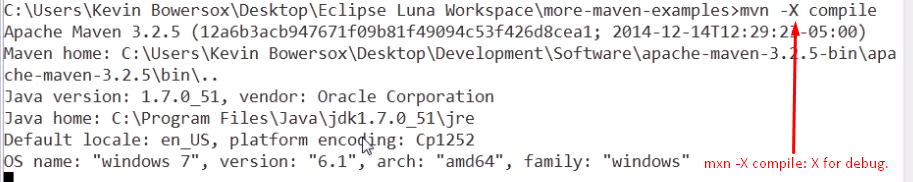
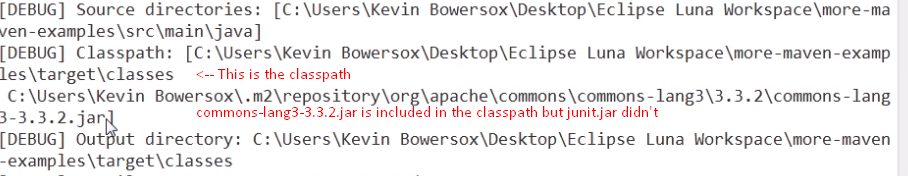
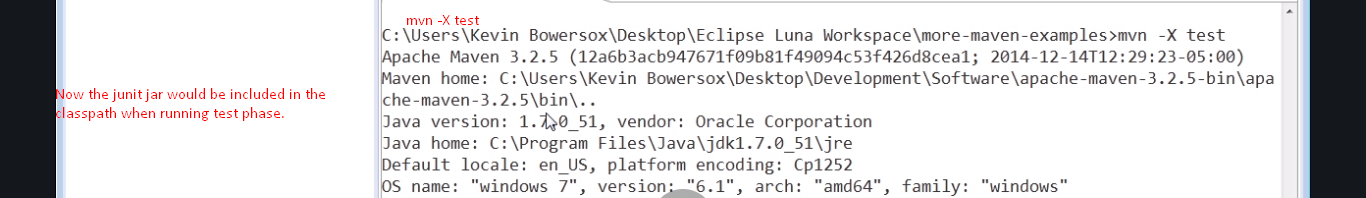
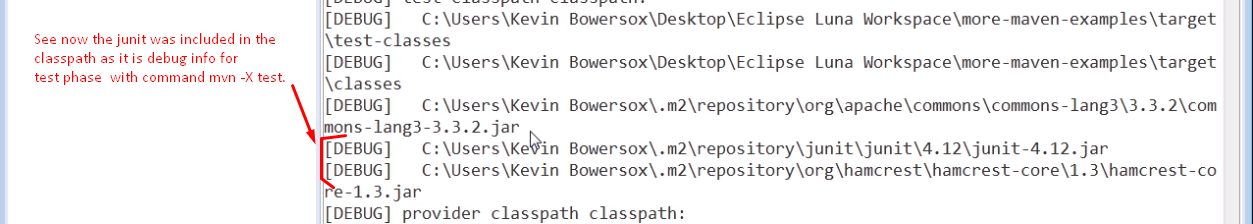
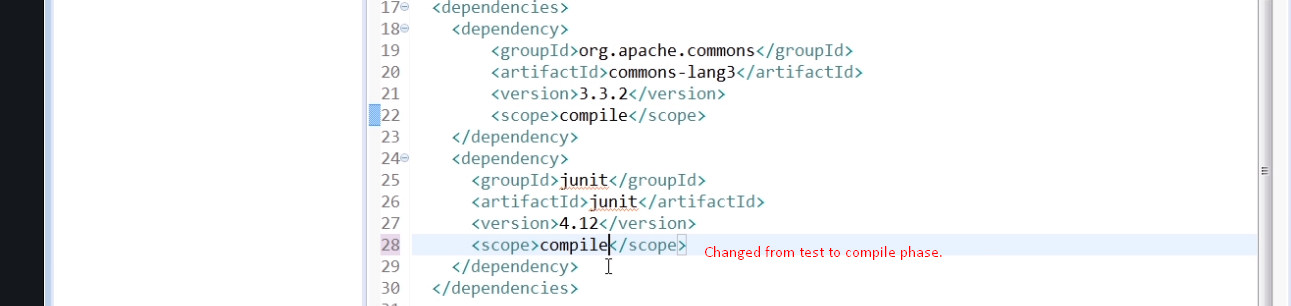
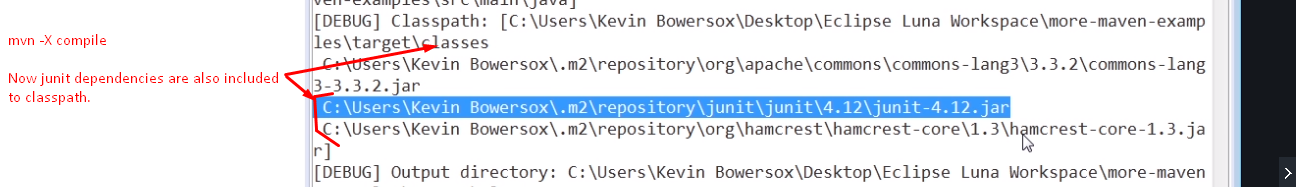
1. **Dependency Scope:** Till when the dependency must be available during maven life cycle.
2. **Let’s see scope in action.**
3. **<scope>:** Default is compile 🡺 <scope>**compile**</scope>
4. compile scope tells that scopes are available on all class pass. So available at building app & runtime.  
   available 🡺 During **build, test, run** phases of our project.   
   
5. 
6. 
7. 
8. 
9. 
10. 
11. **Other** Scopes
    1. **import**
    2. **provided:** Must be provided by JDK or container  
       Dependencies would be available up to build, test, once we deploy or war or jar up our application, those dependencies would be not available in our package.   
       **Example**: Servlet API. Must be in compile classpath but when we run or package, that is provided by the container.
    3. **runtime**: Needed only at runtime not compile time, test time.  
       **Example**: JDBC Driver implementation of JDBC API.
    4. **system**: Very rigid and not recommended.   
       Dependency would be provided by the file system. Path/ directory is specified having dependencies. Then Maven goes there and grabs the dependencies.   
       Problem: If we move pom.xml on different machine, it is going to break because we’re really coupled to that directory structure.