

# Generating a container configuration deployment structure

Ability to create a fully working custom configuration and possibly package some deployables in it; then deliver this configuration as an artifact (ca  
rgo:package).

Example configuration

```

<plugin>
  <groupId>org.codehaus.cargo</groupId>
  <artifactId>cargo-maven2-plugin</artifactId>
  <version>${cargo.version}</version>
  <configuration>
    <container>
      <containerId>tomcat7x</containerId>
      <zipUrlInstaller>

<url>http://archive.apache.org/dist/tomcat/tomcat-7/v7.0.30/bin/apache-tomcat-7.0.30.z
ip</url>
      </zipUrlInstaller>
    </container>
    <configuration>
      <home>${project.build.directory}/catalina-base</home>
      <properties>
        <cargo.servlet.port>${cargo.samples.servlet.port}</cargo.servlet.port>
        <cargo.rmi.port>${cargo.samples.rmi.port}</cargo.rmi.port>

<cargo.tomcat.ajp.port>${cargo.samples.tomcat.ajp.port}</cargo.tomcat.ajp.port>
      </properties>
    </configuration>
    <deployables>
      <deployable>
        <groupId>org.codehaus.cargo</groupId>
        <artifactId>sample-war</artifactId>
        <type>war</type>
      </deployable>
    </deployables>
    <packager>
      <outputLocation>${project.build.directory}/tomcat-packaged</outputLocation>
    </packager>
  </configuration>
  <executions>
    <execution>
      <id>start</id>
      <phase>pre-integration-test</phase>
      <goals>
        <goal>start</goal>
      </goals>
    </execution>
    <execution>
      <id>stop</id>
      <phase>post-integration-test</phase>
      <goals>
        <goal>stop</goal>
      </goals>
    </execution>
    <execution>
      <id>package</id>
      <phase>post-integration-test</phase>
      <goals>
        <goal>package</goal>
      </goals>
    </execution>
  </executions>
</plugin>

```

In the example, after execution of the `cargo:package` goal, a copy of the container and its configuration will be copied to `${project.build.directory}/tomcat-packaged`.

You could then use the [Maven Assembly Plugin](#) to generate an artifact assembly out of it.