

Maven2 Plugin Getting Started

Getting Started

Very quick start

CARGO can be directly run on any existing Maven2 Java EE project (WAR, EAR or other) by running:

```
mvn clean verify
org.codehaus.cargo:cargo-maven2-plugin:run
```

This will create a default [Jetty 7.x installed local container](#) and start it using the Cargo Maven2 plugin with your Maven2 project's deployable (a WAR, for example) deployed to it; so you can run manual tests (as a first introduction).

What is magic is that if you now want to run the same tests with [Tomcat 7.x](#) you simply need to run (in one line):

```
mvn clean verify
org.codehaus.cargo:cargo-maven2-plugin:run
  -Dcargo.maven.containerId=tomcat7x

-Dcargo.maven.containerUrl=http://archive.ap
ache.org/dist/tomcat/tomcat-7/v7.0.16/bin/ap
ache-tomcat-7.0.16.zip
```

That command will automatically download Tomcat 7.0.16 from the specified URL (taking into account any proxy server setting you would have in Maven2/Maven3), instantiate the container, create a local configuration with your application and run it. It will also save the downloaded container in the default directory (see the [Maven2 Plugin Reference Guide](#) for details), so it won't get downloaded when you run the same command twice.

Now, if you want to run this time on [Glassfish 3.x](#) with with the HTTP port set to 9000, run:

```
mvn clean verify
org.codehaus.cargo:cargo-maven2-plugin:run
  -Dcargo.maven.containerId=glassfish3x

-Dcargo.maven.containerUrl=http://download.java.net/glassfish/3.1.1/release/glassfish-3.1.1.zip
  -Dcargo.servlet.port=9000
```

CARGO's main advantage is that the commands and configuration remains the same for any version of any container supported by CARGO -be it Tomcat, Jetty, JBoss, JOnAS, GlassFish, WebLogic, etc.

Like it? Well, keep on reading, then!

More examples

As usual the best way to learn to use a tool is through examples.

We have several [Maven2 Archetypes](#) that contain sample Maven2/Maven3 projects with different use cases for the CARGO plugin, we would really recommend that you check them out. For more details, read here: [Maven2 Archetypes](#).

In addition here are the typical uses cases covered by the plugin:

- [Deploying to a running container](#)
- [Generating a container configuration deployment structure](#)
- [Merging WAR files](#)
- [Starting and stopping a container](#)

The Cargo Maven plugin in detail

Here are the different goals available to call on this plugin:

Goals	Description
-------	-------------

<p><code>cargo: start</code></p>	<p>Start a container. That goal will:</p> <ul style="list-style-type: none"> • If the plugin configuration requires so, installs the container. • If the plugin configuration defines a container with a standalone local configuration, it will create the configuration. • If the plugin configuration contains one or more deployables, it will deploy these to the container automatically. • If the plugin configuration contains no deployables but the project's packaging is Java EE (WAR, EAR, etc.), it will deploy the project's deployable to to the container automatically. • And, of course, start the container. <p>Note: A container that's started with <code>cargo: start</code> will automatically shut down as soon as the parent Maven instance quits (i.e., you see a <code>BUILD SUCCESSFUL</code> or <code>BUILD FAILED</code> message). If you want to start a container and perform manual testing, see our next goal <code>cargo: run</code>.</p>
<p><code>cargo: run</code></p>	<p>Start a container and wait for the user to press <code>CTRL + C</code> to stop. That goal will:</p> <ul style="list-style-type: none"> • If the plugin configuration requires so, installs the container. • If the plugin configuration defines a container with a standalone local configuration, it will create the configuration. • If the plugin configuration contains one or more deployables, it will deploy these to the container automatically. • If the plugin configuration contains no deployables but the project's packaging is Java EE (WAR, EAR, etc.), it will deploy the project's deployable to to the container automatically. • And, of course, start the container and wait for the user to press <code>CTRL + C</code> to stop.
<p><code>cargo: stop</code></p>	<p>Stop a container.</p>
<p><code>cargo: restart</code></p>	<p>Stop and start again a container. If the container was not running before calling <code>cargo: restart</code>, it will simply be started.</p>
<p><code>cargo: configure</code></p>	<p>Create the configuration for a local container, without starting it. Note that the <code>cargo: start</code> and <code>cargo: run</code> goals will also install the container automatically (but will not call <code>cargo: install</code>).</p>

<code>cargo:package</code>	Package the local container .
<code>cargo:daemon-start</code>	Start a container via the daemon. Read more on: Cargo Daemon
<code>cargo:daemon-stop</code>	Stop a container via the daemon. Read more on: Cargo Daemon
<code>cargo:deployer-deploy</code> (aliased to <code>cargo:deploy</code>)	Deploy a deployable to a running container.
<code>cargo:deployer-undeploy</code> (aliased to <code>cargo:undeploy</code>)	Undeploy a deployable from a running container.
<code>cargo:deployer-start</code>	Start a deployable already installed in a running container.
<code>cargo:deployer-stop</code>	Stop a deployed deployable without undeploying it.
<code>cargo:deployer-redeploy</code> (aliased to <code>cargo:redploy</code>)	Undeploy and deploy again a deployable. If the deployable was not deployed before calling <code>cargo:deployer-redeploy</code> (or its alias <code>cargo:redploy</code>) it will simply be deployed.
<code>cargo:uberwar</code>	Merge several WAR files into one.
<code>cargo:install</code>	Installs a container distribution on the file system. Note that the <code>cargo:start</code> goal will also install the container automatically (but will not call cargo:install).
<code>cargo:help</code>	Get help (list of available goals, available options, etc.).

The configuration elements are described in the [Reference Guide](#) section.