

CARGO 1.2.0 now available!

Dear CARGO enthusiasts

First of all, happy New Year!

Since the last release of CARGO a bit more than one month and a half ago, we have had the opportunity of fixing some bugs and adding the functionalities:

- CARGO Maven2/Maven3 plugin fixes and improvements:
 - `ContainerStopMojo` now can resolve dependencies with scope `<test>`, the same way `ContainerRunMojo` and `ContainerStartMojo` can ([CARGO-1069](#) and [CARGO-527](#)).
 - The `functional-tests` artifact of the [Separate Functional Test Module Archetype](#) now uses `packaging=jar` and is compatible with m2e ([CARGO-1065](#)).
 - The CARGO plugin now resolves the required container JARs automatically ([CARGO-975](#)), meaning that if you're for example only using CARGO with the Jetty container than the other containers' JARs will not be downloaded.
- Fixed a bug where the JSR-88 deployers' (used by the GlassFish remote deployer) `undeploy` method ignored the EAR's name ([CARGO-1076](#)).
- All CARGO JBoss containers, from the 10-year-old JBoss 3.x to the brand-new JBoss 7.1.x, now support datasources ([CARGO-426](#)). You can therefore now use CARGO to deploy datasources to any JBoss container; the bigger advantage being that the datasource definitions don't need to change between JBoss versions nor between [all containers that CARGO can deploy datasources to](#).
- The JOnAS remote deployer now works with Windows for XML deployment plans ([CARGO-1067](#)).
- The Resin container has been improved:
 - The Resin management port can be configured ([CARGO-318](#))
 - Resin 3.1.x support has been added ([CARGO-759](#) and [CARGO-494](#))
- Last but not least, the new XML replacement API introduced in CARGO 1.1.4, which avoids having CARGO-specific container configuration files in the distribution, is now used more extensively; as the Jetty ([CARGO-1060](#)), Resin ([CARGO-1075](#)) and Tomcat ([CARGO-1061](#)) container also make use of it.
 - This way, not only we won't have to maintain these CARGO-specific files manually anymore, but also new versions will be supported "out of the box" and CARGO will also work with different flavours of containers.

CARGO 1.2.0 is therefore now ready. Please note that on the API side, this new minor version comes with some behavioural changes and the removal of some elements (which have been deprecated for over 6 months now):

- To achieve automatic resolution of container JARs in the Maven2/Maven3 container, a small change has been done in the `ResourceUtils`: it now loads resources using a class loader (as opposed to a class, which was the case in the past), hence `AbstractLocalConfiguration.RESOURCE_PATH` has been changed from `"/org/codehaus/cargo/container/internal/resources/"` to `"org/codehaus/cargo/container/internal/resources/"` (without the slash at the beginning). If your code uses the same constant, make sure you update it properly.
- The `Deployable.isExpanded*` and `AbstractCopyingInstalledLocalDeployer.setShouldDeployExpanded*s` methods have been removed ([CARGO-905](#) and [CARGO-940](#)). As a replacement:
 - The `Deployable` type has a generic `isExpanded` method.
 - `AbstractCopyingInstalledLocalDeployer` has a generic method `setShouldDeployExpanded` with two arguments.
- File handling methods that manipulate texts now need to specify the encoding ([CARGO-909](#)).
- In the Maven2/Maven3 plugin, the `<zipUrlInstaller>` and `<artifactInstaller>` don't have the `<installDir>` attribute anymore, replaced by `<downloadDir>` and `<extractDir>` attributes ([CARGO-916](#)).
- `TomcatPropertySet.MANAGER_URL` (i.e. `cargo.tomcat.manager.url`) has been removed ([CARGO-965](#)), please use the generic `RemotePropertySet.URI` (i.e. `cargo.remote.uri`) instead.

You can access the full release notes by visiting <https://jira.codehaus.org/secure/ReleaseNote.jspa?version=17088&styleName=Html&projectId=10730>

To use this new version:

- Users of the Java API and/or ANT tasks can download the CARGO uberjar and ANT tasks from the download page: <http://cargo.codehaus.org/Downloads>
- Maven2/Maven3 users simply need to change the plugin's `<version>` to **1.2.0**, and Maven2/Maven3 will download the new version automatically.

Please share your good (and bad) findings on the user@cargo.codehaus.org mailing list.

Enjoy!