Space Details

Key:	CARGO
Name:	Cargo
Description:	Uniform J2EE Container Control System
Creator (Creation Date):	bwalding (Aug 14, 2004)
Last Modifier (Mod. Date):	bwalding (Aug 14, 2004)

Available Pages

- Features
  - Configuration
    - Configuration properties
    - Local Configuration
      - Existing Local Configuration
      - Standalone Local Configuration
    - Runtime Configuration
  - Container
    - Container Classpath
    - Container Factory
    - Container Instantiation
    - Container Start
    - Container Stop
    - Embedded mode
    - Installer
    - Passing system properties
    - Standalone mode
  - Debugging
  - Deployment
    - Deployable
    - Deployer
    - Hot Deployment
    - JSR88
    - Static Deployment
      - Static deployment of EAR
      - Static deployment of expanded WAR
      - Static deployment of WAR
  - Extensions
    - Ant support
    - IntelliJ IDEA Plugin
    - Maven1 plugin
    - Maven2 plugin
    - Netbeans Plugin

    - Module API
- News
- Roadmap
- What for
- Javadocs
- Installation
- Tested on
- Containers
    - Generic JSR88
    - JBoss 3.x
    - JBoss 4.x
    - Jetty 4.x
    - jo 1.x
    - Oc4J 9.x
    - Orion 1.x
    - Orion 2.x
    - Resin 2.x
    - Resin 3.x
    - Tomcat 3.x
    - Tomcat 4.x
    - Tomcat 5.x
    - WebLogic 8.x
- Developers
    - Adding a container
    - Building
    - Contributing
    - Discussions
        - Ant tasks
        - Comparisons with other tools
    - Project Structure
    - Release procedure
    - SVN
- Community
    - Credits
    - IRC
    - License
    - Mailing List Archives
- Misc
    - CenterHeaderFill
    - LeftHeader
    - Navigation
    - QuickLinks
    - RightHeader
- Quick start
- Downloads

    - Archived Downloads
    - Release notes for Cargo 0.5
    - Release notes for Cargo 0.6
    - Release notes for Cargo 0.7
    - Release notes for IntelliJ IDEA Plugin 0.1
- Home 

Features

This page last changed on Oct 08, 2005 by vmassol.

- Configuration — A Configuration specifies how the container is configured
  - Configuration properties — Properties to configure a container (request port, shutdown port, logging level, threads, etc)
  - Local Configuration — A configuration for a container that is running on the local machine where Cargo is executing
    - Existing Local Configuration — Re-use an existing container installation
    - Standalone Local Configuration — Configures your container in a specific directory
  - Runtime Configuration — A configuration for a container that is already started
- Container — A top level interface wrapping a real physical container
  - Container Classpath — How to configure the executing container's classpath
  - Container Factory — Instantiate a container by name
  - Container Instantiation — Create a container instance
  - Container Start — Start a container that is not already running
  - Container Stop — Stop a running container
  - Embedded mode — Use the Container's Java API to control it and execute it in the same JVM where Cargo is running
  - Installer — Installs a container
  - Passing system properties — How to pass system properties that will be available to the container while executing
  - Standalone mode — Use the container based on its installation
- Debugging — Explain how to perform debugging when something doesn't work in Cargo
- Deployment — How to deploy components to a container
  - Deployable — Deployables are archives (WAR, EAR, etc) that can be deployed in the container
  - Deployer — Performs a hot deployment of a Deployable
  - Hot Deployment — Ability to deploy/undeploy Deployables into a running container
  - JSR88 — JSR88-compliant containers support
  - Static Deployment
    - Static deployment of EAR — Deploy an EAR that will be started when the container starts
    - Static deployment of expanded WAR — Deploy an expanded WAR that will be started when the container starts
    - Static deployment of WAR — Deploy a WAR that will be started when the container starts
- Extensions — Extensions are additions to the Cargo core Java API such as build tool plugins, IDE plugins, etc
  - Ant support — Cargo provides Ant tasks to perform all the operations available from the Java API
  - IntelliJ IDEA Plugin
  - Maven1 plugin — Cargo provides a Maven 1 plugin to perform operations available from Ant support.
  - Maven2 plugin — A Maven 2 plugin that wraps the Cargo Java API
  - Netbeans Plugin
- Module API — API to manipulate J2EE archives, including vendor-specific deployment descriptors

Configuration

This page last changed on Nov 26, 2005 by vmassol.

Explanation

A Configuration specifies how the container is configured (logging, security, data sources, location where to put deployables, etc).

Configuration != Installation

The notion of Configuration is different from the notion of Installation. When you install a container, it is usually configured by default too to start and deploy files from where it is installed. However all containers support customizing the configuration and even possibly use a different location where the configuration resides.

(view as slideshow)

Configuration inheritance tree

There are 2 main types of Configurations:

- Local Configuration: You use a local configuration when you're using a Local Container. There are 2 local configuration types: Standalone local configuration and Existing local configuration.
- Runtime Configuration: You use a runtime configuration when you want to access your container as a black box through a remote protocol (JMX, etc). Whereas a local configuration allow you to tune almost all aspects of a container, a runtime configuration only support configuring container properties that can be modified from a distance.

Configuration features

- Configuration properties — Properties to configure a container (request port, shutdown port, logging level, threads, etc)
- Local Configuration — A configuration for a container that is running on the local machine where Cargo is executing
- Runtime Configuration — A configuration for a container that is already started

Custom configuration

The Cargo API allows you to plug your own custom configuration implementation.

You can register your configuration against the `DefaultConfigurationFactory` class. It's optional and only required if you want to let users use the `DefaultConfigurationFactory` class to instantiate your configuration. For example you would write:

```
ConfigurationFactory factory = new DefaultConfigurationFactory();
factory.registerConfiguration("containerIdOfAssociatedContainer", ConfigurationType.STANDALONE,
MyCustomConfiguration.class);
```

Configuration properties

This page last changed on Dec 26, 2005 by vmassol.

Definition

Properties to configure a container (request port, shutdown port, logging level, threads, etc)

Explanations

It is possible to set container configuration properties using the Cargo API. These properties are applied to a Configuration.

Using the Java API you can check if a configuration supports a given property by using {Configuration.getCapability().supportsProperty(String propertyName)}.

For example if we want to check if the configuration supports setting the port property:

```
boolean isPropertySupported =
configuration.getCapability().supportsProperty(ServletPropertySet.PORT);
```

There are 2 kinds of properties:

- General properties
- Container-specific properties. See each container's page for a list of the custom properties it supports.

General properties:

Property name	Java constant to use	Valid values	Description	Example
cargo.servlet.port	ServletPropertySet.PORT	Integer	Port on which the Servlet/JSP container will listen to	"8280"
cargo.hostname	GeneralPropertySet.HOSTNAME	String	Host name on which the container will listen to	"myserver"
cargo.logging	GeneralPropertySet.LOGGING	"low", "medium" or "high"	Level representing the quantity of information we wish to log	"medium"

Example using the Java API

Starting Tomcat 5.x on port 8081:

```
Configuration configuration =
    new CatalinaStandaloneConfiguration("target/tomcat5x"));
configuration.setProperty(ServletPropertySet.PORT, "8081");
[...]
```

Example using the Ant tasks

Starting Tomcat 5.x on port 8081:

```
<cargo containerId="tomcat5x" home="c:/apps/jakarta-tomcat-5.0.29" action="start">
  <configuration>
    <property name="cargo.servlet.port" value="8081"/>
  </configuration>
</cargo>
```

Example using the Maven 2 plugin

Starting Tomcat 5.x on port 8081:

```
<build>
  <plugins>
    <plugin>
      <groupId>org.codehaus.cargo.maven2</groupId>
      <artifactId>cargo-maven2-plugin</artifactId>
      <configuration>
        <container>
          <containerId>tomcat5x</containerId>
          [...]
        </container>
        <configuration>
          <properties>
            <cargo.servlet.port>8081</cargo.servlet.port>
          </properties>
        </configuration>
        [...]
      </configuration>
    </plugin>
  </plugins>
</build>
```

Local Configuration

This page last changed on Nov 26, 2005 by vmassol.

Explanation

A configuration for a container that is running on the local machine where Cargo is executing.

Configuration features

- Existing Local Configuration — Re-use an existing container installation
- Standalone Local Configuration — Configures your container in a specific directory

Existing Local Configuration

This page last changed on Dec 26, 2005 by vmassol.

Definition

Re-use an existing container installation

Explanations

An existing configuration plugs itself onto an existing container installation that exists on your hard disk. This is by opposition to the Standalone Configuration which creates a new container installation from scratch in a directory of your choice.

There are different ways of using an existing configuration:

- By directly instantiating the configuration matching your container. For example:

```
[...]
Configuration configuration = new ResinExistingConfiguration("target/resin3x");

Container container = new Resin3xContainer(configuration);
[...]
```

- By using the `DefaultConfigurationFactory` which automatically maps the right implementation for the container you're using. For example:

```
[...]
ConfigurationFactory factory = new DefaultConfigurationFactory();
Configuration configuration = factory.createConfiguration("resin3x",
    ConfigurationFactory.EXISTING, "c:/apps/resin-3.0.9");

Container container = new Resin3xContainer(configuration);
[...]
```

- By using the `DefaultContainerFactory` which has a constructor for creating a `Container` and a `Configuration` at the same time. For example:

```
[...]
ContainerFactory factory = new DefaultContainerFactory();
Container container = factory.createContainer("resin3x",
    ConfigurationFactory.EXISTING, "c:/apps/resin-3.0.9");
[...]
```

Example using the Ant API

```
<cargo containerId="resin3x" [...]>
  <configuration hint="existing" dir="c:/apps/resin-3.0.9"/>
  [...]
</cargo>
```

Standalone Local Configuration

This page last changed on Dec 26, 2005 by vmassol.

Definition

Configures your container in a specific directory

Explanation

The standalone configuration allows configuring your container so that it is setup to start in a directory you choose (see the configuration page for more general explanations).

There are different ways of using a standalone configuration:

- By directly instantiating the configuration matching your container. For example:

```
[...]
Configuration configuration =
    new Resin3xStandaloneConfiguration("target/resin3x");

Container container = new Resin3xContainer(configuration);
[...]
```

- By using the `DefaultConfigurationFactory` which automatically maps the right implementation for the container you're using. For example:

```
[...]
ConfigurationFactory factory = new DefaultConfigurationFactory();
Configuration configuration = factory.createConfiguration("resin3x",
    ConfigurationType.STANDALONE, "target/resin3x");

Container container = new Resin3xContainer(configuration);
[...]
```

Example using the Ant API

```
<cargo containerId="resin3x" [...]>
  <configuration hint="standalone" dir="target/resin3x"/>
  [...]
</cargo>
```

Runtime Configuration

This page last changed on Nov 26, 2005 by vmassol.

Explanation

A configuration for a container that is already started. The container could be executing on the same machine as where Cargo is executing or anywhere else. The important part is that Cargo is accessing that container using a remote access protocol and the container is considered as a black box.

Container

This page last changed on Nov 22, 2005 by vmassol.

Definition

A top level interface wrapping a real physical container

Explanation

A container is the base concept in Cargo. It represents an existing container. A container is made of a Configuration. There are 2 types of containers:

- Local Container: This is a container is installed on the local machine where Cargo executes. You point to that container by using a path to where it is installed. Local containers can be started and stopped. A local container is always associated with a Local Configuration.
- Remote Container: This is a container that is running on some other machines. It is not under the control of Cargo and cannot be started/stopped by Cargo. The only thing Cargo can do with a remote container is deploy to it using a Remote Deployer. A remote container is always associated with a Runtime Configuration.

Container features

- Container Classpath — How to configure the executing container's classpath
- Container Factory — Instantiate a container by name
- Container Instantiation — Create a container instance
- Container Start — Start a container that is not already running
- Container Stop — Stop a running container
- Embedded mode — Use the Container's Java API to control it and execute it in the same JVM where Cargo is running
- Installer — Installs a container
- Passing system properties — How to pass system properties that will be available to the container while executing
- Standalone mode — Use the container based on its installation

Container support

Container Classpath

This page last changed on Dec 26, 2005 by vmassol.

Definition

How to configure the executing container's classpath

This feature is only available for local containers

Explanation

This topic is not about the classpath requirements to run Cargo (see the Installation page for this); it's about configuring the classpath for the executing container. For most containers, Cargo automatically manages the container's classpath by adding the required container jars to it. However, some containers support being embedded. This is the case of Jetty and the `Jetty4xEmbeddedContainer` implementation class simply starts the container in the running JVM. Thus you'll need to ensure to have the jetty jar + all the other related jars required (jasper-compiler and jasper-runtime jars specifically).

In addition, for all non-embedded container implementations it is possible to add custom jars to the container's execution classpath as shown below.

Example using the Java API

Starting Orion 1.x with Clover jar added to its classpath. For example if you have instrumented your source code with Clover you'll need to add the Clover jar to the classpath.

```
LocalContainer container = new Orion1xLocalContainer(
    new OrionStandaloneLocalConfiguration("target/orion1x");
container.setHome("c:/apps/orion-1.6.0b");

container.setExtraClasspath(new String[] { "libs/clover.jar" });

container.start();
```

Example using the Ant API

Starting Orion 1.x with some additional classpath entries:

```
<cargo containerId="orion1x" home="c:/apps/orion-1.6.0b" action="start">
  <extraClasspath>
    <pathelement location="libs/clover.jar"/>
  </extraClasspath>
</cargo>
```

Container Factory

This page last changed on Nov 22, 2005 by vmassol.

Definition

Instantiate a container by name

Explanation

There are 2 solutions to instantiate a container:

- by explicitely creating a new instance of the container itself (see the Container Instantiation page for more details on creating a container's instance). For example to instantiate a Resin 3.x container:

```
Container container = new Resin3xContainer();
```

- by using the `DefaultContainerFactory` class. The advantage is that you can instantiate the container by name and thus your code can be generic which is nice if you plan to run the same code with different containers. For example, to instantiate a Resin 3.x container:

```
ContainerFactory factory = new DefaultContainerFactory();
Container container = factory.createContainer("resin3x");
```

Container Instantiation

This page last changed on Nov 22, 2005 by vmassol.

Definition

Create a container instance

Explanation

A container instance is created by simply instantiating the Java object implementing the container and passing a Configuration object to it. Each container implementation offers a main Java object wrapping its container and which allows to manipulate the container (start, stop, configure, etc).

The class to use for instantiating a container can be found on each container's documentation page:

- Generic JSR88
- JBoss 3.x
- JBoss 4.x
- Jetty 4.x
- jo 1.x
- Oc4J 9.x
- Orion 1.x
- Orion 2.x
- Resin 2.x
- Resin 3.x
- Tomcat 3.x
- Tomcat 4.x
- Tomcat 5.x
- WebLogic 8.x

In addition it's possible to instantiate a container by name.

Example

```
Container container = new Orion2xLocalContainer(configuration);
[...]
Container container = new Resin3xLocalContainer(configuration);
[...]
Container container = new Weblogic8xLocalContainer(configuration);
[...]
Container container = new TomcatRemoteContainer(configuration);
```

Container Start

This page last changed on Dec 26, 2005 by vmassol.

Definition

Start a container that is not already running

This feature is only available for local containers

Explanation

First you need to create a `Container` instance. This can be done using the container factory or directly by instantating a container implementation class.

Once you have this container instance, starting the container is as simple as calling the `start()` method. Before doing this though you'll need to ensure you have defined the container's `home` (if you're using a container in standalone mode - It's not required for containers in embedded mode).

Of course it you wish to statically deploy archives, you'll need to add deployables to the container.

It is important to note that the `LocalContainer.start()` method will wait until the container is **fully started** before returning.

Example using the Java API

Starting Resin 3.x with no deployable:

```
LocalContainer container = new Resin3xLocalContainer(
  new Resin3xStandaloneLocalConfiguration("target/resin3x"));
container.setHome("c:/apps/resin-3.0.15");

container.start();
```

Example using the Ant API

Before being able to use the Cargo Ant tasks you need to register them against Ant. This is done by using the Ant `<taskdef>` element. See the Ant support page. The action to start the container is specified using the `action="start"` attribute as shown below.

Starting Resin 3.x with no deployable:

```
<cargo containerId="resin3x" home="c:/apps/resin-3.0.15" action="start"/>
```

Container Stop

This page last changed on Dec 26, 2005 by vmassol.

Definition

Stop a running container

This feature is only available for local containers

Note: The stop action waits till the container is fully stopped before returning.

Example using the Java API

Stopping Orion 1.x:

```
LocalContainer container = new Orion1xLocalContainer(
    new OrionStandaloneConfiguration("target/orion1x"));
container.setHome("c:/apps/orion-1.6.0b");

container.stop();
```

Example using the Ant API

Stopping Orion 1.x:

```
<cargo containerId="orion1x" home="c:/apps/orion-1.6.0b" action="stop"/>
```

Embedded mode

This page last changed on Nov 22, 2005 by vmassol.

Definition

Use the Container's Java API to control it and execute it in the same JVM where Cargo is running

Explanation

Cargo provides different container implementations. A Container implementation can be either standalone or embedded. The embedded mode means that Cargo is using directly the container's Java API to control it. If you're using one of the embedded implementation you'll need to ensure that you have the container's jars in your classpath.

Advantages of embbeded mode:

- Faster. There's no need to start a new JVM nor new threads.
- Simpler. There's no need to install the container in a directory

Here is the list of container implementations that support the embedded mode:

- `Jetty4xEmbeddedContainer`: Jetty 4.x implementation

Installer

This page last changed on Nov 22, 2005 by vmassol.

Definition

Installs a container

Explanation

An Installer is meant to install a container. There is currently only a single Installer implementation: `ZipURLInstaller` which downloads a zipped container distribution from a URL and which installs it (i.e. unpacks it) in a specified directory. This is useful if you wish to fully automate a container installation without having to ask the user to manually install a container on their machine.

Of course you don't need to use an Installer and you can rely on the fact that whoever is using Cargo already has a container installed on his machine.

Example

```
Installer installer = new ZipURLInstaller(
    "http://www.caucho.com/download/resin-3.0.9.zip",
    "target/installs");
installer.install();

Container container = new Resin3xContainer(
    new Resin3xStandaloneConfiguration("target/resin3x"));
container.setHomeDir(installer.getHomeDir());
[...]
```

Passing system properties

This page last changed on Dec 26, 2005 by vmassol.

Definition

How to pass system properties that will be available to the container while executing

Explanations

It is sometime useful to pass system properties to the container that is executing. These properties are then available to the code executing in the container.

Example using the Java API

Starting Tomcat 3.x with some System properties set in the container JVM:

```
LocalContainer container = new Tomcat3xLocalContainer(
    new TomcatStandaloneLocalConfiguration("target/tomcat3x"));
container.setHome("c:/apps/jakarta-tomcat-3.3.2");

Map props = new HashMap();
props.put("mypropery", "myvalue");
container.setSystemProperties(props);

container.start();
```

Example using the Ant API

Starting Tomcat 3.x with some System properties set in the container JVM:

```
<cargo containerId="tomcat3x" home="c:/apps/jakarta-tomcat-3.3.2" action="start">
  <sysproperty key="myproperty" value="myvalue"/>
</cargo>
```

Standalone mode

This page last changed on Nov 26, 2005 by vmassol.

Definition

Use the container based on its installation

Explanation

Cargo provides different container implementations. A Container implementation can be either standalone or embedded. The standalone mode configures and controls the container from a proper container installation. Most existing container implementations in Cargo use this standalone mode. For example:

- Resin3xContainer
- Tomcat4xContainer
- Orion2xContainer
- etc

Debugging

This page last changed on Dec 26, 2005 by vmassol.

Definition

Explain how to perform debugging when something doesn't work in Cargo. Indeed it can happen that the container does not start or stop as expected. Or that some deployable does not deploy fine. Or whatever else! Here is a short list of things you can do to try debugging the problem.

Redirecting container output to a file

The `Container.setOutput(File)` API allows you to redirect the container console (stdout) to a file. This is the first file you should check in case of problem.

Example using the Java API

Starting Tomcat 4.x specifying an output console log file:

```
LocalContainer container = new Tomcat4xLocalContainer(
    new CatalinaStandaloneLocalConfiguration("target/tomcat4x"));
container.setHome("c:/apps/jakarta-tomcat-4.1.30");

container.setOutput("target/output.log");

container.start();
```

Use the `container.setAppend(true|false)` method to decide whether the log file is recreated or whether it is appended to, keeping the previous execution logs (by default, the file is recreated at every container start or stop).

Example using the Ant API

Starting Tomcat 4.x specifying an output console log file:

```
<cargo containerId="tomcat4x" home="c:/apps/jakarta-tomcat-4.1.30"
    action="start"
    output="target/output.log"/>
```

Use the `append="true|false"` attribute for controlling the log file creation behavior.

Generating Cargo logs

Some Cargo classes support generation of logs. This is implemented through the notion of {{Monitor}}.

For example to turn on logging monitoring on a `Container` class, you can use:

```
Monitor fileMonitor = new FileMonitor(new File("c:/tmp/cargo.log"), true);
container.setMonitor(fileMonitor);
```

There are several Monitors that are readily available in the Cargo distribution:

- FileMonitor: logs messages to a file
- SimpleMonitor: logs messages to the console (stdout)
- AntMonitor: logs messages using Ant's logging mechanism

Turning on container logs

Cargo is able to configure containers to generate various levels logs. There are 3 levels defined: "low", "medium" and "high" ("medium" is the default value). They represent the quantity of information you wish to have in the generated log file. You can tune container logging by using the following API:

```
container.setProperty(GeneralPropertySet.LOGGING, "medium");
```

The generated log files will then be found in the Working directory you have specified on the container (through the `container.setWorkingDir()` call).

When using the Ant tasks, you can specify the log file by using the `log` attribute. For example:

```
<cargo containerId="resin3x" [...] log="target/cargo.log"/>
```

Deployment

This page last changed on Nov 22, 2005 by vmassol.

Definition

How to deploy components to a container

Explanation

The component to deploy must be wrapped with a Deployable. There are 2 ways to deploy a Deployable:

- By doing a static deployment in a Local Configuration.
- By doing a hot deployment using a Deployer

Deployment features

- Deployable — Deployables are archives (WAR, EAR, etc) that can be deployed in the container
- Deployer — Performs a hot deployment of a Deployable
- Hot Deployment — Ability to deploy/undeploy Deployables into a running container
- JSR88 — JSR88-compliant containers support
- Static Deployment

Deployable

This page last changed on Dec 26, 2005 by vmassol.

Definition

Deployables are archives (WAR, EAR, etc) that can be deployed in the container

Explanation

A `Deployable` class is a wrapper class around a physical archive. `Deployable` are constructed by directly instantiating them (e.g. `new WAR(...)` or `new TomcatWAR(...)`) or by using a `DeployableFactory` (e.g. `DefaultDeployableFactory`). There are 2 generic deployable classes:

- `o.c.c.c.deployable.WAR`
- `o.c.c.c.deployable.EAR`
  There are also some container-specific deployables such as:
- `o.c.c.c.deployable.tomcat.TomcatWAR`
- `o.c.c.c.deployable.jboss.JBossWAR`
  They are there to support container extensions to archives (for example, Tomcat supports `context.xml` files located in your WAR's `META-INF` directory, JBoss allows for a `jboss-web.xml` located in your WAR, etc).

The `DeployableFactory` interface offers a principal method for creating a `Deployable`: `DeployableFactory.createDeployable(String containerId, String deployableLocation, DeployableType type)`. `DeployableType` can be `DeployableType.WAR` or `DeployableType.EAR`.

Once you have a `Deployable` instance wrapping your archive, you'll need to deploy it. This can be done either using Static Deployment or using Hot Deployment.

Example using the Java API

Deploying a WAR in Tomcat 5.x:

```
Container container = new Tomcat5xContainer(
    new CatalinaStandaloneConfiguration("target/tomcat5x"));
container.setHome("c:/apps/tomcat-5.0.29");

WAR war = new WAR("path/to/my.war");

[...]
```

Example using the Generic API

```
[...]

DeployableFactory factory = new DefaultDeployableFactory();
WAR war = factory.createDeployable("tomcat5x", "path/to/my.war",
```

```
      DeployableType.WAR);
```

Example using the Ant API

Statically deploying a WAR in Tomcat 5.x:

```
<cargo containerId="tomcat5x" home="c:/apps/tomcat-5.0.29" action="start">
  <configuration>
    <war warfile="path/to/my.war"/>
  </configuration>
</cargo>
```

Note: In the future there will be an Ant task to support Hot Deployment.

Deployer

This page last changed on Nov 22, 2005 by vmassol.

Definition

Performs a hot deployment of a Deployable

Explanation

TODO: Explain differences between local deployers and remote deployers.

You use a `Deployer` when you wish to deploy a Deployable into a running container (this is known as Hot Deployment). To instantiate a `Deployer` you need to know its class name. A `Deployer` is specific to a container (you can find the class names on the container page listing all containers).

The deployment is done using one of the `Deployer.deploy(...)` APIs. Some `deploy(...)` signatures accept a `DeployableMonitor` which is used to wait till the container has not finished deploying. Cargo currently offers a `URLDeployableMonitor` which waits by polling a provided URL (see below in the example). Whent the URL becomes available the monitor considers that the `Deployable` is fully deployed. In the future, Cargo will provide other `DeployableMonitor` such as a `Jsr88DeployableMonitor`.

Example using the Java API

Example without using a DeployableMonitor

Hot-deploying a WAR on Resin 3.0.9 without waiting for the deployment to finish:

```
Container container = new Resin3xContainer(
    new Resin3xStandaloneConfiguration("target/resin3x"));
container.setHomeDir("c:/apps/resin-3.0.9");

container.start();

DeployableFactory factory = new DefaultDeployableFactory();
WAR war = factory.createDeployable(container.getId(), "path/to/my.war",
    DeployableType.WAR);

Deployer deployer = new ResinDeployer();
deployer.deploy(war);
```

Please note that the `Deployer.deploy()` method call does not wait for the `Deployable` to be fully deployed before returning.

Example using a URLDeployableMonitor

Hot-deploying an WAR on Resin 3.0.9 and waiting for the deployment to finish:

```
Container container = new Resin3xContainer(
    new Resin3xStandaloneConfiguration("target/resin3x"));
container.setHomeDir("c:/apps/resin-3.0.9");

container.start();

DeployableFactory factory = new DefaultDeployableFactory();
WAR war = factory.createDeployable(container.getId(), "path/to/my.war",
    DeployableType.WAR);

Deployer deployer = new ResinDeployer();
deployer.deploy(war, new URLDeployableMonitor("http://server:port/some/url"));
```

The `http://server:port/some/url` must point to a resource that is serviced by the `Deployable` being deployed.

Example using the Ant API

There's currently no Ant task for performing hot deployments.

Hot Deployment

This page last changed on Nov 22, 2005 by vmassol.

Definition

Ability to deploy/undeploy Deployables into a running container

Explanation

Cargo offers a Deployer interface that container implementations can implement to perform hot deployments. At the moment, the following implementations exist:

- `ResinDeployer`
- `JettyDeployer`
- `Jo1xDeployer`

See the Deployer page for more information on how to perform a hot deployment. You can also deploy Deployables before the container is started using Static Deployment.

JSR88

This page last changed on Dec 30, 2005 by vmassol.

Definition

JSR88-compliant containers support

Explanation

> **Warning**
>
> JSR-88 support is not ready to be relased yet and is not included in Cargo 0.7. We're planning to have it in Cargo 0.8. The explanations below are not up to date either and are going to be modified.

Cargo supports JSR 88: J2EE Application Deployment API, allowing it to be used with any JSR88-compliant container.

The core functionality is implemented by the `o.c.c.container.jsr88.JSR88Deployer` class (a Deployer implementation), which acts as a proxy to the JSR88 DeploymentManager. JSR88Deployer assumes its container to implement the `o.c.c.container.jsr88.JSR88Container` interface and relies on it for acquiring all the necessary data (like container URI etc.) as a `o.c.c.container.jsr88.JSR88Info` instance.

If the container type is not known until the run-time, a Generic JSR88 container can be used.

Note: `o.c.c.container.JSR88Container` does **not** extend the `o.c.c.container.Container` interface.

Implementation Limitations

1. It is not possible to stop, undeploy or redeploy a deployable thas was not deployed with the JSR88Deployer being used.

Example

Manipulating a Geronimo container via JSR88 using strongly-typed API:

```
Configuration configuration = new GenericJSR88Configuration(
        new URI("deployer:geronimo:jmx:rmi:///jndi/rmi://localhost:1099/JMXConnector"));

configuration.setProperty(JSR88PropertySet.USERNAME,
        "system");
configuration.setProperty(JSR88PropertySet.PASSWORD,
        "manager");
configuration.setProperty(JSR88PropertySet.DEPLOYTOOL_JAR,
        "geronimo-deploy-tool.jar");
configuration.setProperty(JSR88PropertySet.DEPLOYTOOL_CLASSPATH,
```

```
        "jar-1.jar;jar-2.jar");

Container container = new GenericJSR88Container(configuration);

Deployable deployable = new WAR("test.war");

Deployer deployer = new JSR88Deployer(container);
deployer.deploy(deployable);
```

Manipulating a Geronimo container via JSR88 using generic API:

```
Configuration configuration = (new DefaultConfigurationFactory()).createConfiguration(
        GenericJSR88Container.ID, ConfigurationType.EXISTING,
        new URI("deployer:geronimo:jmx:rmi:///jndi/rmi://localhost:1099/JMXConnector"));

configuration.setProperty(JSR88PropertySet.USERNAME,
        "system");
configuration.setProperty(JSR88PropertySet.PASSWORD,
        "manager");
configuration.setProperty(JSR88PropertySet.DEPLOYTOOL_JAR,
        "geronimo-deploy-tool.jar");
configuration.setProperty(JSR88PropertySet.DEPLOYTOOL_CLASSPATH,
        "jar-1.jar;jar-2.jar");

Container container = new DefaultContainerFactory().
        createContainer(GenericJSR88Container.ID, configuration);

Deployable deployable = new DefaultDeployableFactory().createDeployable(
        GenericJSR88Container.ID, "test.war").getFile(), DeployableType.WAR);

Deployer deployer = new DefaultDeployerFactory().
        createDeployer(container,DefaultDeployerFactory.DEFAULT);
deployer.deploy(deployable);
```

For complete samples, see CARGO-146.

Static Deployment

This page last changed on Nov 22, 2005 by vmassol.

- Static deployment of EAR — Deploy an EAR that will be started when the container starts
- Static deployment of expanded WAR — Deploy an expanded WAR that will be started when the container starts
- Static deployment of WAR — Deploy a WAR that will be started when the container starts

Static deployment of EAR

This page last changed on Dec 26, 2005 by vmassol.

Definition

Deploy an EAR that will be started when the container starts

Example using the Java API

Starting Orion 2.x with an EAR to deploy:

```
Container container = new Orion2xContainer(
    new OrionStandaloneConfiguration("target/orion2x"));
container.setHome("c:/apps/orion-2.0.3");

Deployable ear = container.getDeployableFactory().createEAR(
    "src/data/some.ear");
container.getConfiguration().addDeployable(ear);

container.start();
```

Example using the Ant API

Starting Orion 2.x with an EAR to deploy:

```
<cargo containerId="orion2x" home="c:/apps/orion-2.0.3" action="start">
  <configuration>
    <ear earFile="src/data/some.ear"/>
  </configuration>
</cargo>
```

Static deployment of expanded WAR

This page last changed on Jul 17, 2005 by vmassol.

Definition

Deploy an expanded WAR that will be started when the container starts

Example

```
Container container = new Resin3xContainer(
    new Resin3xStandaloneConfiguration("target/resin3x"));

Deployable war = container.getDeployableFactory().createWAR(
    "some/expanded/war/directory");
container.addDeployable(war);

container.start();
```

Static deployment of WAR

This page last changed on Nov 20, 2005 by vmassol.

Definition

Deploy a WAR that will be started when the container starts

Example

Let's see how to use Jetty 4.x (in embedded mode) with a WAR to deploy in it.

Note: Unlike the other containers, the Jetty integration does not require the Jetty container to be installed. You simply need to add the Jetty jar (`org.mortbay.jetty.jar`), the Servlet API jar (`servletapi.jar`), and the Tomcat Jasper jars (`jasper-compiler.jar`, `jasper-runtime.jar`) to your classpath. Thus the `home` property has not effect.

```
Container container = new Jetty4xEmbeddedContainer(
    new JettyStandaloneConfiguration("target/jetty4x"));

Deployable war = container.getDeployableFactory().createWAR(
    "src/data/some.war");
container.getConfiguration().addDeployable(war);

container.start();
```

Extensions

This page last changed on Nov 18, 2005 by vmassol.

Definition

Extensions are additions to the Cargo core Java API such as build tool plugins, IDE plugins, etc

Explanation

Cargo's core provides a Java API to manipulate containers. In addition the Cargo project also provides extensions to the Cargo's core that make using Cargo easy from your preferred tools. Namely those are:

- Ant support — Cargo provides Ant tasks to perform all the operations available from the Java API
- IntelliJ IDEA Plugin
- Maven1 plugin — Cargo provides a Maven 1 plugin to perform operations available from Ant support.
- Maven2 plugin — A Maven 2 plugin that wraps the Cargo Java API
- Netbeans Plugin

Ant support

This page last changed on Dec 30, 2005 by vmassol.

Definition

Cargo provides Ant tasks to perform all the operations available from the Java API

Explanation

Before using the Ant API you need to register the Cargo Ant tasks into Ant. This is done in the following manner:

```
<taskdef resource="cargo.tasks">
  <classpath>
    <pathelement location="${cargo-core-uberjar-0.7.jar}"/>
    <pathelement location="${cargo-ant-0.7.jar}"/>
  </classpath>
</taskdef>
```

Example

Here's a full example showing how to deploy a WAR, and expanded WAR and an EAR in an Orion 2.x container. Please note that the `output` and `log` attribute are optional. The `property` elements allow you to tune how the container is configured. Here we're telling it to start on port 8180 and to generate the maximum amount of logs in the container `output` file.

```
<taskdef resource="cargo.tasks">
  <classpath>
    <pathelement location="path/to/cargo.jar"/>
  </classpath>
</taskdef>

<cargo containerId="orion2x" home="c:/apps/orion-2.0.3" output="target/output.log"
    log="target/cargo.log" action="start">
  <configuration>
    <property name="cargo.servlet.port" value="8180"/>
    <property name="cargo.logging" value="high"/>
    <war warfile="path/to/my/simple.war"/>
    <war warfile="path/to/my/expandedwar/simple"/>
    <ear earfile="path/to/my/simple.ear"/>
  </configuration>
</cargo>
```

IntelliJ IDEA Plugin

This page last changed on Dec 30, 2005 by vmassol.

Credits

Hendrik Schreiber has written a IntelliJ IDEA 4.5.4 plugin for Cargo.

Installation

You can easily install it through IDEA's plugin manager or download it from our download page and install it manually. Please note that version 0.1 of the plugin will *only* work with IntelliJ IDEA 4.5.4.

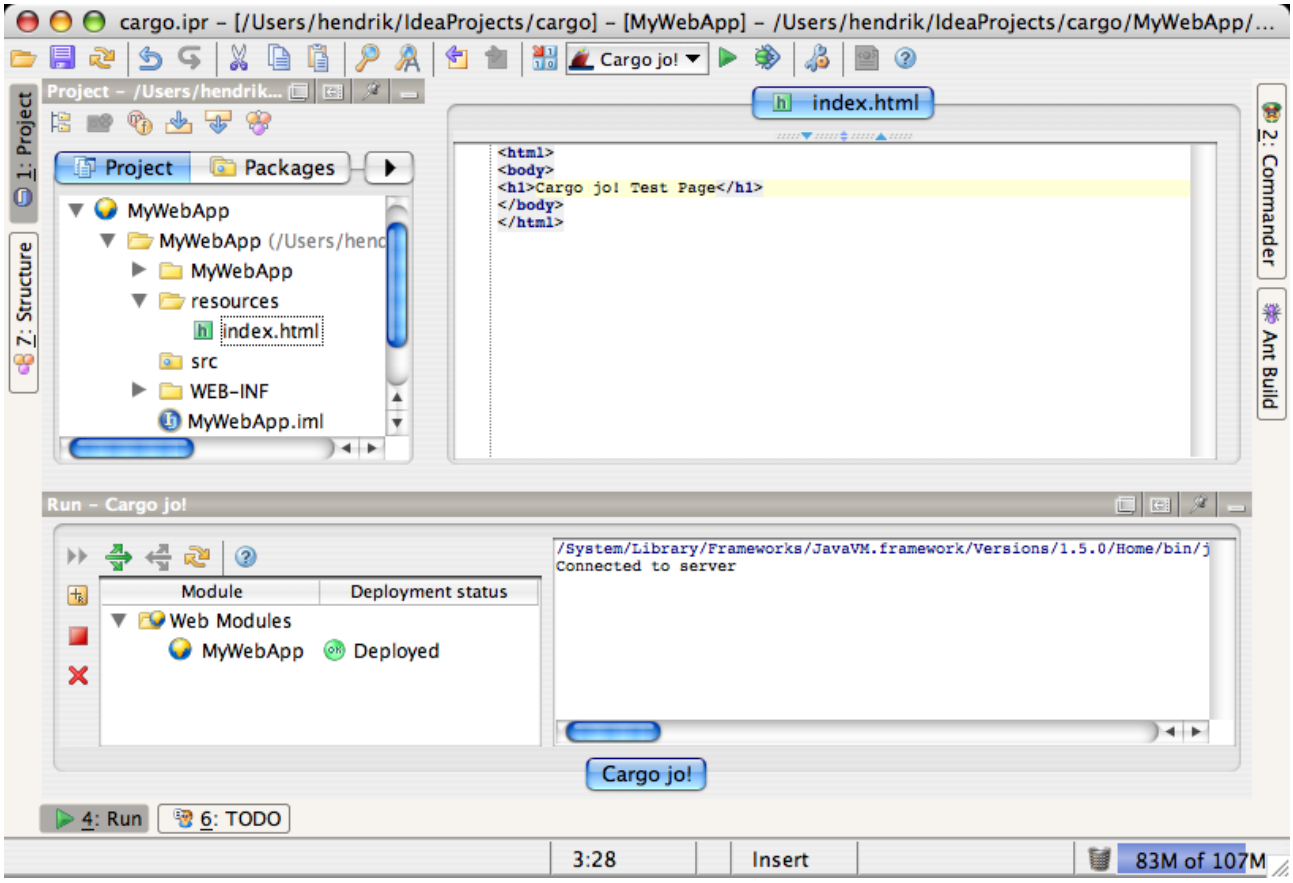
You can download and manually install a working snapshot version for IntelliJ IDEA 5.0.2 or greater from here.

Features

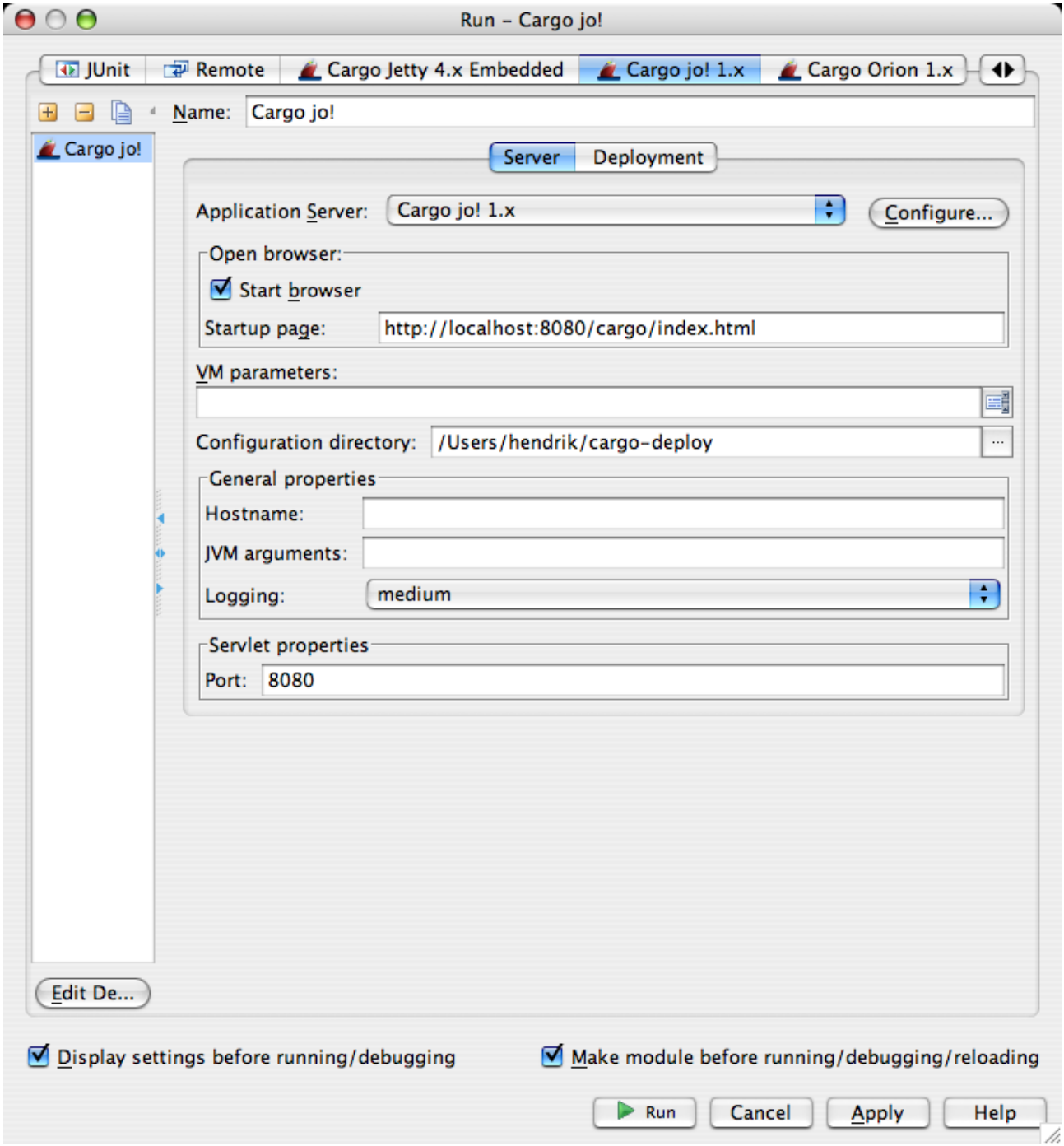
- WAR and EAR deployment to local, standalone containers
- Seamless integration with the IntelliJ IDEA application server API
- Support for Cargo 0.6/IntelliJ IDEA 4.5.4

Screenshots

A running container



An example for a Cargo Config/Run dialog.


Run – Cargo jo!
JUnit
Remote
Cargo Jetty 4.x Embedded
Cargo jo! 1.x
Cargo Orion 1.x
Name:
Cargo jo!
Cargo jo!
Server
Deployment
Application Server:
Cargo jo! 1.x
Configure...
Open browser:
Start browser
Startup page:
http://localhost:8080/cargo/index.html
VM parameters:
Configuration directory:
/Users/hendrik/cargo-deploy
General properties
Hostname:
JVM arguments:
Logging:
medium
Servlet properties
Port:
8080
Edit De...
Display settings before running/debugging
Make module before running/debugging/reloading
Run
Cancel
Apply
Help

[Downloads]

Maven1 plugin

This page last changed on Dec 26, 2005 by vmassol.

Definition

Cargo provides a Maven 1 plugin to perform operations available from Ant support.

> **Maven 2 plugin**
>
> This page is dedicated for the Maven 1 plugin. There's also a Maven2 plugin

Installation

To automatically install the plugin, type the following on a single line:

```
maven plugin:download
  -DgroupId=cargo
  -DartifactId=cargo-maven-plugin
  -Dversion=X.X(.X)
```

where X.X(.X) is the release number you want to install (0.6 is the first available).

Plugin goals

goal	description
cargo:start	Starts all containers defined by the cargo.containers property
cargo:startAndWait	Starts all containers defined by the cargo.containers property and wait for user's input. Without this, Maven stops and the started containers stop as the JVM exits
cargo:stop	Stops all containers defined by the cargo.containers property

Plugin properties

Property name	Required?	Description	Default
cargo.containers		A list of containers ids that specifies on which containers to apply the goal. If this property is empty the Maven plugin will do nothing. You must define for each containerName the	*empty*

		corresponding settings.	
cargo.zipUrlInstaller.<*installerId*>.installUrl		URL from where the container archive can be downloaded.	*empty*
cargo.zipUrlInstaller.<*installerId*>.installDir		The path where the container will be downloaded and extracted.	*empty*
cargo.proxy.host		Proxy hostname (IP or server name).	`${maven.proxy.host}`
cargo.proxy.port		Proxy port.	`${maven.proxy.port}`
cargo.proxy.user		Proxy username.	`${maven.proxy.user}`
cargo.proxy.password		Proxy password.	`${maven.proxy.password}`
cargo.proxy.excludehost		A list of hosts to bypass the proxy on (if any). These should be separated with the vertical bar character '|'. Only in Java 1.4 does FTP use this list.	*empty*
cargo.container.<*containerName*>.containerId		The container id. Valid values are: `jo1x | resin2x | resin3x | orion1x | orion2x | oc4j9x | tomcat3x | tomcat4x | tomcat5x | weblogic8x`.	*empty*
cargo.container.<*containerName*>.home		The path where the container is installed.	*empty*
cargo.container.<*containerName*>.zipUrlInstaller		The id of a zipUrLInstaller.	*empty*
cargo.container.<*containerName*>.output		The path for the file to which output of the container should be written.	*empty*
cargo.container.<*containerName*>.log		The path for the cargo log file.	*empty*
cargo.container.<*containerName*>.timeout		Timeout (in milliseconds) to wait to see if the container is started/stopped.	*empty*
cargo.container.<*containerName*>.deployables		A list of deployable Ids. Each deployable must be defined using the deployables settings.	*empty*
cargo.container.<*containerName*>.config.type		The type to differentiate	*empty*

		the configuration from others for the specified container.Currently the only type supported by cargo is : standalone.	
cargo.container.<*containerName*>.config.dir		The home directory for the configuration of the container.	*empty*
cargo.container.<*containerName*>.config.standalone.servletport		Tomcat servlet port Servlet/JSP container will listen to.	*empty*
cargo.container.<*containerName*>.config.standalone.hostname		Host name on which the container will listen to.	*empty*
cargo.container.<*containerName*>.config.standalone.logging		Level of logging setting the quantity of information we wish to log. Valid values are low | medium | high.	*empty*
cargo.container.<*containerName*>.config.standalone.javaopts		Java options to be used when starting/stopping containers.	*empty*
cargo.container.<*containerName*>.start.output		The path for the file to which output of the container should be written when it starts.	*empty*
cargo.container.<*containerName*>.start.log		The path for the cargo log file when the start action is called for this container.	*empty*
cargo.container.<*containerName*>.stop.output		The path for the file to which output of the container should be written when it stops.	*empty*
cargo.container.<*containerName*>.stop.log		The path for the cargo log file when the stop action is called for this container.	*empty*
cargo.container.<*containerName*>.sysproperties		A list of system properties to be passed to the container, separated by space. Each property is described in a sub property. Example: cargo.container.tomcat.sysproperties = sysprop1 sysprop2	*empty*

		sysprop1 = foo sysprop2 = bar	
cargo.container.<*containerName*>.config.standalone.orion.rmi.port		Orion server RMI port server. (Orion 1x, 2x or Oc4j 9x)	*empty*
cargo.container.<*containerName*>.config.standalone.tomcat.shutdown.port		TCP/IP port number on which this server waits for a shutdown command. (Tomcat 4x or 5x)	*empty*
cargo.deployable.<*deployableId*>.type		Deployable type : war | ear.	*empty*
cargo.deployable.<*deployableId*>.file		Absolute path to the deployable file (or the expanded webapp directory).	*empty*

Be Careful
Exactly one of `cargo.container.<containerName>.home` and `cargo.container.<containerName>.zipUrlInstaller` must defined.

Samples

All properties sample (incoherent settings)

```
cargo.containers = myresin,myorion,myjetty
cargo.zipUrlInstaller.myresin.installUrl = http://www.caucho.com/download/resin-3.0.9.zip
cargo.zipUrlInstaller.myresin.installDir = $\{maven.build.dir\}/installs
cargo.proxy.host = myproxy.mycompany.com
cargo.proxy.port = 1080
cargo.proxy.user = vmassol
cargo.proxy.password = somepassword
cargo.proxy.excludehosts = fozbot.corp.sun.com\|\*.eng.sun.com
cargo.container.myresin.containerId = resin3x
cargo.container.myresin.home = c:/apps/resin/
cargo.container.myContainer.zipUrlInstaller = myContainerInstallerId
cargo.container.myContainer.output = $\{maven.build.dir\}/myContainer/logs/container.log
cargo.container.myContainer.log = $\{maven.build.dir\}/myContainer/logs/cargo.log
cargo.container.myContainer.timeout = 120000
cargo.container.myContainer.deployables = myEarId, myWarId
cargo.container.myContainer.config.type = standalone
cargo.container.myContainer.config.dir = $\{maven.build.dir\}/myContainer/config
cargo.container.myContainer.config.standalone.servlet.port = 8280
cargo.container.myContainer.config.standalone.hostname = myserver
cargo.container.myContainer.config.standalone.logging = high
cargo.container.myContainer.config.standalone.jvmargs = -Xmx64m -Xms2m
cargo.container.myContainer.start.output =
$\{maven.build.dir\}/myContainer/logs/container-start.log
cargo.container.myContainer.start.log = $\{maven.build.dir\}/myContainer/logs/cargo-start.log
cargo.container.myContainer.stop.output =
$\{maven.build.dir\}/myContainer/logs/container-stop.log
cargo.container.myContainer.stop.log = $\{maven.build.dir\}/myContainer/logs/cargo-stop.log
cargo.container.myContainer.config.standalone.orion.rmi.port = 25791
cargo.container.myContainer.config.standalone.tomcat.shutdown.port = 8205
cargo.deployable.myDeployableId.type = war
```

```
cargo.deployable.myDeployableId.file = $\{maven.war.build.dir\}/$\{maven.war.final.name\}
```

Minimal settings to start and stop a container

Example with tomcat already installed:

```
<preGoal name="cargo:start">
  <ant:mkdir dir="${maven.build.dir}/myTomcat/config"/>
</preGoal>
```

```
cargo.containers = myTomcat

cargo.container.myTomcat.containerId = tomcat5x
cargo.container.myTomcat.home = C:/Programs/web/jakarta-tomcat-5.0
cargo.container.myTomcat.config.type = standalone
cargo.container.myTomcat.config.dir = ${maven.build.dir}/myTomcat/config
cargo.container.myTomcat.config.standalone.servlet.port = 8280
cargo.container.myTomcat.config.standalone.logging = high
```

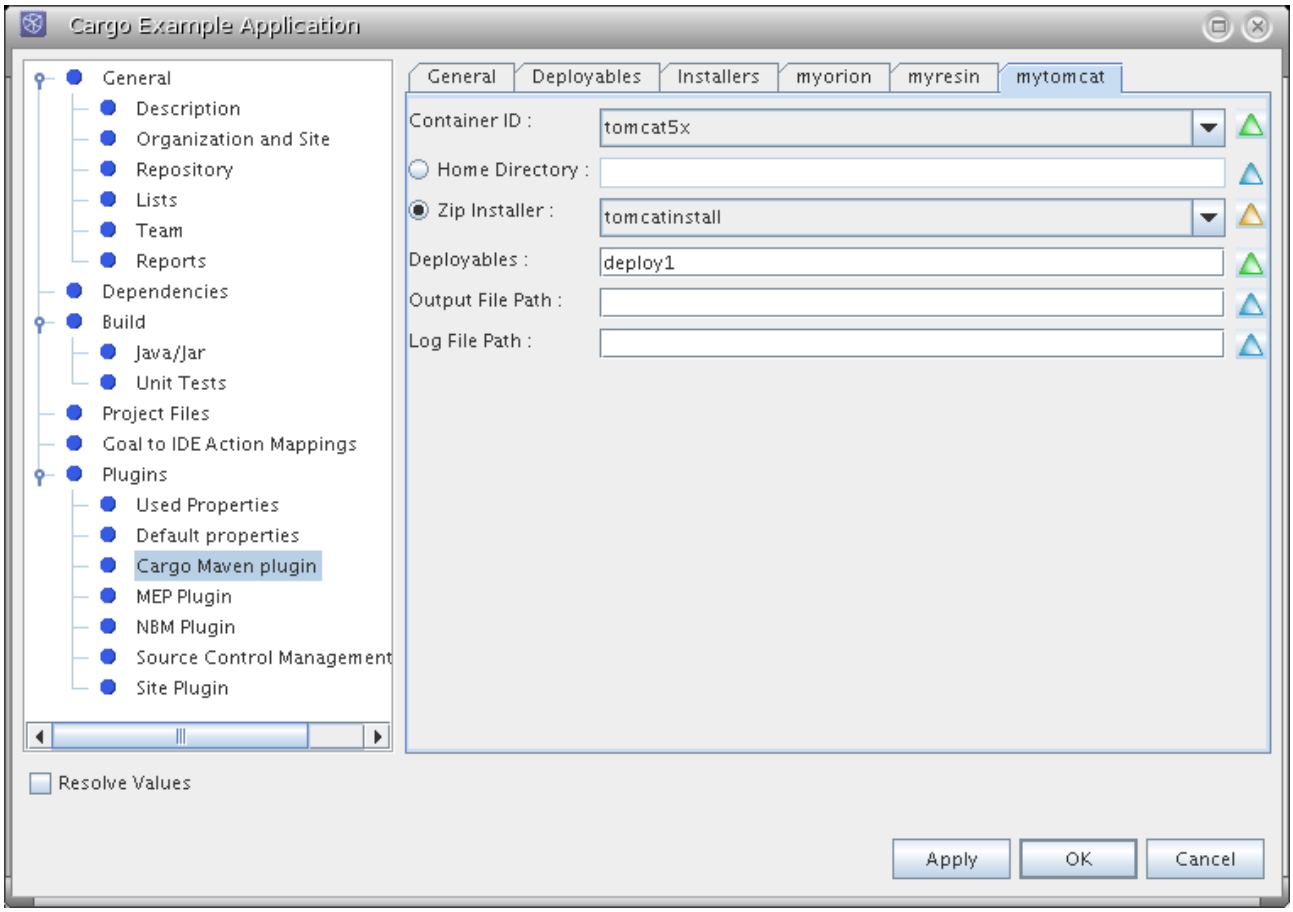
IDE support

Netbeans

There's a Netbeans module working with Mevenide for Netbeans. The module adds a panel into the Maven project's customizer. It eases the setup of the Maven plugin properties and visualizes the current settings.

The current version of the module is 1.0, it works with Mevenide 0.8.1 and later and Netbeans 4.1 and Netbeans 5.0 beta. It can be downloaded here. After download, start Netbeans and install the module through the AutoUpdate Center.

Sample screenshot


Cargo Example Application
General
Description
Organization and Site
Repository
Lists
Team
Reports
Dependencies
Build
Java/Jar
Unit Tests
Project Files
Goal to IDE Action Mappings
Plugins
Used Properties
Default properties
Cargo Maven plugin
MEP Plugin
NBM Plugin
Source Control Management
Site Plugin
General
Deployables
Installers
myorion
myresin
mytomcat
Container ID :
tomcat5x
Home Directory :
Zip Installer :
tomcatinstall
Deployables :
deploy1
Output File Path :
Log File Path :
Resolve Values
Apply
OK
Cancel

Maven2 plugin

This page last changed on Dec 30, 2005 by vmassol.

A Maven 2 plugin that wraps the Cargo Java API

Functional tests

The usage of Cargo for executing functional tests on a container do not need this m2 plugin. You should directly use the Cargo Java API from your Java unit test classes (JUnit, TestNG, etc), as described on http://tinyurl.com/btmwa.

Installation

The Cargo m2 plugin is currently hosted on a private repository on codehaus. This repository is currently not synced to ibiblio so you'll need the following `<pluginRepository>` definition in your `pom.xml` if you want to use the Cargo plugin:

```
<pluginRepositories>
  <pluginRepository>
    <id>cargo m2 release repository</id>
    <url>http://cargo.codehaus.org/dist2</url>
    <releases>
      <enabled>true</enabled>
    </releases>
  </pluginRepository>
  <pluginRepository>
    <id>cargo m2 snapshot repository</id>
    <url>http://cargo.codehaus.org/dist2-snapshot</url>
    <releases>
      <enabled>true</enabled>
    </releases>
  </pluginRepository>
</pluginRepositories>
```

Features

As usual the best way to learn to use a tool is through examples. We have several sample projects that we use as our internal functional tests suite. We'd really recommend that you check them out. In addition here are the typical uses cases coverer by the plugin:

- Start/stop a container
- Deploy to a running container
- Generate container configuration deployment structure

Goals	Description
`cargo:start`	Start a container and optionally deploy deployables (WAR, EAR, etc)
`cargo:stop`	Stop a container
`cargo:deploy`	Deploy a J2EE archive to a running container

The configuration elements are described in the configuration section.

Start/stop a container

Ability to start/stop a container (possibly deploying some deployables to it as it starts). In this scenario Maven 2 is used as a convenience to easily and quickly start a container.

Example of a minimalist configuration:

```
[...]
  <build>
    <plugins>
      <plugin>
        <groupId>org.codehaus.cargo</groupId>
        <artifactId>cargo-maven2-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
[...]
```

Yes, you've read it right, there's no `<configuration>` element! When you use this setup the Cargo m2 plugin will use a Jetty container by default. You can start the container with `mvn cargo:start` and stop it with `mvn cargo:stop`.

Example of a lightweight configuration:

```
[...]
<configuration>

  <!-- Container configuration -->
  <container>
    <containerId>tomcat5x</containerId>
    <home>c:/apps/jakarta-tomcat-5.0.30</home>
  </container>

  <!-- Configuration to use with the container -->
  <configuration>
    <dir>${project.build.directory}/tomcat5x</dir>
  </configuration>

</configuration>
[...]
```

This minimal configuration allows you to configure a default Tomcat 5.x standalone configuration (when the configuration type is not defined as above, the plugin will use a standalone configuration by default) in `${basedir}/target/resin`.

Example of a full-fledged m2 configuration:

```
[...]
<configuration>

  <!-- Container configuration -->
  <container>
    <containerId>orion2x</containerId>
```

```
      <home>c:/apps/orion-2.0.5</home> or
      <zipUrlInstaller>
        <url>http://www.orionserver.com/distributions/orion2.0.5.zip</url>
        <installDir>${java.io.tmpdir}/cargoinstalls</installDir>
      </zipUrlInstaller>
      <output>${project.build.directory}/orion2x/container.log</output>
      <append>false</append>
      <log>${project.build.directory}/orion2x/cargo.log</log>
    </container>

    <!-- Configuration to use with the container or the deployer -->
    <configuration>
      <type>standalone</type>
      <dir>${project.build.directory}/orion2x</dir>
      <properties>
        <cargo.servlet.port>8080</cargo.servlet.port>
        <cargo.logging>high</cargo.logging>
      </properties>

      <deployables>
        <deployable>
          <groupId>war group id</groupId>
          <artifactId>war artifact id</artifactId>
          <type>war</type>
          <properties>
            <context>optional root context</context>
          </properties>
        </deployable>
        <deployable>
          <groupId>ear group id</groupId>
          <artifactId>ear artifact id</artifactId>
          <type>ear</type>
        </deployable>
        [...]
      </deployables>

    </configuration>

  </configuration>
  [...]
```

This example shows the usage of a standalone configuration for configuring Orion 2.x. Note that it's possible to define deployables in the `<configuration>` element and they'll be deployed before the container starts (this is what we call static deployment). We have also defined some configuration properties to tell Cargo to configure Orion 2.x to start on port 8080 and to output highly verbose logs (useful for debugging).

If you have a container that is already installed and configured, say with other deployables already in there, you may want to use an existing configuration. This done by specifying `<type>existing</type>`. In that case you won't be able to control the configuration from Cargo (like port to use, logging levels, etc) as it'll be defined externally.

Deploy to a running container

Cargo supports deploying to an already running container. This feature is called hot deployment). You call it by using the (**`cargo:deploy`**) goal (e.g. `mvn cargo:deploy`).

Note that you can also do static deployment by simply defining the deployables to deploy in the `<configuration>` element as shown above. In that case the deployables will be deployed before the container starts.

Not all containers have a Deployer implemented

We haven't finished implementing Deployers for all containers yet. Please check if your favorite container has it implemented. If not you'll need to deploy your deployables by defining them in a standalone local configuration as shown in the start/stop a container use case above.

Using a local deployer

A local deployer is a deployer that deploys deployables on a local container (i.e. a container installed on the same machine where the deployer is executing). Thus you'll need to use a local container id in `<containerId>`. You can check that by reviewing the supported container list and selecting the container you wish to use.

Example of doing a local deploy to an existing configuration:

```
[...]
<configuration>

  <!-- Container configuration -->
  <container>
    <containerId>resin3x</containerId>
    <home>c:/apps/resin-3.0.9</home> or
    <zipUrlInstaller>
      <url>http://www.caucho.com/download/resin-3.0.9.zip</url>
      <installDir>${basedir}/target/install</installDir>
    </zipUrlInstaller>
  </container>

  <!-- Configuration to use with the container -->
  <configuration>
    <type>existing</type>
    <properties>
      [...]
    </properties>
  </configuration>

  <!-- Deployer configuration -->
  <deployer>
    <type>local</type>
    <deployables>
      <deployable>
        <groupId>war group id</groupId>
        <artifactId>war artifact id</artifactId>
        <type>war</type>
        <properties>
          <context>optional root context</context>
        </properties>
        <pingURL>optional url to ping to know if deployable is done or not</pingURL>
      </deployable>
      <deployable>
        <groupId>ear group id</groupId>
        <artifactId>ear artifact id</artifactId>
        <type>ear</type>
        <pingURL>optional url to ping to know if deployable is done or not</pingURL>
      </deployable>
      [...]
    </deployables>
  </deployer>

</configuration>
[...]
```

In addition, if your project is of type `<packaging>war</packaging>` or `<packaging>ear</packaging>` the generated artifact will be automatically added to the list of deployables to deploy. You can control the

location of the artifact by using the <deployableLocation> element (it defaults to ${project.build.directory}/${project.build.finalName}.${project.packaging}). In addition if you want to wait for the deployment to be finished you can specify a <pingURL> (none is used by default). Here's an example:

```
<plugin>
  <groupId>org.codehaus.cargo</groupId>
  <artifactId>cargo-maven2-plugin</artifactId>
  <configuration>
    <deployableLocation>$\{project.build.directory\}/$\{project.build.finalName\}.$\{project.packaging\}</de
    <pingURL>http://localhost:port/mycontext/index.html</pingURL>
    [...]
```

Using a remote deployer

A remote deployer is a deployer that deploys deployables on a remote container (i.e. a container that is running and that has been started externally from Cargo). Thus you'll need to use an id for a remote container in <containerId> and a runtime configuration.

Example of doing a remote deploy using a runtime configuration:

```
[...]
<configuration>

  <container>
    <containerId>tomcat5x</containerId>
    <type>remote</type>
  </container>

  <configuration>
    <type>runtime</type>
    <properties>
      <cargo.tomcat.manager.url>http://localhost:8080/manager</cargo.tomcat.manager.url>
      <cargo.tomcat.manager.username>username</cargo.tomcat.manager.username>
      <cargo.tomcat.manager.password>password</cargo.tomcat.manager.password>
    </properties>
  </configuration>

  <deployer>
    <type>remote</type>
    <deployables>
      <deployable>
        <groupId>war group id</groupId>
        <artifactId>war artifact id</artifactId>
        <type>war</type>
        <properties>
          <context>optional root context</context>
        </properties>
        <pingURL>optional url to ping to know if deployable is done or not</pingURL>
      </deployable>
      [...]
    </deployables>
  </deployer>

</configuration>
[...]
```

As you can see the information to connect and do the deployment to the remote container is specified in the runtime configuration (`cargo.tomcat.manager.context`, `cargo.tomcat.manager.username` and `cargo.tomcat.manager.password`). The properties to define are deployer-dependent. (TODO: Add link to reference documentation for specific remote deployers once it exists...)

Generate 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 (**`cargo:package`**).

TODO

Configuration

Top level configuration elements	Description
`<container>`	TODO
`<configuration>`	TODO
`<deployer>`	TODO

container elements	Description
`<containerId>`	TODO
`<zipUrlInstaller>`	TODO
`<output>`	TODO
`<append>`	TODO
`<log>`	The path to a file where logs will be go to. If this element is not specified then all logs will be redirected to the Maven 2 console

TODO

Tips

- Starting mutiple containers conditionally

Starting mutiple containers conditionally

Maven 2 supports the notion of profiles which can be used with Cargo to decide for example when to run tests on a specific container. Here's how you could use the Cargo m2 plugin to that effect:

```
<project>
[...]
  <profiles>
    <profile>
```

```
        <id>tomcat5x</id>
        <build>
          <plugins>
            <plugin>
              <groupId>org.codehaus.cargo.maven2</groupId>
              <artifactId>cargo-maven2-plugin</artifactId>
              <executions>
                <execution>
                  <id>tomcat-execution</id>
                  <!-- Ideally this would be bound to some integration-test-prepare phase but
                       that do not exist yet. See http://jira.codehaus.org/browse/MNG-1628 -->
                  <phase>package</phase>
                  <goals>
                    <goal>start</goal>
                  </goals>
                  <configuration>
                    <wait>false</wait>
                    <container>
                      <containerId>tomcat5x</containerId>
                      <zipUrlInstaller>
                        <url>http://www.apache.org/dist/jakarta/tomcat-5/v5.0.30/bin/jakarta-tomcat-5.0.30.zip
                        <installDir>${installDir}</installDir>
                      </zipUrlInstaller>
                    </container>
                    <configuration>
                      <dir>${project.build.directory}/tomcat5x/container</dir>
                    </configuration>
                  </configuration>
                </execution>
              </executions>
            </plugin>
          </plugins>
        </build>
      </profile>
      <profile>
        <id>orion2x</id>
        <build>
          <plugins>
            <plugin>
              <groupId>org.codehaus.cargo.maven2</groupId>
              <artifactId>cargo-maven2-plugin</artifactId>
              <executions>
                <execution>
                  <id>orion-execution</id>
                  <!-- Ideally this would be bound to some integration-test-prepare phase but
                       that do not exist yet. See http://jira.codehaus.org/browse/MNG-1628 -->
                  <phase>package</phase>
                  <goals>
                    <goal>start</goal>
                  </goals>
                  <configuration>
                    <wait>false</wait>
                    <container>
                      <containerId>orion2x</containerId>
                      <zipUrlInstaller>
                        <url>http://www.orionserver.com/distributions/orion2.0.5.zip</url>
                        <installDir>${installDir}</installDir>
                      </zipUrlInstaller>
                    </container>
                    <configuration>
                      <dir>${project.build.directory}/orion2x/container</dir>
                    </configuration>
                  </configuration>
                </execution>
              </executions>
            </plugin>
          </plugins>
        </build>
      </profile>
    </profiles>
  </project>
```

Then to start the tomcat 5.x container you would type `mvn -P tomcat5x integration-test`. if you want to start both containers you would type `mvn -P tomcat5x,orion2x integration-test`.

If you want to define a profile as the default you can use the <activation> element with an activation strategy. For example if you want a profile to be always on, use:

```
<profiles>
    <profile>
      <id>tomcat5x</id>
      <activation>
        <activeByDefault>true</activeByDefault>
       </activation>
[...]
```

TODO: Show how to share configuration data between profiles (this should work by defining the default config data in the <build> element).

Netbeans Plugin

This page last changed on Nov 22, 2005 by vmassol.

Credits

Milos Kleint has started working on a Netbeans plugin for Cargo.
Update: The initial version was released as part of Mevenide for Netbeans 0.7.

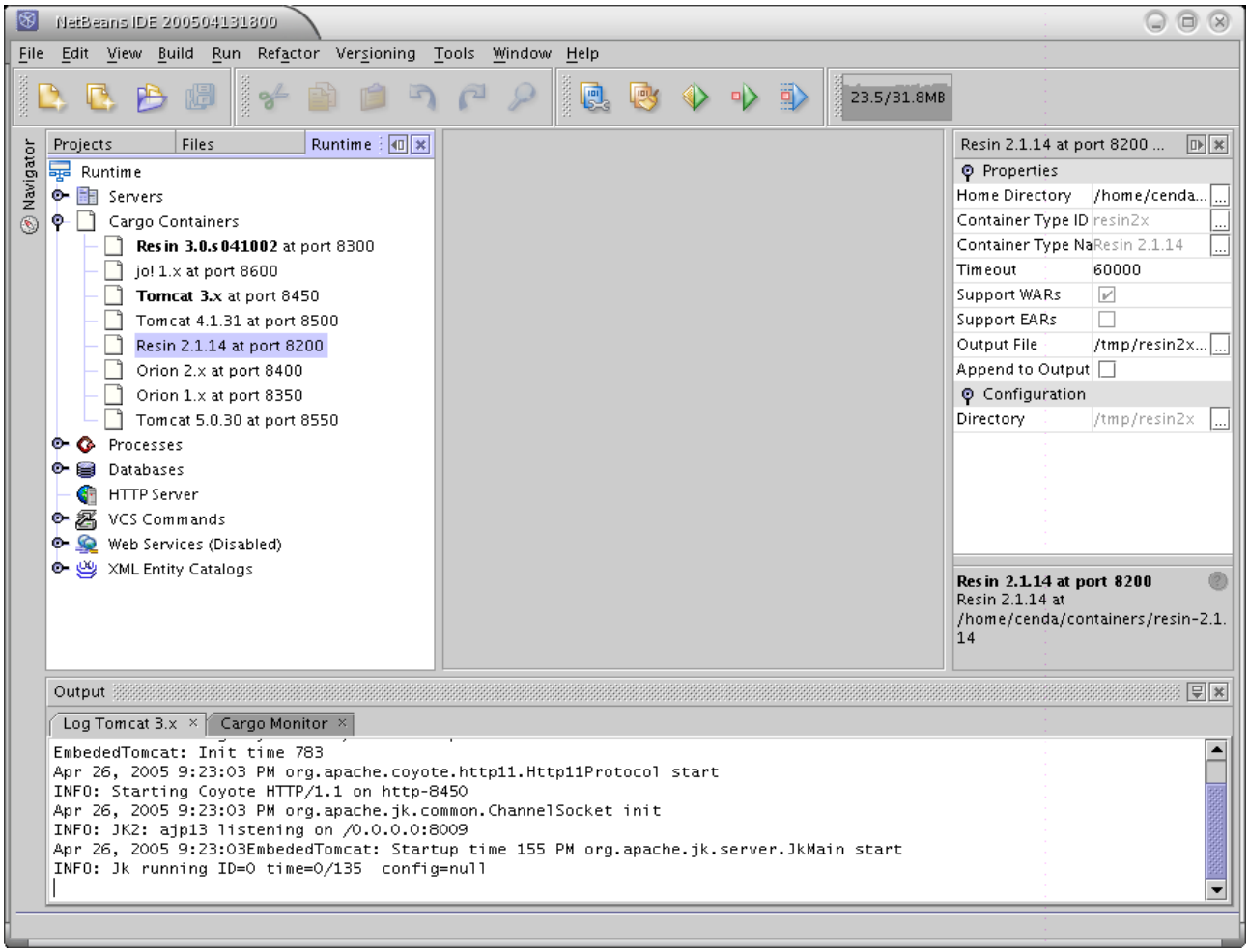
Features

- Addition/removal of container definitions
- Persistence across IDE sessions.
- View container/cargo logs
- WAR deployment for Maven project type (from Mevenide)

Also considered for future:

- Support for Netbeans Ant project types.
- Use maven/ant cargo definitions when deploying project.

Here's a first screenshot of it in action:


NetBeans IDE 200504131800
File Edit View Build Run Refactor Versioning Tools Window Help
23.5/31.8MB
Navigator
Projects Files Runtime
Runtime
Servers
Cargo Containers
Resin 3.0.s041002 at port 8300
jo! 1.x at port 8600
Tomcat 3.x at port 8450
Tomcat 4.1.31 at port 8500
Resin 2.1.14 at port 8200
Orion 2.x at port 8400
Orion 1.x at port 8350
Tomcat 5.0.30 at port 8550
Processes
Databases
HTTP Server
VCS Commands
Web Services (Disabled)
XML Entity Catalogs
Resin 2.1.14 at port 8200 ...
Properties
Home Directory /home/cenda...
Container Type ID resin2x
Container Type Na Resin 2.1.14
Timeout 60000
Support WARs
Support EARs
Output File /tmp/resin2x...
Append to Output
Configuration
Directory /tmp/resin2x
Resin 2.1.14 at port 8200
Resin 2.1.14 at /home/cenda/containers/resin-2.1.14
Output
Log Tomcat 3.x
Cargo Monitor
EmbededTomcat: Init time 783
Apr 26, 2005 9:23:03 PM org.apache.coyote.http11.Http11Protocol start
INFO: Starting Coyote HTTP/1.1 on http-8450
Apr 26, 2005 9:23:03 PM org.apache.jk.common.ChannelSocket init
INFO: JK2: ajp13 listening on /0.0.0.0:8009
Apr 26, 2005 9:23:03EmbededTomcat: Startup time 155 PM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/135 config=null

Module API

This page last changed on Nov 22, 2005 by vmassol.

Definition

API to manipulate J2EE archives, including vendor-specific deployment descriptors

Explanation

This API is located in the `org.codehaus.cargo.module` package and is used internally by the Cargo Java API. You can also use it yourself if you need to manipulate (read/write) J2EE archive files.

It also supports merging two `web.xml` files into one.

See the Javadoc for more details on the API.

Example

```
File warFile = new File("/path/to/simple.war");
WarArchive war = new DefaultWarArchive(warFile);

// Verify existence of class inside the WAR
assertTrue(war.containsClass("test.Test"));

// Verify version of the Servlet specifications used in web.xml
WebXml webXml = war.getWebXml();
assertEquals(WebXmlVersion.V2_3, webXml.getVersion());

// Add a context-param element in web.xml
Element contextParamElement =
  createContextParamElement(doc, "param", "value");
webXml.addContextParam(contextParamElement);
assertTrue(webXml.hasContextParam("param"));

[...]
```

News

This page last changed on Nov 20, 2004 by vmassol.

Confluence RSS Feed (rss_2.0)
(The 15 most recent creations of or modifications to blogposts in space Cargo.)

Roadmap

This page last changed on Apr 25, 2005 by vmassol.

General directions

- Continue adding container support for dynamic deployments
- Support for JSR88 for containers who support it. We can use the JSR88 API to provide a common way of performing deployment/undeployment of WAR/EAR files in target containers.

More information on JSRs potentially useful for Cargo:

- JSR88: This specification defines standard APIs that will enable any deployment tool that uses the deployment APIs to deploy any assembled application onto a J2EE compatible platform. The API will address the three-stage deployment process:
  - Installation - move the properly packaged components to the server
  - Configuration - the resolution of all external dependencies declared by the application
  - Undeployment - removal of the application from the server

- JSR77: The Specification proposes a standard management model for exposing and accessing the management information, operations, and parameters of the Java 2 Platform, Enterprise Edition components. The management model will:
  - Allow rapid development of management solutions for J2EE
  - Provide integration with existing management systems
  - Enable a single management tool to manage multiple vendor implementations of the platform
  - Enable a specific implementation of a platform to use any compliant management tool

Tasks already planned to be implemented

See the JIRA roadmap.

What for

This page last changed on Apr 29, 2005 by vmassol.

Here are some possible use cases for Cargo:

- To start containers for integration and functional tests
- To start containers for applications that require a container to be started (Plugins for IDEs, etc)
- As a framework to manipulate J2EE Module file including container-specific descriptors. For example it can be useful if you wish to implement the JSR88 client side
- To generate container configurations for deployment. For example you may have an application running on Tomcat 5.x and you may want to package a fully working configuration (server.xml, webapps/ dir with your WAR files in there, etc).

Javadocs

This page last changed on Jul 17, 2005 by vmassol.

- Javadoc for Core Util API
- Javadoc for Core Module API
- Javadoc for Core Container API
- Javadoc for Core Generic API
- Javadoc for Ant API

Installation

This page last changed on Nov 14, 2005 by vmassol.

Java API

Cargo offers primarily a Java API. It is meant to be embdedded in your application. You'll need a JDK 1.4+ and the following jars in your classpath:

- Ant 1.5.4 or greater
- Xerces 2.4.0 or greater

Ant

If you use the Cargo Ant tasks, you'll also need to add those jars to the <taskdef> definition (see the Ant support page).

Maven

To use the Cargo plugin for Maven you'll need to install it (see the Maven1 plugin page).

Tested on

This page last changed on Nov 21, 2005 by vmassol.

In this section you can find the test status of the different containers for the different Cargo releases.

Add your own experiences by reporting the configurations that you have tested. You can do this by opening a JIRA issue. Make sure you choose a **`Test`** issue type so that it finds its way on this page! Please also ensure to report your JDK in the **`Tested on JDKs`** field.

Cargo 0.7

jira.codehaus.org (2 issues)			
Summary	**Assignee**	**Reporter**	**Status**
Test on JBoss 4.0.2	Arnaud Heritier	Arnaud Heritier	Closed
Test on Resin 3.0.15	Mark Hobson	Vincent Massol	Closed

All Cargo versions

jira.codehaus.org (11 issues)			
Summary	**Assignee**	**Reporter**	**Status**
Test on WebLogic 8.1 SP3	Vincent Massol	Vincent Massol	Closed
Test on OC4J 9.0.4	Vincent Massol	Vincent Massol	Closed
Test on Tomcat 5.5.3-snapshot	Vincent Massol	Vincent Massol	Closed
Test on Resin 3.0.9	Vincent Massol	Vincent Massol	Closed
Test on Orion 2.0.5	Vincent Massol	Vincent Massol	Closed
Test on Tomcat 4.1.31	Vincent Massol	Vincent Massol	Closed
Test on Orion 2.0.4	Vincent Massol	Vincent Massol	Closed
Test on Resin 3.0.9	Vincent Massol	Vincent Massol	Closed
Test on Resin 2.1.16	Vincent Massol	Vincent Massol	Closed
Test on JBoss 4.0.2	Arnaud Heritier	Arnaud Heritier	Closed
Test on Resin 3.0.15	Mark Hobson	Vincent Massol	Closed

Containers

This page last changed on Nov 22, 2005 by vmassol.

List of supported containers and the extensions that are implemented for each container (Java API, Ant tasks and Maven plugins). The specified version is the Cargo version where the feature was first made available. Click on a container's name to see a detailed list of features it supports.

Container	Java API(version)	Ant tasks(version)	Maven 1 plugin(version)	Maven 2 plugin(version)
JBoss 3.x	0.7	0.7	0.7	0.7
JBoss 4.x	0.7	0.7	0.7	0.7
Jetty 4.x	0.1	???	???	???
jo! 1.x	0.5	0.5	0.5	0.7
OC4J 9.x	0.3	0.3	0.5	0.7
Orion 1.x	0.1	0.1	0.5	0.7
Orion 2.x	0.1	0.1	0.5	0.7
Resin 2.x	0.1	0.1	0.5	0.7
Resin 3.x	0.1	0.1	0.5	0.7
Tomcat 3.x	0.1	0.1	0.5	0.7
Tomcat 4.x	0.1	0.1	0.5	0.7
Tomcat 5.x	0.1	0.1	0.5	0.7
WebLogic 8.x	0.3	0.3	0.5	0.7

We also encourage you to report success and failures on different versions of those containers in the Tested on section.

Generic JSR88

This page last changed on Nov 26, 2005 by vmassol.

Supported Features

Note: JSR-88 support is experimental and should not be used right now. No tests are contained in the Maven build.

Feature name	Java	Ant	Maven	Comment
Container Start	✗	✗	✗	
Container Stop	✗	✗	✗	
Container Instantiation	✓ `o.c.c.container.jsr88.GenericJSR88Container`	✗	✗	This is a Remote Container.
Container Factory	✓ `"genericjsr-88"`	✗	✗	
Container Classpath	✗	✗	✗	
Debugging	✗	✗	✗	
Passing system properties	✗	✗	✗	
Standalone Local Configuration	✗	✗	✗	
Existing local configuration	✓ `o.c.c.container.jsr88.GenericJSR88ExistingConfiguration`	✗	✗	
Static deployment of WAR	✓	✗	✗	
Static deployment of expanded WAR	✓	✗	✗	
Static deployment of EAR	✓	✗	✗	
Standalone mode	✗	✗	✗	
Embedded mode	✗	✗	✗	
Installer	✗	✗	✗	
Hot Deployer - deploy()	✓ `o.c.c.container.jsr88.JSR88Deployer`	✗	✗	
Hot Deployer - undeploy()	✓	✗	✗	
Hot Deployer - start()	✓	✗	✗	
Hot Deployer - stop()	✓	✗	✗	

Supported Configuration properties

Property name	Supported?	Comment
ServletPropertySet.PORT	✖	
GeneralPropertySet.HOSTNAME	✖	
GeneralPropertySet.LOGGING	✖	

Custom configuration properties:

Property name	Java constant to use	Valid values	Description	Example
cargo.jsr88.user	JSR88PropertySet.USERNAME	String	Username to use when acquiring a JSR88 DeploymentManager.	"user"
cargo.jsr88.password	JSR88PropertySet.PASSWORD	String	Password to use when acquiring a JSR88 DeploymentManager.	"password"
cargo.jsr88.deploytool.jar	JSR88PropertySet.DEPLOYTOOL_JAR	String	JAR file to load the JSR88 deployment tool from.	"deploy-tool.jar"
cargo.jsr88.deploytool.classpath	JSR88PropertySet.DEPLOYTOOL_CLASSPATH	String	Extra classpath necessary for the JSR88 deployment tool (**not the container itself**) to function. Semicolon-separated.	"jar-1.jar;jar-2.jar"

JBoss 3.x

This page last changed on Dec 26, 2005 by vmassol.

Supported Features

Feature name	Java	Ant	Maven	Comment
Container Start	✓	✓	✓	
Container Stop	✓	✓	✓	
Container Instantiation	✓ o.c.c.container.jboss.JBoss3xInstalledLocalContainer	✓ <cargo containerId="jboss3x" />	✓ cargo.containers = jboss3x	
Container Factory	✓ "jboss3x"	✓ <cargo containerId="jboss3x" />	✓ cargo.containers = jboss3x	
Container Classpath	✓	✓	✗	
Debugging	✓	✓	✓	
Passing system properties	✓	✓	✗	
Standalone Local Configuration	✓ o.c.c.container.jboss.JBossStandaloneLocalConfiguration	✓	✓	
Existing local configuration	✓ o.c.c.container.jboss.JBossExistingLocalConfiguration	✓	✓	
Static deployment of WAR	✓	✓	✓	
Static deployment of expanded WAR	✓	✓	✓	
Static deployment of EAR	✓	✓	✓	
Standalone mode	✓	✓	✓	
Embedded mode	N/A	N/A	N/A	
Installer	✓	✓	✓	
Hot Deployer - deploy()	✓ o.c.c.container.jboss.JBossDeployer	✗	✗	
Hot Deployer - undeploy()	✗	✗	✗	
Hot Deployer - undeploy()	✗	✗	✗	
Hot Deployer - start()	✗	✗	✗	

Hot Deployer - stop()	✗	✗	✗	

Supported Configuration properties

Property name	Supported?	Comment
ServletPropertySet.PORT	✓	
GeneralPropertySet.HOSTNAME	✗	
GeneralPropertySet.LOGGING	✓	

JBoss 4.x

This page last changed on Dec 26, 2005 by vmassol.

Supported Features

Feature name	Java	Ant	Maven	Comment
Container Start	✓	✓	✓	
Container Stop	✓	✓	✓	
Container Instantiation	✓ o.c.c.container.jboss.JBoss4xInstalledLocalContainer	✓ <cargo containerId="jboss4x" />	✓ = jboss4x	
Container Factory	✓ "jboss4x"	✓ <cargo containerId="jboss4x" />	✓ = jboss4x	
Container Classpath	✓	✓	✗	
Debugging	✓	✓	✓	
Passing system properties	✓	✓	✗	
Standalone Local Configuration	✓ o.c.c.container.jboss.JBossStandaloneLocalConfiguration	✓	✓	
Existing local configuration	✓ o.c.c.container.jboss.JBossExistingLocalConfiguration	✓	✓	
Static deployment of WAR	✓	✓	✓	
Static deployment of expanded WAR	✓	✓	✓	
Static deployment of EAR	✓	✓	✓	
Standalone mode	✓	✓	✓	
Embedded mode	N/A	N/A	N/A	
Installer	✓	✓	✓	
Hot Deployer - deploy()	✓ o.c.c.container.jboss.JBossDeployer	✗	✗	
Hot Deployer - undeploy()	✗	✗	✗	
Hot Deployer - start()	✗	✗	✗	
Hot Deployer - stop()	✗	✗	✗	

Supported Configuration properties

Property name	Supported?	Comment
ServletPropertySet.PORT		
GeneralPropertySet.HOSTNAME		
GeneralPropertySet.LOGGING		

Jetty 4.x

This page last changed on Nov 26, 2005 by vmassol.

Supported Features

Feature name	Java	Ant	Maven	Comment
Container Start	✓	✗	✗	
Container Stop	✓	✗	✗	
Container Instantiation	✓ `o.c.c.container.jetty.Jetty4xEmbeddedLocalContainer`	✗	✗	
Container Factory	✓ `"jetty4x"`	✗	✗	
Container Classpath	✓	✗	✗	
Debugging	✓	✗	✗	
Passing system properties	✓	✗	✗	
Standalone Local Configuration	✓ `o.c.c.container.jetty.JettyStandaloneLocalConfiguration`	✗	✗	
Existing local configuration	✗	✗	✗	
Static deployment of WAR	✓	✗	✗	
Static deployment of expanded WAR	✓	✗	✗	
Static deployment of EAR	N/A	N/A	N/A	
Standalone mode	✗	✗	✗	
Embedded mode	✓	✗	✗	
Installer	✗	✗	✗	
Hot Deployer - deploy()	✓ `o.c.c.container.jetty.JettyDeployer`	✗	✗	
Hot Deployer - undeploy()	✗	✗	✗	
Hot Deployer - start()	✗	✗	✗	
Hot Deployer - stop()	✗	✗	✗	

Supported Configuration properties

Property name	Supported?	Comment
ServletPropertySet.PORT	✓	
GeneralPropertySet.HOSTNAME	✗	
GeneralPropertySet.LOGGING	✓	

jo 1.x

This page last changed on Dec 26, 2005 by vmassol.

Supported Features

Feature name	Java	Ant	Maven	Comment
Container Start	✓	✓	✓	
Container Stop	✓	✓	✓	
Container Instantiation	✓ `o.c.c.container.jo.Jo1xContainer`	✓ `<cargo containerId="jo1x">`	✓ `cargo.containers = jo1x`	
Container Factory	✓ `"jo1x"`	✓ `<cargo containerId="jo1x">`	✓ `cargo.containers = jo1x`	
Container Classpath	✓	✓	✗	
Debugging	✓	✓	✓	
Passing system properties	✓	✓	✗	
Standalone Local Configuration	✓ `o.c.c.container.jo.Jo1xStandaloneConfiguration`	✓	✓	
Existing local configuration	✗	✗	✗	
Static deployment of WAR	✓	✓	✓	
Static deployment of expanded WAR	✓	✓	✓	
Static deployment of EAR	N/A	N/A	N/A	
Standalone mode	✓	✓	✓	
Embedded mode	✗	✗	✗	
Installer	✓	✓	✓	
Hot Deployer - deploy()	✓ `o.c.c.container.jo.Jo1xDeployer`	✗	✗	
Hot Deployer - undeploy()	N/A	N/A	N/A	Can jo! support this?
Hot Deployer - start()	N/A	N/A	N/A	Can jo! support this?
Hot Deployer - stop()	N/A	N/A	N/A	Can jo! support this?

Supported Configuration properties

Property name	Supported?	Comment
ServletPropertySet.PORT		
GeneralPropertySet.HOSTNAME		
GeneralPropertySet.LOGGING		

Notes

Currently only jo! >= 1.1 is supported.

Oc4J 9.x

This page last changed on Dec 26, 2005 by vmassol.

Supported Features

Feature name	Java	Ant	Maven	Comment
Container Start	✓	✓	✓	
Container Stop	✓	✓	✓	
Container Instantiation	✓ `o.c.c.container.orion.Oc4j9xStandaloneContainer`	✓ `<cargo containerId="oc4j9x"/>`	✓ `cargo.containers = oc4j9x`	
Container Factory	✓ `"oc4j9x"`	✓ `<cargo containerId="oc4j9x"/>`	✓ `cargo.containers = oc4j9x`	
Container Classpath	✓	✓	✗	
Debugging	✓	✓	✓	
Passing system properties	✓	✓	✗	
Standalone Local Configuration	✓ `o.c.c.container.orion.OrionStandaloneLocalConfiguration`	✓	✓	
Existing local configuration	✗	✗	✗	
Static deployment of WAR	✓	✓	✓	
Static deployment of expanded WAR	✓	✓	✓	
Static deployment of EAR	✓	✓	✓	
Standalone mode	✓	✓	✓	
Embedded mode	✗	✗	✗	
Installer	✓	✓	✓	
Hot Deployer - deploy()	✗	✗	✗	
Hot Deployer - undeploy()	N/A	N/A	N/A	Can Oc4j9x support this?
Hot Deployer - start()	N/A	N/A	N/A	Can Oc4j9x support this?
Hot Deployer - stop()	N/A	N/A	N/A	Can Oc4j9x support this?

Supported Configuration properties

Property name	Supported?	Comment
ServletPropertySet.PORT	✓	
GeneralPropertySet.HOSTNAME	✗	
GeneralPropertySet.LOGGING	✗	

Custom configuration properties:

Property name	Java constant to use	Valid values	Description	Example
cargo.orion.rmi.port	OrionPropertySet.RMI_PORT	Integer	Port for the Orion RMI server	"25791"

Orion 1.x

This page last changed on Dec 26, 2005 by vmassol.

Supported Features

Feature name	Java	Ant	Maven	Comment
Container Start	✓	✓	✓	
Container Stop	✓	✓	✓	
Container Instantiation	✓ `o.c.c.container.orion.Orion1xLocalContainer`	✓ `<cargo containerId="orion1x" />`	✓ `cargo.containers = orion1x`	
Container Factory	✓ `"orion1x"`	✓ `<cargo containerId="orion1x" />`	✓ `cargo.containers = orion1x`	
Container Classpath	✓	✓	✗	
Debugging	✓	✓	✓	
Passing system properties	✓	✓	✗	
Standalone Local Configuration	✓ `o.c.c.container.orion.OrionStandaloneLocalConfiguration`	✓	✓	
Existing local configuration	✗	✗	✗	
Static deployment of WAR	✓	✓	✓	
Static deployment of expanded WAR	✓	✓	✓	
Static deployment of EAR	✓	✓	✓	
Standalone mode	✓	✓	✓	
Embedded mode	✗	✗	✗	
Installer	✓	✓	✓	
Hot Deployer - deploy()	✗	✗	✗	
Hot Deployer - undeploy()	N/A	N/A	N/A	Can Orion 1.x support this?
Hot Deployer - start()	N/A	N/A	N/A	Can Orion 1.x support this?
Hot Deployer - stop()	N/A	N/A	N/A	Can Orion 1.x support this?

Supported Configuration properties

Property name	Supported?	Comment
ServletPropertySet.PORT	✓	
GeneralPropertySet.HOSTNAME	✗	
GeneralPropertySet.LOGGING	✗	

Custom configuration properties:

Property name	Java constant to use	Valid values	Description	Example
cargo.orion.rmi.port	OrionPropertySet.RMI_PORT	Integer	Port for the Orion RMI server	"25791"

Orion 2.x

This page last changed on Dec 26, 2005 by vmassol.

Supported Features

Feature name	Java	Ant	Maven	Comment
Container Start	✓	✓	✓	
Container Stop	✓	✓	✓	
Container Instantiation	✓ `o.c.c.container.orion.Orion2xContainer`	✓ `<cargo containerId="orion2x"/>`	✓ `cargo.containers = orion2x`	
Container Factory	✓ `"orion2x"`	✓ `<cargo containerId="orion2x"/>`	✓ `cargo.containers = orion2x`	
Container Classpath	✓	✓	✗	
Debugging	✓	✓	✓	
Passing system properties	✓	✓	✗	
Standalone Local Configuration	✓ `o.c.c.container.orion.OrionStandaloneLocalConfiguration`	✓	✓	
Existing local configuration	✗	✗	✗	
Static deployment of WAR	✓	✓	✓	
Static deployment of expanded WAR	✓	✓	✓	
Static deployment of EAR	✓	✓	✓	
Standalone mode	✓	✓	✓	
Embedded mode	✗	✗	✗	
Installer	✓	✓	✓	
Hot Deployer - deploy()	✗	✗	✗	
Hot Deployer - undeploy()	N/A	N/A	N/A	Can Orion 2.x support this?
Hot Deployer - start()	N/A	N/A	N/A	Can Orion 2.x support this?
Hot Deployer - stop()	N/A	N/A	N/A	Can Orion 2.x support this?

Supported Configuration properties

Property name	Supported?	Comment
ServletPropertySet.PORT	✓	
GeneralPropertySet.HOSTNAME	✗	
GeneralPropertySet.LOGGING	✗	

Custom configuration properties:

Property name	Java constant to use	Valid values	Description	Example
cargo.orion.rmi.port	OrionPropertySet.RMI_PORT	Integer	Port for the Orion RMI server	"25791"

Resin 2.x

This page last changed on Dec 26, 2005 by vmassol.

Supported Features

Feature name	Java	Ant	Maven	Comment
Container Start	✓	✓	✓	
Container Stop	✓	✓	✓	
Container Instantiation	✓ o.c.c.container.resin.Resin2xLocalContainer	✓ <cargo containerId="resin2x" />	✓ cargo.containers = resin2x	
Container Factory	✓ "resin2x"	✓ <cargo containerId="resin2x" />	✓ cargo.containers = resin2x	
Container Classpath	✓	✓	✗	
Debugging	✓	✓	✓	
Passing system properties	✓	✓	✗	
Standalone Local Configuration	✓ o.c.c.container.resin.Resin2xStandaloneLocalConfiguration	✓	✓	
Existing local configuration	✓ o.c.c.container.resin.ResinExistingLocalConfiguration	✓	✓	
Static deployment of WAR	✓	✓	✓	
Static deployment of expanded WAR	✓	✓	✓	
Static deployment of EAR	N/A	N/A	N/A	
Standalone mode	✓	✓	✓	
Embedded mode	✗	✗	✗	
Installer	✓	✓	✓	
Hot Deployer - deploy()	✓ o.c.c.container.resin.ResinDeployer	✗	✗	
Hot Deployer - undeploy()	N/A	N/A	N/A	Does Resin support this?
Hot Deployer - start()	N/A	N/A	N/A	Does Resin support this?
Hot Deployer - stop()	N/A	N/A	N/A	Does Resin support this?

Supported Configuration properties

Property name	Supported?	Comment
ServletPropertySet.PORT	✓	
GeneralPropertySet.HOSTNAME	✓	
GeneralPropertySet.LOGGING	✓	

Resin 3.x

This page last changed on Dec 26, 2005 by vmassol.

Supported Features

Feature name	Java	Ant	Maven	Comment
Container Start	✓	✓	✓	
Container Stop	✓	✓	✓	
Container Instantiation	✓ o.c.c.container.resin.Resin3xLocalContainer	✓ <cargo containerId="resin3x" />	✓ cargo.containers = resin3x	
Container Factory	✓ "resin3x"	✓ <cargo containerId="resin3x" />	✓ cargo.containers = resin3x	
Container Classpath	✓	✓	✗	
Debugging	✓	✓	✓	
Passing system properties	✓	✓	✗	
Standalone Local Configuration	✓ o.c.c.container.resin.Resin3xStandaloneLocalConfiguration	✓	✓	
Existing local configuration	✓ o.c.c.container.resin.ResinExistingLocalConfiguration	✓	✓	
Static deployment of WAR	✓	✓	✓	
Static deployment of expanded WAR	✓	✓	✓	
Static deployment of EAR	✗	✗	✗	Apparently Resin 3.x supports EAR
Standalone mode	✓	✓	✓	
Embedded mode	✗	✗	✗	
Installer	✓	✓	✓	
Hot Deployer - deploy()	✓ o.c.c.container.resin.ResinDeployer	✗	✗	Resin 3.0.12 and above have an issue. See RSN-186
Hot Deployer - undeploy()	N/A	N/A	N/A	Does Resin support this?
Hot Deployer - start()	N/A	N/A	N/A	Does Resin support this?
Hot Deployer -	N/A	N/A	N/A	Does Resin

stop()				support this?

Supported Configuration properties

Property name	Supported?	Comment
ServletPropertySet.PORT		
GeneralPropertySet.HOSTNAME		
GeneralPropertySet.LOGGING		

Tomcat 3.x

This page last changed on Dec 26, 2005 by vmassol.

Supported Features

Feature name	Java	Ant	Maven	Comment
Container Start	✓	✓	✓	
Container Stop	✓	✓	✓	
Container Instantiation	✓ `o.c.c.container.tomcat.Tomcat3xStandaloneContainer`	✓ `<cargo containerId="tomcat3x"/>`	✓ `cargo.containers = tomcat3x`	
Container Factory	✓ `"tomcat3x"`	✓ `<cargo containerId="tomcat3x"/>`	✓ `cargo.containers = tomcat3x`	
Container Classpath	✓	✓	✗	
Debugging	✓	✓	✓	
Passing system properties	✓	✓	✗	
Standalone Local Configuration	✓ `o.c.c.container.tomcat.TomcatStandaloneLocalConfiguration`	✓	✓	
Existing local configuration	✗	✗	✗	
Static deployment of WAR	✓	✓	✓	
Static deployment of expanded WAR	✓	✓	✓	
Static deployment of EAR	N/A	N/A	N/A	
Standalone mode	✓	✓	✓	
Embedded mode	✗	✗	✗	
Installer	✓	✓	✓	
Hot Deployer - deploy()	✗	✗	✗	
Hot Deployer - undeploy()	N/A	N/A	N/A	Can Tomcat 3.x support this?
Hot Deployer - start()	N/A	N/A	N/A	Can Orion 3.x support this?
Hot Deployer - stop()	N/A	N/A	N/A	Can Orion 3.x support this?

Supported Configuration properties

Property name	Supported?	Comment
ServletPropertySet.PORT		
GeneralPropertySet.HOSTNAME		
GeneralPropertySet.LOGGING		

Tomcat 4.x

This page last changed on Dec 26, 2005 by vmassol.

Supported Features

Feature name	Java	Ant	Maven	Comment
Container Start	✓	✓	✓	
Container Stop	✓	✓	✓	
Container Instantiation	✓ `o.c.c.container.tomcat.Tomcat4x`	✓ `<cargo containerId="tomcat4x">`	✓ `Cargo containers = tomcat4x`	
Container Factory	✓ `"tomcat4x"`	✓ `<cargo containerId="tomcat4x">`	✓ `Cargo containers = tomcat4x`	
Container Classpath	✓	✓	✗	
Debugging	✓	✓	✓	
Passing system properties	✓	✓	✗	
Standalone Local Configuration	✓ `o.c.c.container.tomcat.CatalinaStandaloneLocalConfiguration`	✓	✓	
Existing local configuration	✗	✗	✗	
Static deployment of WAR	✓	✓	✓	Does not support `META-INF/context.xml` files yet
Static deployment of expanded WAR	✓	✓	✓	
Static deployment of EAR	N/A	N/A	N/A	
Standalone mode	✓	✓	✓	
Embedded mode	✗	✗	✗	
Installer	✓	✓	✓	
Hot Deployer - deploy()	✗	✗	✗	
Hot Deployer - undeploy()	N/A	N/A	N/A	Can Tomcat 4.x support this?
Hot Deployer - start()	N/A	N/A	N/A	Can Orion 4.x support this?
Hot Deployer - stop()	N/A	N/A	N/A	Can Orion 4.x support this?

Supported Configuration properties

Property name	Supported?	Comment
ServletPropertySet.PORT	✓	
GeneralPropertySet.HOSTNAME	✓	
GeneralPropertySet.LOGGING	✓	

Custom configuration properties:

Property name	Java constant to use	Valid values	Description	Example
cargo.tomcat.shutdown.port	TomcatPropertySet.SHUTDOWN_PORT	Integer	TCP/IP port number on which this server waits for a shutdown command	"8205"

Tomcat 5.x

This page last changed on Dec 26, 2005 by vmassol.

Supported Features

Note: Tomcat 5.5.x is supported by requires JDK 1.5+

Feature name	Java	Ant	Maven	Comment
Container Start	✓	✓	✓	
Container Stop	✓	✓	✓	
Container Instantiation	✓ `o.c.c.container.tomcat.Tomcat5xContainer`	✓ `<cargo containerId="tomcat5x" />`	✓ `cargo.containers = tomcat5x`	
Container Factory	✓ `"tomcat5x"`	✓ `<cargo containerId="tomcat5x" />`	✓ `cargo.containers = tomcat5x`	
Container Classpath	✓	✓	✗	
Debugging	✓	✓	✓	
Passing system properties	✓	✓	✗	
Standalone Local Configuration	✓ `o.c.c.container.tomcat.CatalinaStandaloneLocalConfiguration`	✓	✓	
Existing local configuration	✗	✗	✗	
Static deployment of WAR	✓	✓	✓	
Static deployment of expanded WAR	✓	✓	✓	
Static deployment of EAR	N/A	N/A	N/A	
Standalone mode	✓	✓	✓	
Embedded mode	✗	✗	✗	
Installer	✓	✓	✓	
Hot Deployer - deploy()	✗	✗	✗	
Hot Deployer - undeploy()	✗	✗	✗	
Hot Deployer - start()	✗	✗	✗	
Hot Deployer -	✗	✗	✗	

stop()				

Supported Configuration properties

Property name	Supported?	Comment
ServletPropertySet.PORT	✓	
GeneralPropertySet.HOSTNAME	✓	
GeneralPropertySet.LOGGING	✓	

Custom configuration properties:

Property name	Java constant to use	Valid values	Description	Example
cargo.tomcat.shutdown.port	TomcatPropertySet.SHUTDOWN_PORT	Integer	TCP/IP port number on which this server waits for a shutdown command	"8205"

WebLogic 8.x

This page last changed on Dec 26, 2005 by vmassol.

Supported Features

Feature name	Java	Ant	Maven	Comment
Container Start	✓	✓	✓	
Container Stop	✓	✓	✓	
Container Instantiation	✓ `o.c.c.container.weblogic.WebLogic8xContainer`	✓ `<cargo containerId="weblogic8x" />`	✓ `containers = weblogic8x`	
Container Factory	✓ `"weblogic8x"`	✓ `<cargo containerId="weblogic8x" />`	✓ `containers = weblogic8x`	
Container Classpath	✓	✓	✗	
Debugging	✓	✓	✓	
Passing system properties	✓	✓	✗	
Standalone Local Configuration	✓ `o.c.c.container.weblogic.WebLogicStandaloneConfiguration`	✓	✓	
Existing local configuration	✓ `o.c.c.container.weblogic.WebLogicExistingConfiguration`	✓	✓	
Static deployment of WAR	✓	✓	✓	
Static deployment of expanded WAR	✗	✗	✗	
Static deployment of EAR	✓	✓	✓	
Standalone mode	✓	✓	✓	
Embedded mode	✗	✗	✗	
Installer	✓	✓	✓	
Hot Deployer - deploy()	✗	✗	✗	
Hot Deployer - undeploy()	✗	✗	✗	
Hot Deployer - start()	✗	✗	✗	
Hot Deployer - stop()	✗	✗	✗	

Supported Configuration properties

Property name	Supported?	Comment
ServletPropertySet.PORT	✓	
GeneralPropertySet.HOSTNAME	✗	
GeneralPropertySet.LOGGING	✗	
WebLogicPropertySet.ADMIN_USER	✓	WebLogic admin user name. Defaults to "weblogic"
WebLogicPropertySet.ADMIN_PWD	✓	WebLogic admin user password. Defaults to "weblogic"
WebLogicPropertySet.SERVER	✓	WebLogic server name. Defaults to "server"
WebLogicPropertySet.DOMAIN	✓	WebLogic domain name. Defaults to "domain"

Developers

This page last changed on Nov 22, 2005 by vmassol.

- Adding a container
- Building
- Contributing
- Discussions
- Project Structure
- Release procedure
- SVN

Adding a container

This page last changed on Dec 04, 2005 by vmassol.

Before you start you might be interested in reading the Project Structure tutorial which shows the directory organization of the Cargo sources. The Building tutorial explains how to build Cargo from sources and the Contributing tutorial explains what rules to follow when contributing code.

Here are some quick steps to follow if you wish to add support for a new container in Cargo:

- Subscribe to the cargo dev mailing list and ask as many question you'd like there!
- Create a JIRA issue on http://jira.codehaus.org (you'll need to register). I'll then add you to the cargo-developers group in JIRA and assign the issue to you
- Checkout Cargo from SVN trunk
- Understand the Cargo project's directory structure. Container implementations are located in `trunk/core/containers/`*`ContainerName`*.
- Have a look at existing container implementations (search for example for `Resin3xLocalContainer` or `Orion2xLocalContainer`).
- Create a `org.codehaus.cargo.container.`*`containerName`* package if it doesn't already exist.
- Create the following classes:
  - A container implementation class named `_ServerNameNxContainerType_Container` where `ServerName` is the name of the container, `N` the version and `ContainerType` the type of container (Local or Remote). For example: `JBoss3xLocalContainer`.
  - A configuration implementation class named `_ServerNameConfigurationType_Configuration` where `ConfigurationType` can be `StandaloneLocal` or `ExistingLocal`. For example `JBossStandaloneLocalConfiguration`.
  - You may need to implement some ancillary classes but those are the main 2 required. Check how the other container are implemented to see how to implement them and what other classes you may need to implement.
- Cargo has an SPI that you should use and that should make it easy for you. Your container class should extend `org.codehaus.cargo.container.spi.Abstract_ContainerType_Container` and your configuration class should extend `org.codehaus.cargo.container.spi.configuration.Abstract_ConfigurationType_Configuration`.
- Register your new classes in the generic API in the Factory classes `trunk/core/api/generic` so that users can use your new container by using the generic API.
- Add your container to the tests in `trunk/samples/java`. This means editing the `*Test.java` classes and adding your container in the `suite()` method.
- Run the Cargo build to ensure everything is working. You'll probably find that you haven't followed the Cargo project's coding conventions... Fix those and build again until it passes!
- Register on Codehaus' confluence. Once this is done I'll add you to the cargo-developers user group so that you have the right to edit yourself the Cargo web site pages
- Document the new container on the Cargo web site
- Create a SVN patch and attach it to the JIRA issue you have created above

Thanks and happy coding!

Building

This page last changed on Nov 26, 2005 by vmassol.

Prequisites

- Check out Cargo from SVN into a CARGOHOME directory (wherever you want on your machine)
- Install Maven 1 or Maven 2. Verify your installation works by typing "maven --version" (for Maven 1) or "mvn --version" (for Maven 2) at a command prompt.

Building with Maven 1

First time build

The Cargo build contains functional tests. Those tests are run on different containers. The first time you build Cargo it will download those container distributions which will take some time (the containers are installed into CARGOHOME/target/installs). If you want to tell Cargo to run only on some specific container, see below.

Maven 2 plugin cannot be built with Maven 1

The Maven 2 plugin located in CARGOHOME/extensions/maven2 cannot currently be built with Maven 1. The same applies for its functional tests located in CARGOHOME/samples/extensions/maven2.

- Go to CARGOHOME and type "maven". This will build the full Cargo project and the distribution jars will be generated in CARGOHOME/distribution/target/maven. The functional tests will be run on the default container set (see the section on "Selecting containers" below for more on that).
- If you want to build a single project, cd to that project and type "maven". Note that the build will fail if you've never built the dependent projects. Thuse it is recommended to build the full Cargo project at least once.
- If you wish to clean all build-generated files, cd to CARGOHOME and type "maven cargo:clean".

Selecting containers

The default list of containers to run on depends on the subproject being built:

- For the java samples, the list is in CARGOHOME/samples/java/project.properties. If you want to define a different list, simply create a build.properties file either in your home directory or in samples/java. In this file, create a cargo.containers listing the containers you wish to run on. For example if you only want to run on Tomcat 5.x you'd write:

```
cargo.containers = tomcat5x
```

- For the Ant samples, the list is in CARGOHOME/samples/extensions/ant/project.properties. Once again if you want to define a different list, simply create a build.properties file either in your

home directory or in `samples/extensions/ant`.
- For the Maven1 samples, the list is in `CARGOHOME/samples/extensions/maven/project.properties`.

Tips

- Type `maven -o` to work offline. This improves the build speed as Maven 1 does not check for updates on the remote repository for SNAPSHOTs.

Building with Maven 2

First time build

The Cargo build contains functional tests. Those tests are run on different containers. The first time you build Cargo it will download those container distributions which will take some time (the containers are installed into `\${java.io.tmpdir}/cargoinstalls`). If you want to tell Cargo to run only on some specific container, see below.

Maven 2 build not fully finished

The Cargo Maven 2 build is still not completely finished. Most subprojects have been converted to build with Maven 2 except for the following:

- samples/**
- extensions/intellijidea
- extensions/netbeans

- Go to `CARGOHOME` and type "`mvn install`". This will build the full Cargo project and the distribution jars will be generated in `CARGOHOME/distribution/target`.
- If you wish to clean all build-generated files, cd to `CARGOHOME` and type "`mvn clean`".

Contributing

This page last changed on Nov 26, 2005 by vmassol.

We're always looking for contributions! Here are some ways to participat in Cargo's development:

- by sending feedback to the user or dev mailing lists. The feedback could be about something that does not work, something that could be improve, a feature you'd like to see, etc. Or simply it could be that you're a happy user. Letting us know helps a lot!
- by answering emails from others on the mailing lists.
- by sending code patches. In that case there are a few rules you need to know.
- by spreading the word about Cargo!

Coding rules

If you submit a patch you need to follow these rules:

- copyright your code to Vincent Massol (see license explanations)
- ensure that your code passes the build. Note that the build contains some checkstyle checks that your code must pass.
- use the same code formatting as the existing code.
- create a JIRA issue and attach your patch to it.
- add your name on the Credits page.

In addition if you plan to contribute big pathes that impact existing code, we recommend discussing it on the mailing list first.

Thanks!

Discussions

This page last changed on Nov 22, 2005 by vmassol.

This page gathers ongoing design discussions (well, of course most of the discussions happen on the mailing list so you should check there too). Once we get a consensu and enough content the idea is to move the content of the discussions as proper documentation in the main stream of documentation.

Note: If you want to add a new discussion, simply create a child page of this one and it'll automatically appear in the list below.

Ongoing discussions

- Ant tasks — Discussion on the design of the Ant tasks.
- Comparisons with other tools — This is an attempt to compare Cargo with other tools.

Ant tasks

This page last changed on Oct 16, 2005 by aheritier.

Discussion on the design of the Ant tasks.

Here are the main discussions we had about the Ant tasks for Cargo:

- **Remove cargo-XXX tasks** : Actually there's one task for each container Id. It's not really evolutionary and works only with the containers referenced in the default Containers factory. We prefer to remove these aliases (CARGO-132) and to keep the generic one using either a container id as referenced in the default containers factory or the name of the container class.

- **Create a separate task for ZipUrlInstaller** : The zipUrlInstaller is actually a sub-element of the cargo task. We propose (CARGO-75) to extract it in a real task to be able to reuse it.

- **Create a separate task for deployers** : We can't reuse deployables between several servers and we can't have actually a remote deployer. We did a proposal (Thread) several months ago to solve this.

- **Extract configuration in an external (XML) file** : We discussed several times (Thread, CARGO-135) about to store the configuration in a file instead of in the build script.

Comparisons with other tools

This page last changed on Nov 21, 2005 by vmassol.

This is an attempt to compare Cargo with other tools.

Features	Cargo	Smartfrog	Geronimo
Configure containers/Generate configuration	★★	★★	★
Start/stop containers	★	★	★
Number of containers supported	★★	★	★
Distributed install of containers	✖	★★	?
Types of containers/nodes supported	★	★★★	★★

Notes on Cargo

- Its usage is currently focused on the development environment.
- Doesn't support distributed installations
- Supports remote deployments of J2EE archives
- Currently focused on J2EE containers but the architecture is general and can support any type of containers (database, IDEs, EAIs, etc).

Notes on Smartfrog

- Has a language to describe configurations
- Require agents to be installed
- Focused on container deployment, management and monitoring

Notes on Geronimo

- Completely wraps containers/components. Need adapter (GBean) to match component lifeycle with Geronimo's lifecycle.
- Focused on runtime and management of containers/components.

Project Structure

This page last changed on Dec 29, 2005 by vmassol.

Cargo's directory organization can be daunting for a newcomer. So here's some information on how the project is organized.

(view as slideshow)

Cargo Project Hierarchy

Legend

- **directory/** : represents a directory
- **directory/** : represents a directory containing a Maven project

- **cargo/**
  - **core/** : Core Java APIs
    - **api/**
      - **util/** : Some useful classes for both module/ and container/ (logging interface, etc)
      - **module/** : API to create/parse J2EE Module files
      - **container/** : API to start/stop/configure containers and deploy J2EE Modules in them
      - **generic/** : Generic and untyped Java API sitting on top of the Core Container Java API
    - **containers/**
      - **resin/** : Resin implementation
      - **tomcat/** : Tomcat implementation
      - **.../** : Other container implementations
    - **uberjar/** : Uberjar packaging all `api/` and containers/ jars in one big jar
  - **build-tools/** : Common files and tools related to the Maven 1 and Maven 2 builds (common checkstyle files, etc).
  - **extensions/** : Cargo extensions
    - **ant/** : Ant tasks that uses the core/ projects API
    - **maven/** : Maven 1 plugin (based on the Ant tasks)
    - **maven2/** : Maven 2 plugin
    - **netbeans/** : Netbeans plugin
    - **intellijidea/** : IntellijJ IDEA plugin
  - **samples/** : Sample projects that also serve as functional tests
    - **java/** : Samples exercising the Java core API
    - **extensions/**
      - **ant/** : Samples exercising the Ant tasks
      - **maven/** : Samples exercising the Maven 1 plugin
      - **maven2/** : Samples exercising the Maven 2 plugin
    - **testdata/** : Test data for all the Sample projects
      - **authentication-war** : Generates WAR with BASIC authentication defined in

web.xml

- **empty-ear** : Generates en empty EAR
- **empty-jar** : Generates en empty JAR
- **expanded-war** : Generates an expanded WAR
- **simple-ear** : Generates a simple EAR containing a simple WAR
- **simple-war** : Generates a simple WAR containing an HTML file
- **tomcat-context** : Generates a WAR containing a Tomcat context.xml file redefining the Context Root

Release procedure

This page last changed on Dec 30, 2005 by vmassol.

Releases are now performed using Maven 2 so you need to have Maven 2 installed (see the building page).

1. Create a branch in SVN so that others can keep working on the trunk. Create the branch in `svn+ssh://svn.cargo.codehaus.org/home/projects/cargo/scm/cargo/branches/<version>` and check it out on your local machine. Example:

```
svn copy svn+ssh://svn.cargo.codehaus.org/home/projects/cargo/scm/cargo/trunk \
  svn+ssh://svn.cargo.codehaus.org/home/projects/cargo/scm/cargo/branches/<N>
svn co svn+ssh://svn.cargo.codehaus.org/home/projects/cargo/scm/cargo/branches/<N>
```

2. On the SVN trunk change all references of `(N)-SNAPSHOT` by `(N+1)-SNAPSHOT` (for ex from `0.6-SNAPSHOT` to `0.7-SNAPSHOT`) and commit. Perform this by doing a global search and replace. Note: You may want to perform a clean before doing the search and replace to avoid changing all the build `target` directories.
3. In the new branch, do the same and replace all references of `(N)-SNAPSHOT` by `(N)` and commit (Ex: from `0.3-SNAPSHOT` to `0.3`).
4. Delete all the `(N)-SNAPSHOT` artifacts from your local maven repositories to have a clean slate.
5. Run `mvn install` at the top level of the release branch to build the different artifacts and ensure everything is ok
6. Run `mvn assembly:assembly` in `core/uberjar` of the release branch. Note: in the future this will be done automatically when you call `mvn install`.
7. Deploy everything by typing `mvn deploy` at the top level of the release branch. For this to work you'll need to create a `settings.xml` file (in your `.m2` directory or in your user home directory. Inside you'll need to define your credentials and more. Here's an example `settings.xml`:

```
<settings>
  [...]
  <servers>
    <server>
      <id>cargo-snapshot</id>
      <username>vmassol</username>
      <privateKey>/my/private/ssh/key</privateKey>
      <filePermissions>664</filePermissions>
      <directoryPermissions>775</directoryPermissions>
    </server>
    <server>
      <id>cargo-release</id>
      <username>vmassol</username>
      <privateKey>/my/private/ssh/key</privateKey>
      <filePermissions>664</filePermissions>
      <directoryPermissions>775</directoryPermissions>
    </server>
  </servers>
</settings>
```

8. Log onto Cargo JIRA, release the current version and add the next version
9. Check that the Cargo wiki is up to date. Specifically, perform the following updates:
    a. modify the status on the home page about the delivery
    b. modify the Downloads page to include the latest download links
    c. create a release notes page for the new version
    d. export the wiki to a zipped HTML file and add it the Downloads page
10. Tag the branch created in step 1 to `svn+ssh://svn.cargo.codehaus.org/home/projects/cargo/scm/cargo/tags/<version>`
11. Send an announcement email to Cargo mailing lists (and to other relevent sites)
12. Create a blog post

SVN

This page last changed on Dec 26, 2005 by vmassol.

For general information see the SVN page on Codehaus.

Repository browsing

Check out Fisheye at http://fisheye.codehaus.org/viewrep/cargo/cargo/trunk. Alternatively you can check the ViewCVS installation at http://svn.cargo.codehaus.org but it's not as nice as Fisheye...

Anonymous SVN Access

Using the 'svn' protocol:

```
svn co svn://svn.cargo.codehaus.org/cargo/scm/cargo/trunk
```

Using the 'https' protocol

```
https://svn.codehaus.org/cargo/cargo/trunk
```

Developer SVN Access

Using the 'svn+ssh' protocol:

```
svn co svn+ssh://svn.cargo.codehaus.org/home/projects/cargo/scm/cargo/trunk
```

Using the 'https' protocol

```
https://svn.codehaus.org/cargo/cargo/trunk
```

Community

This page last changed on Nov 22, 2005 by vmassol.

- Credits
- IRC
- License
- Mailing List Archives

Credits

This page last changed on Dec 01, 2005 by vmassol.

The following persons deserve credit for Cargo.

Committers:

- Vincent Massol  vmassol: Lead developer of Cargo
- Desire ATANGA: Initial implementation of Tomcat and WebLogic support
- Arnaud Heritier  arnaudheritier: Implementation of the Maven 1.x plugin
- Hendrik Schreiber: Implementation of jo! support and creation of the IDEA IntelliJ plugin
- Milos Kleint: Netbeans extensions
- I Nyoman Winardi: JBoss 3.x and 4.x support
- Lev Olkhovich: JSR-88 support and remote containers support
- Mark Hobson: Tomcat Hot deployment implementation + implementation of the Maven 2.x plugin
- Magnus Grimsell: Lots of improvements to the Deployment API.

Contributors:

- Jerome Lacoste: General ideas and discussions about Cargo
- Tim Shadel: Implementation of OC4J support
- Matt Raible: Asked for improvements to the Tomcat support so that Cargo can support nicely AppFuse. Provided patches to improve Tomcat support.
- Jan Zuchhold: Improvements to the Maven plugin (passing system properties)
- Eoghan McIlwaine: Cargo logo and banner. Eoghan has also done a larger, vertical version of the Cargo logo. Eoghan, you rock!
- Nigel Magnay: Several improvements to the Modules API and implementation of several container-specific merger classes.
- Bill Dudney: Implementation of the Tomcat Existing Local Configuration + beta tester of the m2 plugin

Special thanks

- Apache and The Jakarta cactus project: Cargo started as a refactoring of the Cactus Ant integration subproject
- Christopher Lenz: Has developed most of the Cactus Ant integration code that spawned the Cargo project

If we have forgotten anyone, please accept our apologies and feel free to edit the page yourself to correct it (ask me if you need the rights).

Artwork

- Eoghan's vertical logo for Cargo:


CARGO
CODEHAUS

IRC

This page last changed on Nov 26, 2005 by vmassol.

You can always pop in to the Cargo IRC where you can chat with Cargo developers and users.

If you have a IRC client use:

irc://irc.codehaus.org/#cargo

or if you don't you can use the http client interface

http://irc.codehaus.org

If you use the http client, please give yourself a nickname and set the channel to `#cargo`.

License

This page last changed on Nov 26, 2005 by vmassol.

Copyright

This product is copyrighted Vincent Massol (see below for details).

Portions of the code were copied from the Jakarta Cactus project in 2004 and thereafter modified. These portions of code are copyrighted The Apache Sofware Foundation. These portions of code were originally developed by Vincent Massol and Christopher Lenz on the Jakarta Cactus project.

Apache Software License

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or Object
      form, that is based on (or derived from) the Work and for which the
      editorial revisions, annotations, elaborations, or other modifications
      represent, as a whole, an original work of authorship. For the purposes
      of this License, Derivative Works shall not include works that remain
      separable from, or merely link (or bind by name) to the interfaces of,
      the Work and Derivative Works thereof.

      "Contribution" shall mean any work of authorship, including
      the original version of the Work and any modifications or additions
      to that Work or Derivative Works thereof, that is intentionally
      submitted to Licensor for inclusion in the Work by the copyright owner
      or by an individual or Legal Entity authorized to submit on behalf of
      the copyright owner. For the purposes of this definition, "submitted"
```

means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

```
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!)  The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

Copyright Massol! Why not copyright myself?

This portion of text has been copied from the Clirr's contributing page.

Licensing does matter and can get very confusing to maintain. Handing over copyrights to Massol keeps things simple for us to manage licensing issues. We do this for free and do not want to waste time on legal issues and managing who has copyright of which parts of the code. The only safe option is to have people hand over copyright to a person (like Massol) or an organisation (like Apache).

Some people feel that handing over copyright to Massol limits their rights in the code they contribute.

In fact it doesn't. Think about it: You give copyright to Massol, Massol immediately gives you the right to use the code under the ASL. What have you lost? You can still

- claim authorship on the Credits page and in JIRA issues. In addition your name will be added to SVN commits
- redistribute the code in source or binary form (provided the terms of the ASL are met).
- use the code in a commercial environment or link it into IDEs.
- fork the codebase if you are not happy with the way Massol is running the project.

Because the code is licensed under the ASL, the only thing you give up by assigning copyright to Massol is the right to veto a re-licensing of the code. For example Massol could re-license Cargo to another open source license without having to contact everybody who has ever contributed. Note that you would not lose any of the work you (and others) have done as it would still be licensed under ASL - noone can ever take these rights away from you!

All that said, the reason I decided to Copyright Cargo under my name is because I didn't know what other Copyright to use... Saying "The Cargo team" does not mean anything and does not solve the issue mentioned above. We can't license it to "Codehaus" either as Codehaus is not offering this service. Ideally we'd need a third party not-for-profit organization who could handle this for us (get signed CLAs, ensure code is clean, offer legal protection, etc). If aynone has any idea, please send it on the mailing list. I've always been uneasy about copyrighting this under my name and I'd love to find a better solution.

Mailing List Archives

This page last changed on Nov 22, 2005 by vmassol.

Archives on Nabble

- Dev list
- User list
- Announcements list
- All lists combined

Archives On GMANE

- User list
- Developer list

News Readers on GMANE

- User list News Reader
- Developer list News Reader

Archives on Codehaus

- User list
- Developer list
- Announcements list
- SCM list
- All lists

Misc

This page last changed on Nov 22, 2005 by vmassol.

- CenterHeaderFill
- LeftHeader
- Navigation
- QuickLinks
- RightHeader

CenterHeaderFill

This page last changed on Nov 22, 2005 by vmassol.

LeftHeader

This page last changed on Nov 22, 2005 by vmassol.

Navigation

This page last changed on Dec 30, 2005 by vmassol.

Cargo 0.7 doc

- Home
- Quick start
- What for?
- News
- Roadmap
- Features
- Installation
- Javadocs

Extensions

- Ant
- Maven 1
- Maven 2
- Netbeans
- IntelliJ IDEA

Downloads

- Downloads

Containers

- JBoss 3.x
- JBoss 4.x
- Jetty 4.x
- jo 1.x
- Oc4j 9.x
- Orion 1.x
- Orion 2.x
- Resin 2.x
- Resin 3.x
- Tomcat 3.x
- Tomcat 4.x
- Tomcat 5.x
- Weblogic 8.x

Support

- Issues
- Roadmap

- Change log
- Mailing List Archives

Community

- Mailing Lists
- News Reader
- IRC
- Discussions
- License
- Credits

Developers

- SVN
- Project Structure
- Contributing
- Building
- Wiki
- Release procedure
- CI status !http://ci.codehaus.org/beetlejuice/images/rss.gif!
- Adding a container

QuickLinks

This page last changed on Nov 22, 2005 by vmassol.

Download | Mailing Lists | IRC | SVN | Issue Tracker | Search

RightHeader

This page last changed on Nov 22, 2005 by vmassol.

Quick start

This page last changed on Nov 22, 2005 by vmassol.

The following examples demonstrate how to configure Resin 3.0.15 to start in `target/resin3x` and deploy a WAR located in `path/to/simple.war`. The default port is 8080. Please note that the `container.start()` and `container.stop()` methods wait until the container is fully started and fully stopped before continuing. Thus, for any action you are executing after, you are assured the container is completely operational.

Static deployment

Static deployment means that the Deployabe is deployed before the container is started. Here's an example using the strongly type Java API:

```
Deployable war = new WAR("path/to/simple.war");

Configuration configuration =
    new Resin3xStandaloneLocalConfiguration("target/myresin3x"));
configuration.addDeployable(war);

LocalContainer container = new Resin3xLocalContainer(configuration);
container.setHome("c:/apps/resin-3.0.15");

container.start();
// Here you are assured the container is started.

container.stop();
// Here you are assured the container is stopped.
```

Here's the same example using the generic untyped API:

```
Deployable war = new DefaultDeployableFactory().createDeployable(
    "resin3x", "path/to/simple.war", DeployableType.WAR);

LocalConfiguration configuration =
    (LocalConfigutation) new DefaultConfigurationFactory(
        "resin3x", ConfigurationType.STANDALONE);
configuration.addDeployable(war);

LocalContainer container =
    (LocalContainer) new DefaultContainerFactory().createContainer(
        "resin3x", configuration);
container.setHome("c:/apps/resin-3.0.15");

container.start();
// Here you are assured the container is started.

container.stop();
// Here you are assured the container is stopped.
```

Dynamic deployment

Dynamic deployment means that the Deployable is deployed after the container is started.

```
LocalContainer container = new Resin3xLocalContainer(
    new Resin3xStandaloneLocalConfiguration("target/myresin3x"));
```

```
container.setHome("c:/apps/resin-3.0.15");

container.start();

// Here you are assured the container is started.

Deployable war = new WAR("path/to/simple.war");
Deployer deployer = new ResinDeployer(container);
deployer.deploy(war)

// Here you are NOT sure the WAR has finished deploying. To be sure you
// need to use a DeployableMonitor to monitor the deployment. For example
// the following code deploys the WAR and wait until it is available to
// serve requests (the URL should point to a resource inside your WAR):
deployer.deploy(war, new URLDeployableMonitor("http://server:port/some/url"));

container.stop();

// Here you are assured the container is stopped.
```

Downloads

This page last changed on Dec 30, 2005 by vmassol.

> **Info**
>
> Older downloads can be found in the Archived Downloads section.

The downloads you need to pick depends on how you plan to use Cargo:

- If you want to integrate Cargo in your Java code you'll need to add the following jars to your classpath:
  - the `cargo-core-api-*` jars and the container implementations you wish to use (`cargo-core-container-*` jars).
  - **or** the `cargo-core-uberjar` jar which aggregates all the required jars.
- If you want to use Cargo from Ant, you'll need to pick the same jars as above in addition to the `cargo-ant` jar.
- If you want to use Cargo from Maven 1, you'll only need to install the `cargo-maven-plugin` jar in your local Maven installation.
- If you want to use Cargo from Maven 2, you don't need to install anything at all as Maven 2 will automatically download the required jars when you first use the plugin.
- If you want to use Cargo from IntelliJ IDEA, you only need to pick the `cargo-intellijidea` zip as it contains everything required.

Category	Version	Artifacts	Description	Release notes
Core API	0.7	cargo-core-api-util	Utility classes used by other core API jars	Notes
	0.7	cargo-core-api-module	API to parse/create J2EE Modules (WAR, EAR, etc)	Notes
	0.7	cargo-core-api-container	The main Cargo Container API and all associated object	Notes
	0.7	cargo-core-api-generic	Generic API wrapping the main Cargo Container API	Notes
Core Containers	0.7	cargo-core-container-jboss	JBoss implementation of the Core Container API	Notes
	0.7	cargo-core-container-jetty	Jetty implementation of the Core Container API	Notes
	0.7	cargo-core-container-jo	Jo implementation	Notes

			of the Core Container API	
	0.7	cargo-core-container-orion	Orion implementation of the Core Container API	Notes
	0.7	cargo-core-container-resin	Resin implementatio of the Core Container API	Notes
	0.7	cargo-core-container-tomcat	Tomcat implementation of the Core Container API	Notes
	0.7	cargo-core-container-weblogic	Weblogic implementation of the Core Container API	Notes
Core Uberjar	0.7	cargo-core-uberjar	Convenience jar containing all the other jars from above	Notes
Extensions	0.7	cargo-ant	Ant tasks for Cargo.	Notes
	0.7	cargo-maven-plugin	Maven 1 plugin for Cargo	Notes
	N/A	N/A	Maven 2 plugin for Cargo. Not released yet. See the Maven 2 plugin documentation page for how to use a development version of the Maven 2 Cargo plugin.	
	0.1	cargo-intellijidea	IntelliJ IDEA plugin for Cargo	Notes
	N/A	N/A	Netbeans plugin for Cargo. Not released yet.	

Documentation

TODO

Continuous Builds

Cargo is using Codehaus's Bettlejuice to build Cargo whenever there's a commit. This allows us to ensure that the Cargo build works at all times. As a side effect, this also means that you can grab the latest Cargo artifacts from Beetlejuice. Unfortunately Beetlejuice does not provide a URL to point directly to the artifacts so you'll need to perform the following actions:

- Go to the cargo project
- Click the item in the "last build" column
- Click on "the build has N Artifacts". You'll be presented with a list of artifacts built during the last build. Grab the one you need.

Archived Downloads

This page last changed on Dec 30, 2005 by vmassol.

Archived downloads

Version	Links	Release notes
0.1	Jar download|Documentation	Released on 11/09/04
0.2	Jar download|Documentation	Released on 03/10/04
0.3	Jar download|Documentation	Released on 30/10/04
0.4	Jar download|Documentation	Released on 26/11/04
0.5	Jar download|Documentation	Released on 30/04/05
0.6	Jar download|Documentation	Released on 21/07/05

Release notes for Cargo 0.5

This page last changed on Dec 30, 2005 by vmassol.

Implemented issues

jira.codehaus.org (25 issues)					
T	Key	Res	Summary	Assignee	Reporter
	CARGO-134	FIXED	Add public Container.getExtraClasspath() API	Vincent Massol	Vincent Massol
	CARGO-133	FIXED	Add public Container.getSystemProperties() API	Vincent Massol	Vincent Massol
	CARGO-131	FIXED	Test on Resin 3.0.9	Vincent Massol	Vincent Massol
	CARGO-130	FIXED	Test on Resin 2.1.16	Vincent Massol	Vincent Massol
	CARGO-123	FIXED	Move all Monitor-related classes to a o.c.c.util.monitor package	Vincent Massol	Vincent Massol
	CARGO-122	FIXED	Split the core subproject into 3: container, util and module	Vincent Massol	Vincent Massol
	CARGO-121	FIXED	Extract superclass for Resin and jo! deployer	Hendrik Schreiber	Hendrik Schreiber
	CARGO-119	FIXED	Functionality for adding ejb-refs to web applications	Vincent Massol	Magnus Grimsell
	CARGO-115	FIXED	Create a Jetty4xEmbedded Deployer	Vincent Massol	Vincent Massol
	CARGO-114	FIXED	Add Container.isAppend() API	Vincent Massol	Vincent Massol
	CARGO-113	FIXED	Add Container.getOutput() API	Vincent Massol	Vincent Massol
	CARGO-112	FIXED	Create URLDeployableMonitor to monitor deployable using a URL	Vincent Massol	Vincent Massol
	CARGO-111	FIXED	Add DeployableMonitor/DeployableMonitorListener	Vincent Massol	Vincent Massol

			interfaces to monitor deployable hot deployments		
	CARGO-110	FIXED	Functionality for adding web modules to application.xml	Vincent Massol	Magnus Grimsell
	CARGO-109	FIXED	WebXmlMerger cannot merge the run-as element of a servlet	Vincent Massol	Magnus Grimsell
	CARGO-108	FIXED	Ability to get Jetty Server object for configuration	Vincent Massol	Vincent Massol
	CARGO-105	FIXED	Move all configuration-related API to a new org.codehaus.cargo.container.configuration package	Vincent Massol	Vincent Massol
	CARGO-104	FIXED	Move static deployments to Standalone Configuration	Vincent Massol	Vincent Massol
	CARGO-103	FIXED	Create a Resin Deployer	Vincent Massol	Vincent Massol
	CARGO-102	FIXED	Add migration guide for migrating from Cargo 0.4 to Cargo 0.5	Vincent Massol	Vincent Massol
	CARGO-101	FIXED	Add support for jo!	Hendrik Schreiber	Hendrik Schreiber
	CARGO-98	FIXED	[Maven plugin] Container stops with maven	Arnaud Heritier	Arnaud Heritier
	CARGO-97	CANNOT REPRODUCE	Missing tools.jar in CP for WebLogic 8.x implementation	Vincent Massol	Vincent Massol
	CARGO-96	FIXED	Proxy settings do not seem to work when using the ZipURLInstaller	Arnaud Heritier	Vincent Massol
	CARGO-92	FIXED	Missing Xerces dependency when using the Maven Eclipse plugin	Vincent Massol	Arnaud Heritier

Source code changes

Changes detected by Clirr between `cargo-0.4.jar` and `cargo-0.5.jar`:

- [Core] `org.codehaus.cargo.deployment.*` package moved to `org.codehaus.cargo.module`

```
ERROR: 8001: org.codehaus.cargo.deployment.DefaultJarArchive: Class
org.codehaus.cargo.deployment.DefaultJarArchive removed
ERROR: 8001: org.codehaus.cargo.deployment.JarArchive: Class
org.codehaus.cargo.deployment.JarArchive removed
ERROR: 8001: org.codehaus.cargo.deployment.application.ApplicationXml: Class
org.codehaus.cargo.deployment.application.ApplicationXml removed
ERROR: 8001: org.codehaus.cargo.deployment.application.ApplicationXmlIo: Class
org.codehaus.cargo.deployment.application.ApplicationXmlIo removed
ERROR: 8001:
org.codehaus.cargo.deployment.application.ApplicationXmlIo$ApplicationXmlEntityResolver: Class
org.codehaus.cargo.deployment.application.ApplicationXmlIo$ApplicationXmlEntityResolver removed
ERROR: 8001: org.codehaus.cargo.deployment.application.ApplicationXmlTag: Class
org.codehaus.cargo.deployment.application.ApplicationXmlTag removed
ERROR: 8001: org.codehaus.cargo.deployment.application.ApplicationXmlVersion: Class
org.codehaus.cargo.deployment.application.ApplicationXmlVersion removed
ERROR: 8001: org.codehaus.cargo.deployment.application.DefaultApplicationXml: Class
org.codehaus.cargo.deployment.application.DefaultApplicationXml removed
ERROR: 8001: org.codehaus.cargo.deployment.application.DefaultEarArchive: Class
org.codehaus.cargo.deployment.application.DefaultEarArchive removed
ERROR: 8001: org.codehaus.cargo.deployment.application.EarArchive: Class
org.codehaus.cargo.deployment.application.EarArchive removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.AbstractDescriptor: Class
org.codehaus.cargo.deployment.webapp.AbstractDescriptor removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.AbstractDescriptorIo: Class
org.codehaus.cargo.deployment.webapp.AbstractDescriptorIo removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.AbstractDescriptorTag: Class
org.codehaus.cargo.deployment.webapp.AbstractDescriptorTag removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.DefaultWarArchive: Class
org.codehaus.cargo.deployment.webapp.DefaultWarArchive removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.WarArchive: Class
org.codehaus.cargo.deployment.webapp.WarArchive removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.WebXml: Class
org.codehaus.cargo.deployment.webapp.WebXml removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.WebXmlIo: Class
org.codehaus.cargo.deployment.webapp.WebXmlIo removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.WebXmlIo$WebXmlEntityResolver: Class
org.codehaus.cargo.deployment.webapp.WebXmlIo$WebXmlEntityResolver removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.WebXmlMerger: Class
org.codehaus.cargo.deployment.webapp.WebXmlMerger removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.WebXmlTag: Class
org.codehaus.cargo.deployment.webapp.WebXmlTag removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.WebXmlVersion: Class
org.codehaus.cargo.deployment.webapp.WebXmlVersion removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.jboss.JBossWarArchive: Class
org.codehaus.cargo.deployment.webapp.jboss.JBossWarArchive removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.jboss.JBossWebXml: Class
org.codehaus.cargo.deployment.webapp.jboss.JBossWebXml removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.jboss.JBossWebXmlIo: Class
org.codehaus.cargo.deployment.webapp.jboss.JBossWebXmlIo removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.jboss.JBossWebXmlTag: Class
org.codehaus.cargo.deployment.webapp.jboss.JBossWebXmlTag removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.tomcat.TomcatContextXml: Class
org.codehaus.cargo.deployment.webapp.tomcat.TomcatContextXml removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.tomcat.TomcatContextXmlIo: Class
org.codehaus.cargo.deployment.webapp.tomcat.TomcatContextXmlIo removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.tomcat.TomcatContextXmlTag: Class
org.codehaus.cargo.deployment.webapp.tomcat.TomcatContextXmlTag removed
ERROR: 8001: org.codehaus.cargo.deployment.webapp.tomcat.TomcatWarArchive: Class
org.codehaus.cargo.deployment.webapp.tomcat.TomcatWarArchive removed
INFO: 8000: org.codehaus.cargo.module.DefaultJarArchive: Class
org.codehaus.cargo.module.DefaultJarArchive added
INFO: 8000: org.codehaus.cargo.module.Descriptor: Class org.codehaus.cargo.module.Descriptor
added
INFO: 8000: org.codehaus.cargo.module.Dtd: Class org.codehaus.cargo.module.Dtd added
INFO: 8000: org.codehaus.cargo.module.Dtd$DtdHandler: Class
org.codehaus.cargo.module.Dtd$DtdHandler added
INFO: 8000: org.codehaus.cargo.module.Dtd$XmlEntityResolver: Class
org.codehaus.cargo.module.Dtd$XmlEntityResolver added
INFO: 8000: org.codehaus.cargo.module.DtdParseException: Class
org.codehaus.cargo.module.DtdParseException added
INFO: 8000: org.codehaus.cargo.module.Grammar: Class org.codehaus.cargo.module.Grammar added
INFO: 8000: org.codehaus.cargo.module.JarArchive: Class org.codehaus.cargo.module.JarArchive
```

```
added
INFO: 8000: org.codehaus.cargo.module.application.ApplicationXml: Class
org.codehaus.cargo.module.application.ApplicationXml added
INFO: 8000: org.codehaus.cargo.module.application.ApplicationXmlIo: Class
org.codehaus.cargo.module.application.ApplicationXmlIo added
INFO: 8000:
org.codehaus.cargo.module.application.ApplicationXmlIo$ApplicationXmlEntityResolver: Class
org.codehaus.cargo.module.application.ApplicationXmlIo$ApplicationXmlEntityResolver added
INFO: 8000: org.codehaus.cargo.module.application.ApplicationXmlTag: Class
org.codehaus.cargo.module.application.ApplicationXmlTag added
INFO: 8000: org.codehaus.cargo.module.application.ApplicationXmlVersion: Class
org.codehaus.cargo.module.application.ApplicationXmlVersion added
INFO: 8000: org.codehaus.cargo.module.application.DefaultApplicationXml: Class
org.codehaus.cargo.module.application.DefaultApplicationXml added
INFO: 8000: org.codehaus.cargo.module.application.DefaultEarArchive: Class
org.codehaus.cargo.module.application.DefaultEarArchive added
INFO: 8000: org.codehaus.cargo.module.application.EarArchive: Class
org.codehaus.cargo.module.application.EarArchive added
INFO: 8000: org.codehaus.cargo.module.ejb.DefaultEjbArchive: Class
org.codehaus.cargo.module.ejb.DefaultEjbArchive added
INFO: 8000: org.codehaus.cargo.module.ejb.EjbArchive: Class
org.codehaus.cargo.module.ejb.EjbArchive added
INFO: 8000: org.codehaus.cargo.module.ejb.EjbDef: Class org.codehaus.cargo.module.ejb.EjbDef
added
INFO: 8000: org.codehaus.cargo.module.ejb.EjbJarXml: Class
org.codehaus.cargo.module.ejb.EjbJarXml added
INFO: 8000: org.codehaus.cargo.module.ejb.EjbJarXmlIo: Class
org.codehaus.cargo.module.ejb.EjbJarXmlIo added
INFO: 8000: org.codehaus.cargo.module.ejb.EjbJarXmlIo$EjbJarXmlEntityResolver: Class
org.codehaus.cargo.module.ejb.EjbJarXmlIo$EjbJarXmlEntityResolver added
INFO: 8000: org.codehaus.cargo.module.ejb.EjbJarXmlTag: Class
org.codehaus.cargo.module.ejb.EjbJarXmlTag added
INFO: 8000: org.codehaus.cargo.module.ejb.EjbJarXmlVersion: Class
org.codehaus.cargo.module.ejb.EjbJarXmlVersion added
INFO: 8000: org.codehaus.cargo.module.ejb.Entity: Class org.codehaus.cargo.module.ejb.Entity
added
INFO: 8000: org.codehaus.cargo.module.ejb.Session: Class org.codehaus.cargo.module.ejb.Session
added
INFO: 8000: org.codehaus.cargo.module.ejb.VendorEjbDescriptor: Class
org.codehaus.cargo.module.ejb.VendorEjbDescriptor added
INFO: 8000: org.codehaus.cargo.module.ejb.orion.OrionEjbJarXml: Class
org.codehaus.cargo.module.ejb.orion.OrionEjbJarXml added
INFO: 8000: org.codehaus.cargo.module.ejb.orion.OrionEjbJarXmlIo: Class
org.codehaus.cargo.module.ejb.orion.OrionEjbJarXmlIo added
INFO: 8000: org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXml: Class
org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXml added
INFO: 8000: org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXmlIo: Class
org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXmlIo added
INFO: 8000: org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXmlTag: Class
org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXmlTag added
INFO: 8000: org.codehaus.cargo.module.ejb.websphere.IbmEjbJarBndXmi: Class
org.codehaus.cargo.module.ejb.websphere.IbmEjbJarBndXmi added
INFO: 8000: org.codehaus.cargo.module.ejb.websphere.IbmEjbJarBndXmiGrammar: Class
org.codehaus.cargo.module.ejb.websphere.IbmEjbJarBndXmiGrammar added
INFO: 8000: org.codehaus.cargo.module.ejb.websphere.IbmEjbJarBndXmiIo: Class
org.codehaus.cargo.module.ejb.websphere.IbmEjbJarBndXmiIo added
INFO: 8000: org.codehaus.cargo.module.webapp.AbstractDescriptor: Class
org.codehaus.cargo.module.webapp.AbstractDescriptor added
INFO: 8000: org.codehaus.cargo.module.webapp.AbstractDescriptorIo: Class
org.codehaus.cargo.module.webapp.AbstractDescriptorIo added
INFO: 8000: org.codehaus.cargo.module.webapp.AbstractDescriptorTag: Class
org.codehaus.cargo.module.webapp.AbstractDescriptorTag added
INFO: 8000: org.codehaus.cargo.module.webapp.DefaultWarArchive: Class
org.codehaus.cargo.module.webapp.DefaultWarArchive added
INFO: 8000: org.codehaus.cargo.module.webapp.VendorWebAppDescriptor: Class
org.codehaus.cargo.module.webapp.VendorWebAppDescriptor added
INFO: 8000: org.codehaus.cargo.module.webapp.WarArchive: Class
org.codehaus.cargo.module.webapp.WarArchive added
INFO: 8000: org.codehaus.cargo.module.webapp.WebXml: Class
org.codehaus.cargo.module.webapp.WebXml added
INFO: 8000: org.codehaus.cargo.module.webapp.WebXmlIo: Class
org.codehaus.cargo.module.webapp.WebXmlIo added
INFO: 8000: org.codehaus.cargo.module.webapp.WebXmlIo$WebXmlEntityResolver: Class
org.codehaus.cargo.module.webapp.WebXmlIo$WebXmlEntityResolver added
INFO: 8000: org.codehaus.cargo.module.webapp.WebXmlMerger: Class
org.codehaus.cargo.module.webapp.WebXmlMerger added
INFO: 8000: org.codehaus.cargo.module.webapp.WebXmlTag: Class
org.codehaus.cargo.module.webapp.WebXmlTag added
```

```
INFO: 8000: org.codehaus.cargo.module.webapp.WebXmlVersion: Class
org.codehaus.cargo.module.webapp.WebXmlVersion added
INFO: 8000: org.codehaus.cargo.module.webapp.jboss.JBossWarArchive: Class
org.codehaus.cargo.module.webapp.jboss.JBossWarArchive added
INFO: 8000: org.codehaus.cargo.module.webapp.jboss.JBossWebXml: Class
org.codehaus.cargo.module.webapp.jboss.JBossWebXml added
INFO: 8000: org.codehaus.cargo.module.webapp.jboss.JBossWebXmlIo: Class
org.codehaus.cargo.module.webapp.jboss.JBossWebXmlIo added
INFO: 8000: org.codehaus.cargo.module.webapp.jboss.JBossWebXmlTag: Class
org.codehaus.cargo.module.webapp.jboss.JBossWebXmlTag added
INFO: 8000: org.codehaus.cargo.module.webapp.orion.OrionWebXml: Class
org.codehaus.cargo.module.webapp.orion.OrionWebXml added
INFO: 8000: org.codehaus.cargo.module.webapp.orion.OrionWebXmlIo: Class
org.codehaus.cargo.module.webapp.orion.OrionWebXmlIo added
INFO: 8000: org.codehaus.cargo.module.webapp.tomcat.TomcatContextXml: Class
org.codehaus.cargo.module.webapp.tomcat.TomcatContextXml added
INFO: 8000: org.codehaus.cargo.module.webapp.tomcat.TomcatContextXmlIo: Class
org.codehaus.cargo.module.webapp.tomcat.TomcatContextXmlIo added
INFO: 8000: org.codehaus.cargo.module.webapp.tomcat.TomcatContextXmlTag: Class
org.codehaus.cargo.module.webapp.tomcat.TomcatContextXmlTag added
INFO: 8000: org.codehaus.cargo.module.webapp.tomcat.TomcatWarArchive: Class
org.codehaus.cargo.module.webapp.tomcat.TomcatWarArchive added
INFO: 8000: org.codehaus.cargo.module.webapp.weblogic.WeblogicXml: Class
org.codehaus.cargo.module.webapp.weblogic.WeblogicXml added
INFO: 8000: org.codehaus.cargo.module.webapp.weblogic.WeblogicXmlIo: Class
org.codehaus.cargo.module.webapp.weblogic.WeblogicXmlIo added
INFO: 8000: org.codehaus.cargo.module.webapp.weblogic.WeblogicXmlTag: Class
org.codehaus.cargo.module.webapp.weblogic.WeblogicXmlTag added
INFO: 8000: org.codehaus.cargo.module.webapp.websphere.IbmWebBndXmi: Class
org.codehaus.cargo.module.webapp.websphere.IbmWebBndXmi added
INFO: 8000: org.codehaus.cargo.module.webapp.websphere.IbmWebBndXmiGrammar: Class
org.codehaus.cargo.module.webapp.websphere.IbmWebBndXmiGrammar added
INFO: 8000: org.codehaus.cargo.module.webapp.websphere.IbmWebBndXmiIo: Class
org.codehaus.cargo.module.webapp.websphere.IbmWebBndXmiIo added
```

- [Core] Removed ability to add deployables to a Container. They must now either be added to a Configuration for static deployment or using a Deployer for dynamic deployments
- [Core] Added a `StandaloneConfiguration` interface which defines ths `addDeployable()` method.
- [Ant] Moved `<war>` and `<ear>` elements inside the `<configuration>` element

```
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public void
addDeployable(org.codehaus.cargo.container.deployable.Deployable)' has been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public java.util.List
getDeployables()' has been removed
ERROR: 7002: org.codehaus.cargo.ant.CargoTask: Method 'public void
addConfiguredEar(org.codehaus.cargo.ant.EARElement)' has been removed
ERROR: 7002: org.codehaus.cargo.ant.CargoTask: Method 'public void
addConfiguredWar(org.codehaus.cargo.ant.WARElement)' has been removed
ERROR: 7002: org.codehaus.cargo.ant.CargoTask: Method 'protected java.util.List getEars()' has
been removed
ERROR: 7002: org.codehaus.cargo.ant.CargoTask: Method 'protected java.util.List getWars()' has
been removed
ERROR: 7002: org.codehaus.cargo.ant.CargoTask: Method 'protected void
setupDeployables(org.codehaus.cargo.container.Container)' has been removed
INFO: 7011: org.codehaus.cargo.ant.ConfigurationElement: Method 'public void
addConfiguredEar(org.codehaus.cargo.ant.EARElement)' has been added
INFO: 7011: org.codehaus.cargo.ant.ConfigurationElement: Method 'public void
addConfiguredWar(org.codehaus.cargo.ant.WARElement)' has been added
INFO: 7011: org.codehaus.cargo.ant.ConfigurationElement: Method 'protected java.util.List
getEars()' has been added
INFO: 7011: org.codehaus.cargo.ant.ConfigurationElement: Method 'protected java.util.List
getWars()' has been added
INFO: 4000: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.StandaloneConfiguration to the set of implemented
interfaces
INFO: 4000: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.StandaloneConfiguration to the set of implemented
interfaces
INFO: 4000: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.StandaloneConfiguration to the set of implemented
interfaces
INFO: 4000: org.codehaus.cargo.container.resin.Resin2xStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.StandaloneConfiguration to the set of implemented
interfaces
INFO: 4000: org.codehaus.cargo.container.resin.Resin3xStandaloneConfiguration: Added
```

```
org.codehaus.cargo.container.configuration.StandaloneConfiguration to the set of implemented
interfaces
INFO: 4000: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.StandaloneConfiguration to the set of implemented
interfaces
INFO: 4000: org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.StandaloneConfiguration to the set of implemented
interfaces
INFO: 4000: org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.StandaloneConfiguration to the set of implemented
interfaces
INFO: 4000: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.StandaloneConfiguration to the set of implemented
interfaces
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public void
addDeployable(org.codehaus.cargo.container.deployable.Deployable)' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public java.util.List
getDeployables()' has been removed
INFO: 7011: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Method 'public
void addDeployable(org.codehaus.cargo.container.deployable.Deployable)' has been added
INFO: 7011: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Method 'public
java.util.List getDeployables()' has been added
```

- [Core] Added Deployer interface for dynamic deployments + implementation for Resin, Jetty and Jo1x

```
INFO: 8000: org.codehaus.cargo.container.deployer.DefaultDeployerFactory: Class
org.codehaus.cargo.container.deployer.DefaultDeployerFactory added
INFO: 8000: org.codehaus.cargo.container.deployer.DeployableMonitor: Class
org.codehaus.cargo.container.deployer.DeployableMonitor added
INFO: 8000: org.codehaus.cargo.container.deployer.DeployableMonitorListener: Class
org.codehaus.cargo.container.deployer.DeployableMonitorListener added
INFO: 8000: org.codehaus.cargo.container.deployer.Deployer: Class
org.codehaus.cargo.container.deployer.Deployer added
INFO: 8000: org.codehaus.cargo.container.deployer.DeployerFactory: Class
org.codehaus.cargo.container.deployer.DeployerFactory added
INFO: 8000: org.codehaus.cargo.container.deployer.URLDeployableMonitor: Class
org.codehaus.cargo.container.deployer.URLDeployableMonitor added
INFO: 8000: org.codehaus.cargo.container.jetty.JettyDeployer: Class
org.codehaus.cargo.container.jetty.JettyDeployer added
INFO: 8000: org.codehaus.cargo.container.orion.OrionDeployer: Class
org.codehaus.cargo.container.orion.OrionDeployer added
INFO: 8000: org.codehaus.cargo.container.resin.ResinDeployer: Class
org.codehaus.cargo.container.resin.ResinDeployer added
INFO: 8000: org.codehaus.cargo.container.jo.Jo1xDeployer: Class
org.codehaus.cargo.container.jo.Jo1xDeployer added
INFO: 8000: org.codehaus.cargo.container.spi.DeployerWatchdog: Class
org.codehaus.cargo.container.spi.DeployerWatchdog added
INFO: 8000: org.codehaus.cargo.container.spi.AbstractCopyingDeployer: Class
org.codehaus.cargo.container.spi.AbstractCopyingDeployer added
```

- [Core] Moved Configuration objects to package `org.codehaus.cargo.container.configuration` (they were previously in `org.codehaus.cargo.container`)
- [Core] Moved `org.codehaus.cargo.container.configuration.ConfigurationFactory` to an interface and added a `org.codehaus.cargo.container.configuration.DefaultConfigurationFactory`
- [Core] Added new `org.codehaus.cargo.container.configuration.ConfigurationCapability` class

```
ERROR: 7005: org.codehaus.cargo.container.Container: Parameter 1 of 'public void
setConfiguration(org.codehaus.cargo.container.Configuration)' has changed its type to
org.codehaus.cargo.container.configuration.Configuration
ERROR: 7006: org.codehaus.cargo.ant.ConfigurationElement: Return type of method 'public
org.codehaus.cargo.container.Configuration
createConfiguration(org.codehaus.cargo.container.Container)' has been changed to
org.codehaus.cargo.container.configuration.Configuration
ERROR: 8001: org.codehaus.cargo.container.Configuration: Class
org.codehaus.cargo.container.Configuration removed
ERROR: 8001: org.codehaus.cargo.container.ConfigurationFactory: Class
org.codehaus.cargo.container.ConfigurationFactory removed
ERROR: 8001: org.codehaus.cargo.container.ConfigurationFactory$ConfigurationKey: Class
```

```
org.codehaus.cargo.container.ConfigurationFactory$ConfigurationKey removed
INFO: 8000: org.codehaus.cargo.container.configuration.Configuration: Class
org.codehaus.cargo.container.configuration.Configuration added
INFO: 8000: org.codehaus.cargo.container.configuration.StandaloneConfiguration: Class
org.codehaus.cargo.container.configuration.StandaloneConfiguration added
INFO: 8000: org.codehaus.cargo.container.configuration.ConfigurationFactory: Class
org.codehaus.cargo.container.configuration.ConfigurationFactory added
INFO: 8000: org.codehaus.cargo.container.configuration.DefaultConfigurationFactory: Class
org.codehaus.cargo.container.configuration.DefaultConfigurationFactory added
INFO: 8000: org.codehaus.cargo.container.configuration.ConfigurationCapability: Class
org.codehaus.cargo.container.configuration.ConfigurationCapability added
ERROR: 7006: org.codehaus.cargo.container.spi.AbstractContainer: Return type of method 'public
org.codehaus.cargo.container.Configuration getConfiguration()' has been changed to
org.codehaus.cargo.container.configuration.Configuration
ERROR: 7006: org.codehaus.cargo.container.Container: Return type of method 'public
org.codehaus.cargo.container.Configuration getConfiguration()' has been changed to
org.codehaus.cargo.container.configuration.Configuration
ERROR: 7005: org.codehaus.cargo.container.spi.AbstractContainer: Parameter 1 of 'public void
setConfiguration(org.codehaus.cargo.container.Configuration)' has changed its type to
org.codehaus.cargo.container.configuration.Configuration
INFO: 7011: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Method 'public
org.codehaus.cargo.container.configuration.ConfigurationCapability getCapability()' has been
added
INFO: 7011: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Method 'public
org.codehaus.cargo.container.configuration.ConfigurationCapability getCapability()' has been
added
INFO: 7011: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: Method
'public org.codehaus.cargo.container.configuration.ConfigurationCapability getCapability()' has
been added
INFO: 8000: org.codehaus.cargo.container.spi.AbstractStandaloneConfigurationCapability: Class
org.codehaus.cargo.container.spi.AbstractStandaloneConfigurationCapability added
INFO: 7011: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: Method
'public org.codehaus.cargo.container.configuration.ConfigurationCapability getCapability()' has
been added
INFO: 4000: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.Configuration to the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.Configuration to the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.Configuration to the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.resin.Resin2xStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.Configuration to the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.resin.Resin3xStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.Configuration to the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.spi.AbstractConfiguration: Added
org.codehaus.cargo.container.configuration.Configuration to the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.Configuration to the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.spi.ContainerConfiguration: Added
org.codehaus.cargo.container.configuration.Configuration to the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.Configuration to the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.Configuration to the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: Added
org.codehaus.cargo.container.configuration.Configuration to the set of implemented interfaces
ERROR: 4001: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Removed
org.codehaus.cargo.container.Configuration from the set of implemented interfaces
ERROR: 4001: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Removed
org.codehaus.cargo.container.Configuration from the set of implemented interfaces
ERROR: 4001: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: Removed
org.codehaus.cargo.container.Configuration from the set of implemented interfaces
ERROR: 4001: org.codehaus.cargo.container.resin.Resin2xStandaloneConfiguration: Removed
org.codehaus.cargo.container.Configuration from the set of implemented interfaces
ERROR: 4001: org.codehaus.cargo.container.resin.Resin3xStandaloneConfiguration: Removed
org.codehaus.cargo.container.Configuration from the set of implemented interfaces
ERROR: 4001: org.codehaus.cargo.container.spi.AbstractConfiguration: Removed
org.codehaus.cargo.container.Configuration from the set of implemented interfaces
ERROR: 4001: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Removed
org.codehaus.cargo.container.Configuration from the set of implemented interfaces
ERROR: 4001: org.codehaus.cargo.container.spi.ContainerConfiguration: Removed
org.codehaus.cargo.container.Configuration from the set of implemented interfaces
ERROR: 4001: org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration: Removed
org.codehaus.cargo.container.Configuration from the set of implemented interfaces
ERROR: 4001: org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration: Removed
org.codehaus.cargo.container.Configuration from the set of implemented interfaces
ERROR: 4001: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: Removed
org.codehaus.cargo.container.Configuration from the set of implemented interfaces
```

```
INFO: 7003: org.codehaus.cargo.container.spi.AbstractConfiguration: Method 'public void
configure()' has been removed, but an inherited definition exists.
INFO: 7003: org.codehaus.cargo.container.spi.AbstractConfiguration: Method 'public void
verifyProperties()' has been removed, but an inherited definition exists.
INFO: 7003: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public java.lang.String
getId()' has been removed, but an inherited definition exists.
INFO: 7003: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public java.lang.String
getName()' has been removed, but an inherited definition exists.
```

- [Core] Added Jo 1.x support

```
INFO: 8000: org.codehaus.cargo.ant.jo.Jo1xCargoTask: Class
org.codehaus.cargo.ant.jo.Jo1xCargoTask added
INFO: 8000: org.codehaus.cargo.container.jo.Jo1xContainer: Class
org.codehaus.cargo.container.jo.Jo1xContainer added
INFO: 8000: org.codehaus.cargo.container.jo.Jo1xDeployer: Class
org.codehaus.cargo.container.jo.Jo1xDeployer added
INFO: 8000: org.codehaus.cargo.container.jo.Jo1xStandaloneConfiguration: Class
org.codehaus.cargo.container.jo.Jo1xStandaloneConfiguration added
INFO: 8000: org.codehaus.cargo.container.jo.JoPropertySet: Class
org.codehaus.cargo.container.jo.JoPropertySet added
```

- [Core] Renamed `org.codehaus.cargo.container.Capability` to `org.codehaus.cargo.container.ContainerCapability`

```
ERROR: 8001: org.codehaus.cargo.container.Capability: Class
org.codehaus.cargo.container.Capability removed
INFO: 8000: org.codehaus.cargo.container.ContainerCapability: Class
org.codehaus.cargo.container.ContainerCapability added
ERROR: 7006: org.codehaus.cargo.container.Container: Return type of method 'public
org.codehaus.cargo.container.Capability getCapability()' has been changed to
org.codehaus.cargo.container.ContainerCapability
ERROR: 7006: org.codehaus.cargo.container.jetty.Jetty4xEmbeddedContainer: Return type of method
'public org.codehaus.cargo.container.Capability getCapability()' has been changed to
org.codehaus.cargo.container.ContainerCapability
ERROR: 7006: org.codehaus.cargo.container.orion.AbstractOrionContainer: Return type of method
'public org.codehaus.cargo.container.Capability getCapability()' has been changed to
org.codehaus.cargo.container.ContainerCapability
ERROR: 7006: org.codehaus.cargo.container.resin.AbstractResinContainer: Return type of method
'public org.codehaus.cargo.container.Capability getCapability()' has been changed to
org.codehaus.cargo.container.ContainerCapability
ERROR: 7006: org.codehaus.cargo.container.tomcat.AbstractTomcatContainer: Return type of method
'public org.codehaus.cargo.container.Capability getCapability()' has been changed to
org.codehaus.cargo.container.ContainerCapability
ERROR: 7006: org.codehaus.cargo.container.weblogic.AbstractWebLogicContainer: Return type of
method 'public org.codehaus.cargo.container.Capability getCapability()' has been changed to
org.codehaus.cargo.container.ContainerCapability
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public
org.codehaus.cargo.container.Capability getCapability()' has been removed
```

- [Core] Moved all Monitor-related classes from `org.codehaus.cargo.util` to `org.codehaus.cargo.util.monitor`

```
ERROR: 4001: org.codehaus.cargo.container.Container: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.Container: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 4001: org.codehaus.cargo.container.deployable.Deployable: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.deployable.Deployable: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 4001: org.codehaus.cargo.container.deployable.EAR: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.deployable.EAR: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.deployable.EAR: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.deployable.EAR: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.deployable.WAR: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.deployable.WAR: Added
```

```
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.deployable.WAR: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.deployable.WAR: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.deployable.jboss.JBossWAR: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.deployable.jboss.JBossWAR: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.deployable.jboss.JBossWAR: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.deployable.jboss.JBossWAR: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.deployable.tomcat.TomcatWAR: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.deployable.tomcat.TomcatWAR: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.deployable.tomcat.TomcatWAR: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.deployable.tomcat.TomcatWAR: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.installer.Installer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.installer.Installer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.installer.Proxy: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
INFO: 5000: org.codehaus.cargo.container.installer.Proxy: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.installer.ZipURLInstaller: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.installer.ZipURLInstaller: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.installer.ZipURLInstaller: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.installer.ZipURLInstaller: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.jetty.Jetty4xEmbeddedContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.jetty.Jetty4xEmbeddedContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.jetty.Jetty4xEmbeddedContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.jetty.Jetty4xEmbeddedContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.orion.AbstractOrionContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.orion.AbstractOrionContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.orion.AbstractOrionContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.orion.AbstractOrionContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.orion.Oc4j9xContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.orion.Oc4j9xContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.orion.Oc4j9xContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.orion.Oc4j9xContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.orion.Orion1xContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.orion.Orion1xContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.orion.Orion1xContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.orion.Orion1xContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
```

```
ERROR: 4001: org.codehaus.cargo.container.orion.Orion2xContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.orion.Orion2xContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.orion.Orion2xContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.orion.Orion2xContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.resin.AbstractResinContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.resin.AbstractResinContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.resin.AbstractResinContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.resin.AbstractResinContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.resin.Resin2xContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.resin.Resin2xContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.resin.Resin2xContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.resin.Resin2xContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.resin.Resin2xStandaloneConfiguration: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.resin.Resin2xStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.resin.Resin2xStandaloneConfiguration: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.resin.Resin2xStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.resin.Resin3xContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.resin.Resin3xContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.resin.Resin3xContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.resin.Resin3xContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.resin.Resin3xStandaloneConfiguration: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.resin.Resin3xStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.resin.Resin3xStandaloneConfiguration: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.resin.Resin3xStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.spi.AbstractConfiguration: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.spi.AbstractConfiguration: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.spi.AbstractConfiguration: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.spi.AbstractConfiguration: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.spi.AbstractContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.spi.AbstractContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.spi.AbstractContainer: Removed
```

```
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.spi.AbstractContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.spi.ContainerConfiguration: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.spi.ContainerConfiguration: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 4001: org.codehaus.cargo.container.tomcat.AbstractCatalinaContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.tomcat.AbstractCatalinaContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.tomcat.AbstractCatalinaContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.tomcat.AbstractCatalinaContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.tomcat.AbstractTomcatContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.tomcat.AbstractTomcatContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.tomcat.AbstractTomcatContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.tomcat.AbstractTomcatContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.tomcat.Tomcat3xContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.tomcat.Tomcat3xContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.tomcat.Tomcat3xContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.tomcat.Tomcat3xContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.tomcat.Tomcat4xContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.tomcat.Tomcat4xContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.tomcat.Tomcat4xContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.tomcat.Tomcat4xContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.tomcat.Tomcat5xContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.tomcat.Tomcat5xContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.tomcat.Tomcat5xContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.tomcat.Tomcat5xContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.weblogic.AbstractWebLogicContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.weblogic.AbstractWebLogicContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.weblogic.AbstractWebLogicContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
```

```
INFO: 5000: org.codehaus.cargo.container.weblogic.AbstractWebLogicContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.weblogic.WebLogic8xContainer: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.weblogic.WebLogic8xContainer: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.weblogic.WebLogic8xContainer: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.weblogic.WebLogic8xContainer: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 4001: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: Removed
org.codehaus.cargo.util.Monitorable from the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
ERROR: 5001: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: Removed
org.codehaus.cargo.util.MonitoredObject from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 8001: org.codehaus.cargo.util.FileMonitor: Class org.codehaus.cargo.util.FileMonitor
removed
ERROR: 8001: org.codehaus.cargo.util.Monitor: Class org.codehaus.cargo.util.Monitor removed
ERROR: 8001: org.codehaus.cargo.util.Monitorable: Class org.codehaus.cargo.util.Monitorable
removed
ERROR: 8001: org.codehaus.cargo.util.MonitoredObject: Class
org.codehaus.cargo.util.MonitoredObject removed
ERROR: 8001: org.codehaus.cargo.util.NullMonitor: Class org.codehaus.cargo.util.NullMonitor
removed
ERROR: 8001: org.codehaus.cargo.util.SimpleMonitor: Class org.codehaus.cargo.util.SimpleMonitor
removed
INFO: 8000: org.codehaus.cargo.util.monitor.FileMonitor: Class
org.codehaus.cargo.util.monitor.FileMonitor added
INFO: 8000: org.codehaus.cargo.util.monitor.Monitor: Class
org.codehaus.cargo.util.monitor.Monitor added
INFO: 8000: org.codehaus.cargo.util.monitor.Monitorable: Class
org.codehaus.cargo.util.monitor.Monitorable added
INFO: 8000: org.codehaus.cargo.util.monitor.MonitoredObject: Class
org.codehaus.cargo.util.monitor.MonitoredObject added
INFO: 8000: org.codehaus.cargo.util.monitor.NullMonitor: Class
org.codehaus.cargo.util.monitor.NullMonitor added
INFO: 8000: org.codehaus.cargo.util.monitor.SimpleMonitor: Class
org.codehaus.cargo.util.monitor.SimpleMonitor added
```

- [Core] Added new `org.codehaus.cargo.util.monitor.AntMonitor` class

```
INFO: 8000: org.codehaus.cargo.util.monitor.AntMonitor: Class
org.codehaus.cargo.util.monitor.AntMonitor added
```

- [Core] Promoted `org.codehaus.cargo.container.internal.util.FileUtils` class to a public `org.codehaus.cargo.util.FileUtils` class as it is now used in the `org.codehaus.cargo.module` package (and we do want to have any dependency from `org.codehaus.cargo.module` to `org.codehaus.cargo.container` - only in the other direction)

```
INFO: 8000: org.codehaus.cargo.util.FileUtils: Class org.codehaus.cargo.util.FileUtils added
ERROR: 8001: org.codehaus.cargo.container.internal.util.FileUtils: Class
org.codehaus.cargo.container.internal.util.FileUtils removed
ERROR: 7006: org.codehaus.cargo.container.spi.AbstractConfiguration: Return type of method
'protected org.codehaus.cargo.container.internal.util.FileUtils getFileUtils()' has been
changed to org.codehaus.cargo.util.FileUtils
ERROR: 7006: org.codehaus.cargo.container.spi.AbstractContainer: Return type of method
'protected org.codehaus.cargo.container.internal.util.FileUtils getFileUtils()' has been
changed to org.codehaus.cargo.util.FileUtils
```

- [Core] Added new `org.codehaus.cargo.util.CargoException` which is the base of all Cargo exceptions.

```
INFO: 8000: org.codehaus.cargo.util.CargoException: Class
org.codehaus.cargo.util.CargoException added
WARNING: 5000: org.codehaus.cargo.container.ContainerException: Added
org.codehaus.cargo.util.CargoException to the list of superclasses
INFO: 7000: org.codehaus.cargo.container.ContainerException: Method 'public java.lang.Throwable
```

```
getOriginalThrowable()' is now implemented in superclass org.codehaus.cargo.util.CargoException
INFO: 7003: org.codehaus.cargo.container.ContainerException: Method 'public void
printStackTrace()' has been removed, but an inherited definition exists.
INFO: 7003: org.codehaus.cargo.container.ContainerException: Method 'public void
printStackTrace(java.io.PrintStream)' has been removed, but an inherited definition exists.
INFO: 7003: org.codehaus.cargo.container.ContainerException: Method 'public void
printStackTrace(java.io.PrintWriter)' has been removed, but an inherited definition exists.
```

- [Core] Added public APIs `org.codehaus.cargo.container.Container.getOutput()` and `org.codehaus.cargo.container.Container.isAppend()`

```
ERROR: 7012: org.codehaus.cargo.container.Container: Method 'public java.io.File getOutput()'
has been added to an interface
ERROR: 7012: org.codehaus.cargo.container.Container: Method 'public boolean isAppend()' has
been added to an interface
INFO: 7010: org.codehaus.cargo.container.spi.AbstractContainer: Accessibility of method
'protected java.io.File getOutput()' has been increased from protected to public
INFO: 7010: org.codehaus.cargo.container.spi.AbstractContainer: Accessibility of method
'protected boolean isAppend()' has been increased from protected to public
```

- [Core] Fixed proxy support in the Installer by adding a ProxyAuthenticator class
- [Core] Added tests for ZipURLInstaller and made small modifications to improve testability

```
INFO: 8000: org.codehaus.cargo.container.installer.Proxy$ProxyAuthenticator: Class
org.codehaus.cargo.container.installer.Proxy$ProxyAuthenticator added
INFO: 7010: org.codehaus.cargo.container.installer.ZipURLInstaller: Accessibility of method
'private void download()' has been increased from private to protected
INFO: 7011: org.codehaus.cargo.container.installer.ZipURLInstaller: Method 'protected void
setAntTaskFactory(org.codehaus.cargo.container.internal.util.AntTaskFactory)' has been added
```

- [Core] Added handy class when implementing Cargo factories based on a container id and a hint

```
INFO: 8000: org.codehaus.cargo.container.spi.AbstractGenericHintFactory: Class
org.codehaus.cargo.container.spi.AbstractGenericHintFactory added
INFO: 8000: org.codehaus.cargo.container.spi.AbstractGenericHintFactory$GenericParameters:
Class org.codehaus.cargo.container.spi.AbstractGenericHintFactory$GenericParameters added
```

- [Core] Start of a Resin ExistingConfiguration implementation. Does not work yet and must not be used.

```
INFO: 8000: org.codehaus.cargo.container.resin.ResinExistingConfiguration: Class
org.codehaus.cargo.container.resin.ResinExistingConfiguration added
```

- [Core] Added property for supporting container authentication

```
INFO: 6000: org.codehaus.cargo.container.property.ServletPropertySet: Added public field USERS
INFO: 8000: org.codehaus.cargo.container.property.User: Class
org.codehaus.cargo.container.property.User added
INFO: 8000: org.codehaus.cargo.container.tomcat.AbstractTomcatStandaloneConfiguration: Class
org.codehaus.cargo.container.tomcat.AbstractTomcatStandaloneConfiguration added
INFO: 5000: org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration: Added
org.codehaus.cargo.container.tomcat.AbstractTomcatStandaloneConfiguration to the list of
superclasses
INFO: 5000: org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration: Added
org.codehaus.cargo.container.tomcat.AbstractTomcatStandaloneConfiguration to the list of
superclasses
INFO: 7011: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Method 'protected
java.lang.String getRoleToken()' has been added
INFO: 7011: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Method 'protected
java.lang.String getUserToken()' has been added
INFO: 7011: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: Method
'protected java.lang.String getSecurityToken(java.lang.String, java.lang.String)' has been
added
```

- [Core] Added `org.codehaus.cargo.container.jetty.Jetty4xEmbeddedContainer.getServer()` API to allow users to further configure a Jetty server

```
INFO: 7011: org.codehaus.cargo.container.jetty.Jetty4xEmbeddedContainer: Method 'public
java.lang.Object getServer()' has been added
```

- [Core] Added possibility to pass JVM arguments to Configurations by introducing a new `cargo.jvmargs` property

```
INFO: 6000: org.codehaus.cargo.container.property.GeneralPropertySet: Added public field
JVMARGS
```

Release notes for Cargo 0.6

This page last changed on Dec 30, 2005 by vmassol.

Implemented issues

jira.codehaus.org (25 issues)					
T	Key	Res	Summary	Assignee	Reporter
	CARGO-168	FIXED	Distribution creation fails	Vincent Massol	Arnaud Heritier
	CARGO-167	FIXED	maven-findbugs-plugin not found when creating the distribution	Arnaud Heritier	Arnaud Heritier
	CARGO-166	FIXED	Add DeployableMonitor.undeploy() API to monitor undeployments	Vincent Massol	Vincent Massol
	CARGO-165	FIXED	Current distribution is not compatible with Maven2	Vincent Massol	Vincent Massol
	CARGO-163	FIXED	Add new ConfigurationFactory.isConfigurationRegistered() API method	Vincent Massol	Vincent Massol
	CARGO-162	FIXED	Move addDeployable/getDeployables from StandaloneConfiguration to Configuration	Vincent Massol	Vincent Massol
	CARGO-160	FIXED	Move all extensions to the core in a new extensions/ directory	Vincent Massol	Vincent Massol
	CARGO-159	FIXED	Fix SCM <connection> tags in build POMs which are using an invalid "http://" protocol	Vincent Massol	jeremi Joslin
	CARGO-158	FIXED	Transform ContainerFactory class into an interface	Vincent Massol	Vincent Massol
	CARGO-157	FIXED	Rename containerKey attribute in containerId in the Cargo task and add a new	Vincent Massol	Vincent Massol

			class attribute for custom container implementations		
	CARGO-156	FIXED	Completely separate DeployableFactory from Container interface	Vincent Massol	Vincent Massol
	CARGO-155	FIXED	Separate Generic Java API from the core container API	Vincent Massol	Vincent Massol
	CARGO-154	FIXED	Added new DeployableType class	Vincent Massol	Vincent Massol
	CARGO-153	FIXED	Remove dependency on Container in Configuration's constructors and Add Configuration to Container's constructors	Vincent Massol	Vincent Massol
	CARGO-152	FIXED	Add new Container.setDeployableFactory API	Vincent Massol	Vincent Massol
	CARGO-151	FIXED	Replace DeployableFactory.createWAR()/EAR() by a generic createDeployable()	Vincent Massol	Vincent Massol
	CARGO-150	FIXED	Introduce new ConfigurationType enumeration class	Vincent Massol	Vincent Massol
	CARGO-145	DUPLICATE	Add support for JBoss Application Server	Vincent Massol	I Nyoman Winardi
	CARGO-143	FIXED	Allow Tomcat META-INF/context.xml to be used in expanded webapps	Vincent Massol	Jan Zuchhold
	CARGO-142	CANNOT REPRODUCE	Ant tasks do not clear old temp files.	Vincent Massol	Michael Rimov
	CARGO-141	FIXED	EjbJarXml.getVendorDescriptor() returns null for WebSphere	Vincent Massol	Magnus Grimsell
	CARGO-140	FIXED	Maven cargo plugin references cargo-ant as a dependency	Vincent Massol	Brice Copy
	CARGO-139	FIXED	dtd:s are loaded	Vincent Massol	Magnus Grimsell

			from the web		
	CARGO-135	WON'T FIX	Add Cargo config file	Vincent Massol	Vincent Massol
	CARGO-127	FIXED	Add support for passing system properties in the Maven plugin	Vincent Massol	Vincent Massol

Source code changes

Changes detected by Clirr between `cargo-0.5.jar` and `cargo-0.6.jar`.

Binary compatibility breaks:

```
ERROR: org.codehaus.cargo.ant.CargoTask: In method 'protected void
executeActions(org.codehaus.cargo.container.Container)' the number of arguments has changed
ERROR: org.codehaus.cargo.ant.CargoTask: Method 'protected java.lang.String getContainerKey()'
has been removed
ERROR: org.codehaus.cargo.ant.CargoTask: Method 'public void setContainerKey(java.lang.String)'
has been removed
ERROR: org.codehaus.cargo.ant.CargoTask: Method 'protected void
setupConfiguration(org.codehaus.cargo.container.Container)' has been removed
ERROR: org.codehaus.cargo.ant.CargoTask: In method 'protected void
setupExtraClasspath(org.codehaus.cargo.container.Container)' the number of arguments has
changed
ERROR: org.codehaus.cargo.ant.CargoTask: In method 'protected void
setupHomeDir(org.codehaus.cargo.container.Container)' the number of arguments has changed
ERROR: org.codehaus.cargo.ant.CargoTask: In method 'protected void
setupMonitor(org.codehaus.cargo.container.Container)' the number of arguments has changed
ERROR: org.codehaus.cargo.ant.CargoTask: In method 'protected void
setupOutput(org.codehaus.cargo.container.Container)' the number of arguments has changed
ERROR: org.codehaus.cargo.ant.CargoTask: In method 'protected void
setupSystemProperties(org.codehaus.cargo.container.Container)' the number of arguments has
changed
ERROR: org.codehaus.cargo.ant.CargoTask: In method 'protected void
setupTimeout(org.codehaus.cargo.container.Container)' the number of arguments has changed
ERROR: org.codehaus.cargo.ant.ConfigurationElement: Parameter 1 of 'public
org.codehaus.cargo.container.configuration.Configuration
createConfiguration(org.codehaus.cargo.container.Container)' has changed its type to
java.lang.String
ERROR: org.codehaus.cargo.ant.ConfigurationElement: Method 'public java.lang.String getHint()'
has been removed
ERROR: org.codehaus.cargo.ant.ConfigurationElement: Method 'public void
setHint(java.lang.String)' has been removed
ERROR: org.codehaus.cargo.ant.EARElement: Parameter 1 of 'public
org.codehaus.cargo.container.deployable.EAR createEAR(org.codehaus.cargo.container.Container)'
has changed its type to java.lang.String
ERROR: org.codehaus.cargo.ant.WARElement: Parameter 1 of 'public
org.codehaus.cargo.container.deployable.WAR createWAR(org.codehaus.cargo.container.Container)'
has changed its type to java.lang.String
ERROR: org.codehaus.cargo.container.Container: Method 'public
org.codehaus.cargo.container.deployable.DeployableFactory getDeployableFactory()' has been
removed
ERROR: org.codehaus.cargo.container.ContainerFactory: Class
org.codehaus.cargo.container.ContainerFactory removed
ERROR: org.codehaus.cargo.container.configuration.Configuration: Method 'public void
addDeployable(org.codehaus.cargo.container.deployable.Deployable)' has been added to an
interface
ERROR: org.codehaus.cargo.container.configuration.Configuration: Method 'public java.util.List
getDeployables()' has been added to an interface
ERROR: org.codehaus.cargo.container.configuration.ConfigurationFactory: Class
org.codehaus.cargo.container.configuration.ConfigurationFactory removed
ERROR: org.codehaus.cargo.container.configuration.DefaultConfigurationFactory: Class
org.codehaus.cargo.container.configuration.DefaultConfigurationFactory removed
ERROR:
org.codehaus.cargo.container.configuration.DefaultConfigurationFactory$ConfigurationFactoryParameters:Class
org.codehaus.cargo.container.configuration.DefaultConfigurationFactory$ConfigurationFactoryParameters
removed
```

```
ERROR: org.codehaus.cargo.container.configuration.StandaloneConfiguration: Class
org.codehaus.cargo.container.configuration.StandaloneConfiguration removed
ERROR: org.codehaus.cargo.container.deployable.DeployableFactory: Class
org.codehaus.cargo.container.deployable.DeployableFactory removed
ERROR: org.codehaus.cargo.container.deployer.DefaultDeployerFactory: Class
org.codehaus.cargo.container.deployer.DefaultDeployerFactory removed
ERROR: org.codehaus.cargo.container.deployer.DeployableMonitorListener: Method 'public void
undeployed()' has been added to an interface
ERROR: org.codehaus.cargo.container.deployer.DeployerFactory: Class
org.codehaus.cargo.container.deployer.DeployerFactory removed
ERROR: org.codehaus.cargo.container.internal.util.HintKey: Class
org.codehaus.cargo.container.internal.util.HintKey removed
ERROR: org.codehaus.cargo.container.jetty.Jetty4xEmbeddedContainer: In method 'public
Jetty4xEmbeddedContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Removed
org.codehaus.cargo.container.configuration.StandaloneConfiguration from the set of implemented
interfaces
ERROR: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Parameter 1 of 'public
JettyStandaloneConfiguration(org.codehaus.cargo.container.Container)' has changed its type to
java.io.File
ERROR: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Method 'public
JettyStandaloneConfiguration(org.codehaus.cargo.container.Container, java.io.File)' has been
removed
ERROR: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: In method 'public void
configure()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.jo.Jo1xContainer: In method 'public Jo1xContainer()' the
number of arguments has changed
ERROR: org.codehaus.cargo.container.jo.Jo1xStandaloneConfiguration: Removed
org.codehaus.cargo.container.configuration.StandaloneConfiguration from the set of implemented
interfaces
ERROR: org.codehaus.cargo.container.jo.Jo1xStandaloneConfiguration: Parameter 1 of 'public
Jo1xStandaloneConfiguration(org.codehaus.cargo.container.Container)' has changed its type to
java.io.File
ERROR: org.codehaus.cargo.container.jo.Jo1xStandaloneConfiguration: Method 'public
Jo1xStandaloneConfiguration(org.codehaus.cargo.container.Container, java.io.File)' has been
removed
ERROR: org.codehaus.cargo.container.jo.Jo1xStandaloneConfiguration: In method 'public void
configure()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.orion.AbstractOrionContainer: In method 'public
AbstractOrionContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.orion.Oc4j9xContainer: In method 'public Oc4j9xContainer()'
the number of arguments has changed
ERROR: org.codehaus.cargo.container.orion.Orion1xContainer: In method 'public
Orion1xContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.orion.Orion2xContainer: In method 'public
Orion2xContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.orion.OrionDeployer: Class
org.codehaus.cargo.container.orion.OrionDeployer removed
ERROR: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Removed
org.codehaus.cargo.container.configuration.StandaloneConfiguration from the set of implemented
interfaces
ERROR: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Parameter 1 of 'public
OrionStandaloneConfiguration(org.codehaus.cargo.container.Container)' has changed its type to
java.io.File
ERROR: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Method 'public
OrionStandaloneConfiguration(org.codehaus.cargo.container.Container, java.io.File)' has been
removed
ERROR: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: In method 'public void
configure()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.resin.AbstractResinContainer: In method 'public
AbstractResinContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: Removed
org.codehaus.cargo.container.configuration.StandaloneConfiguration from the set of implemented
interfaces
ERROR: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: Parameter 1 of
'public AbstractResinStandaloneConfiguration(org.codehaus.cargo.container.Container)' has
changed its type to java.io.File
ERROR: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: Method 'public
AbstractResinStandaloneConfiguration(org.codehaus.cargo.container.Container, java.io.File)' has
been removed
ERROR: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: In method
'public void configure()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: In method
'protected void prepareAdditions(org.apache.tools.ant.types.FilterChain)' the number of
arguments has changed
ERROR: org.codehaus.cargo.container.resin.Resin2xContainer: In method 'public
Resin2xContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.resin.Resin2xStandaloneConfiguration: Removed
```

```
org.codehaus.cargo.container.configuration.StandaloneConfiguration from the set of implemented
interfaces
ERROR: org.codehaus.cargo.container.resin.Resin2xStandaloneConfiguration: Parameter 1 of
'public Resin2xStandaloneConfiguration(org.codehaus.cargo.container.Container)' has changed its
type to java.io.File
ERROR: org.codehaus.cargo.container.resin.Resin2xStandaloneConfiguration: Method 'public
Resin2xStandaloneConfiguration(org.codehaus.cargo.container.Container, java.io.File)' has been
removed
ERROR: org.codehaus.cargo.container.resin.Resin2xStandaloneConfiguration: In method 'protected
void prepareAdditions(org.apache.tools.ant.types.FilterChain)' the number of arguments has
changed
ERROR: org.codehaus.cargo.container.resin.Resin3xContainer: In method 'public
Resin3xContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.resin.Resin3xStandaloneConfiguration: Removed
org.codehaus.cargo.container.configuration.StandaloneConfiguration from the set of implemented
interfaces
ERROR: org.codehaus.cargo.container.resin.Resin3xStandaloneConfiguration: Parameter 1 of
'public Resin3xStandaloneConfiguration(org.codehaus.cargo.container.Container)' has changed its
type to java.io.File
ERROR: org.codehaus.cargo.container.resin.Resin3xStandaloneConfiguration: Method 'public
Resin3xStandaloneConfiguration(org.codehaus.cargo.container.Container, java.io.File)' has been
removed
ERROR: org.codehaus.cargo.container.resin.Resin3xStandaloneConfiguration: In method 'protected
void prepareAdditions(org.apache.tools.ant.types.FilterChain)' the number of arguments has
changed
ERROR: org.codehaus.cargo.container.resin.ResinExistingConfiguration: Parameter 1 of 'public
ResinExistingConfiguration(org.codehaus.cargo.container.Container)' has changed its type to
java.io.File
ERROR: org.codehaus.cargo.container.resin.ResinExistingConfiguration: Method 'public
ResinExistingConfiguration(org.codehaus.cargo.container.Container, java.io.File)' has been
removed
ERROR: org.codehaus.cargo.container.resin.ResinExistingConfiguration: In method 'public void
configure()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.spi.AbstractConfiguration: In method 'public
AbstractConfiguration(org.codehaus.cargo.container.Container, java.io.File)' the number of
arguments has changed
ERROR: org.codehaus.cargo.container.spi.AbstractConfiguration: Method 'public
org.codehaus.cargo.container.Container getContainer()' has been removed
ERROR: org.codehaus.cargo.container.spi.AbstractContainer: In method 'public
AbstractContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public
org.codehaus.cargo.container.deployable.DeployableFactory getDeployableFactory()' has been
removed
ERROR: org.codehaus.cargo.container.spi.AbstractContainer: Method 'protected void
setDeployableFactory(org.codehaus.cargo.container.deployable.DeployableFactory)' has been
removed
ERROR: org.codehaus.cargo.container.spi.AbstractGenericHintFactory: Class
org.codehaus.cargo.container.spi.AbstractGenericHintFactory removed
ERROR: org.codehaus.cargo.container.spi.AbstractGenericHintFactory$GenericParameters: Class
org.codehaus.cargo.container.spi.AbstractGenericHintFactory$GenericParameters removed
ERROR: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Removed
org.codehaus.cargo.container.configuration.StandaloneConfiguration from the set of implemented
interfaces
ERROR: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Parameter 1 of 'public
AbstractStandaloneConfiguration(org.codehaus.cargo.container.Container)' has changed its type
to java.io.File
ERROR: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Method 'public
AbstractStandaloneConfiguration(org.codehaus.cargo.container.Container, java.io.File)' has been
removed
ERROR: org.codehaus.cargo.container.spi.ContainerConfiguration: In method 'public void
configure()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.spi.DefaultDeployableFactory: Class
org.codehaus.cargo.container.spi.DefaultDeployableFactory removed
ERROR: org.codehaus.cargo.container.spi.DeployerWatchdog: Method 'public void
waitForDeployment()' has been removed
ERROR: org.codehaus.cargo.container.tomcat.AbstractCatalinaContainer: In method 'public
AbstractCatalinaContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.tomcat.AbstractTomcatContainer: In method 'public
AbstractTomcatContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.tomcat.AbstractTomcatStandaloneConfiguration: Removed
org.codehaus.cargo.container.configuration.StandaloneConfiguration from the set of implemented
interfaces
ERROR: org.codehaus.cargo.container.tomcat.AbstractTomcatStandaloneConfiguration: Parameter 1
of 'public AbstractTomcatStandaloneConfiguration(org.codehaus.cargo.container.Container)' has
changed its type to java.io.File
ERROR: org.codehaus.cargo.container.tomcat.AbstractTomcatStandaloneConfiguration: Method
'public AbstractTomcatStandaloneConfiguration(org.codehaus.cargo.container.Container,
java.io.File)' has been removed
```

```
ERROR: org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration: Removed
org.codehaus.cargo.container.configuration.StandaloneConfiguration from the set of implemented
interfaces
ERROR: org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration: Parameter 1 of
'public CatalinaStandaloneConfiguration(org.codehaus.cargo.container.Container)' has changed
its type to java.io.File
ERROR: org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration: Method 'public
CatalinaStandaloneConfiguration(org.codehaus.cargo.container.Container, java.io.File)' has been
removed
ERROR: org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration: In method 'public
void configure()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.tomcat.Tomcat3xContainer: In method 'public
Tomcat3xContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.tomcat.Tomcat4xContainer: In method 'public
Tomcat4xContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.tomcat.Tomcat5xContainer: In method 'public
Tomcat5xContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.tomcat.TomcatDeployableFactory: Class
org.codehaus.cargo.container.tomcat.TomcatDeployableFactory removed
ERROR: org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration: Removed
org.codehaus.cargo.container.configuration.StandaloneConfiguration from the set of implemented
interfaces
ERROR: org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration: Parameter 1 of
'public TomcatStandaloneConfiguration(org.codehaus.cargo.container.Container)' has changed its
type to java.io.File
ERROR: org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration: Method 'public
TomcatStandaloneConfiguration(org.codehaus.cargo.container.Container, java.io.File)' has been
removed
ERROR: org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration: In method 'public
void configure()'the number of arguments has changed
ERROR: org.codehaus.cargo.container.weblogic.AbstractWebLogicContainer: In method 'public
AbstractWebLogicContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.weblogic.WebLogic8xContainer: In method 'public
WebLogic8xContainer()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: Removed
org.codehaus.cargo.container.configuration.StandaloneConfiguration from the set of implemented
interfaces
ERROR: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: Parameter 1 of
'public WebLogicStandaloneConfiguration(org.codehaus.cargo.container.Container)' has changed
its type to java.io.File
ERROR: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: Method 'public
WebLogicStandaloneConfiguration(org.codehaus.cargo.container.Container, java.io.File)' has been
removed
ERROR: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: In method 'public
void configure()' the number of arguments has changed
ERROR: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: In method
'protected void setupDeployables(java.io.File)' the number of arguments has changed
ERROR: org.codehaus.cargo.module.Dtd$XmlEntityResolver: Class
org.codehaus.cargo.module.Dtd$XmlEntityResolverremoved
```

Changes not affecting compatibility:

```
INFO: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Method 'public
java.lang.String toString()' has been added
INFO: org.codehaus.cargo.ant.CargoTask: Method 'protected java.lang.Class getContainerClass()'
has been added
INFO: org.codehaus.cargo.ant.CargoTask: Method 'protected java.lang.String getContainerId()'
has been added
INFO: org.codehaus.cargo.ant.CargoTask: Method 'protected
org.codehaus.cargo.util.monitor.Monitor getMonitor()' has been added
INFO: org.codehaus.cargo.ant.CargoTask: Method 'protected
org.codehaus.cargo.container.Container makeContainer()' has been added
INFO: org.codehaus.cargo.ant.CargoTask: Method 'public void setClass(java.lang.Class)' has been
added
INFO: org.codehaus.cargo.ant.CargoTask: Method 'public void
setContainerFactory(org.codehaus.cargo.generic.ContainerFactory)' has been added
INFO: org.codehaus.cargo.ant.CargoTask: Method 'public void setContainerId(java.lang.String)'
has been added
INFO: org.codehaus.cargo.ant.ConfigurationElement: Method 'public
org.codehaus.cargo.generic.configuration.ConfigurationType getType()' has been added
INFO: org.codehaus.cargo.ant.ConfigurationElement: Method 'public void
setType(java.lang.String)' has been added
INFO: org.codehaus.cargo.container.jo.Jo1xStandaloneConfiguration: Method 'public
java.lang.String toString()' has been added
```

```
INFO: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Method 'public
java.lang.String toString()' has been added
INFO: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: Method 'public
java.lang.String toString()' has been added
INFO: org.codehaus.cargo.container.resin.ResinExistingConfiguration: Method 'public
java.lang.String toString()' has been added
INFO: org.codehaus.cargo.container.spi.AbstractConfiguration: Method 'public void
addDeployable(org.codehaus.cargo.container.deployable.Deployable)' has been added
INFO: org.codehaus.cargo.container.spi.AbstractConfiguration: Method 'public java.util.List
getDeployables()' has been added
INFO: org.codehaus.cargo.container.spi.AbstractCopyingDeployer: Method 'public void
deploy(java.util.List)' has been added
INFO: org.codehaus.cargo.container.spi.AbstractCopyingDeployer: Method 'public void
setShouldDeployExpandedWARs(boolean)' has been added
INFO: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Method 'public void
addDeployable(org.codehaus.cargo.container.deployable.Deployable)' has been removed, but an
inherited definition exists.
INFO: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Method 'public
java.util.List getDeployables()' has been removed, but an inherited definition exists.
INFO: org.codehaus.cargo.container.spi.DeployerWatchdog: Method 'public void undeployed()' has
been added
INFO: org.codehaus.cargo.container.spi.DeployerWatchdog: Method 'public void watch(boolean)'
has been added
INFO: org.codehaus.cargo.container.spi.DeployerWatchdog: Method 'public void
watchForAvailability()' has been added
INFO: org.codehaus.cargo.container.spi.DeployerWatchdog: Method 'public void
watchForUnavailability()' has been added
INFO: org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration: Method 'public
java.lang.String toString()' has been added
INFO: org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration: Method 'public
java.lang.String toString()' has been added
INFO: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: Method 'public
java.lang.String toString()' has been added
INFO: org.codehaus.cargo.generic.ContainerFactory: Class
org.codehaus.cargo.generic.ContainerFactory added
INFO: org.codehaus.cargo.generic.DefaultContainerFactory: Class
org.codehaus.cargo.generic.DefaultContainerFactory added
INFO: org.codehaus.cargo.generic.configuration.ConfigurationFactory: Class
org.codehaus.cargo.generic.configuration.ConfigurationFactory added
INFO: org.codehaus.cargo.generic.configuration.ConfigurationType: Class
org.codehaus.cargo.generic.configuration.ConfigurationType added
INFO: org.codehaus.cargo.generic.configuration.DefaultConfigurationFactory: Class
org.codehaus.cargo.generic.configuration.DefaultConfigurationFactory added
INFO:
org.codehaus.cargo.generic.configuration.DefaultConfigurationFactory$ConfigurationFactoryParameters:
Class
org.codehaus.cargo.generic.configuration.DefaultConfigurationFactory$ConfigurationFactoryParameters
added
INFO: org.codehaus.cargo.generic.deployable.DefaultDeployableFactory: Class
org.codehaus.cargo.generic.deployable.DefaultDeployableFactory added
INFO: org.codehaus.cargo.generic.deployable.DeployableFactory: Class
org.codehaus.cargo.generic.deployable.DeployableFactory added
INFO: org.codehaus.cargo.generic.deployable.DeployableType: Class
org.codehaus.cargo.generic.deployable.DeployableType added
INFO: org.codehaus.cargo.generic.deployer.DefaultDeployerFactory: Class
org.codehaus.cargo.generic.deployer.DefaultDeployerFactory added
INFO: org.codehaus.cargo.generic.deployer.DefaultDeployerFactory$DeployerFactoryParameters:
Class org.codehaus.cargo.generic.deployer.DefaultDeployerFactory$DeployerFactoryParameters
added
INFO: org.codehaus.cargo.generic.deployer.DeployerFactory: Class
org.codehaus.cargo.generic.deployer.DeployerFactory added
INFO: org.codehaus.cargo.generic.internal.util.HintKey: Class
org.codehaus.cargo.generic.internal.util.HintKey added
INFO: org.codehaus.cargo.generic.spi.AbstractGenericHintFactory: Class
org.codehaus.cargo.generic.spi.AbstractGenericHintFactory added
INFO: org.codehaus.cargo.generic.spi.AbstractGenericHintFactory$GenericParameters: Class
org.codehaus.cargo.generic.spi.AbstractGenericHintFactory$GenericParameters added
INFO: org.codehaus.cargo.module.XmlEntityResolver: Class
org.codehaus.cargo.module.XmlEntityResolver added
INFO: org.codehaus.cargo.module.webapp.WebXmlVersion: Method 'public boolean
equals(java.lang.Object)' has been removed, but an inherited definition exists.
INFO: org.codehaus.cargo.module.webapp.WebXmlVersion: Method 'public int hashCode()' has been
removed, but an inherited definition exists.
```

Release notes for Cargo 0.7

This page last changed on Dec 30, 2005 by vmassol.

Main changes

Here are the highlights of Cargo 0.7:

- Major refactoring to support Remote containers (i.e. containers that are started somewhere and with which we don't interact through the file system).
- Added JBoss 3.x and 4.x support.
- Added new Deployable type: EJB. Note that this new deployable is supported only by the JBoss container.
- Added a Tomcat Existing Configuration implementation.
- Added a Tomcat Local Deployer and a Tomcat Remote Deployer which uses the Tomcat Manager app for deployment.
- Added a WebLogic Existing Configuration implementation.

The list of all issues fixed and features implemented is available below.

Please note that version 0.7 has broken almost all APIs. This was majoritarly due to the Remote container support refactoring. This is one of the last 0.X version before the 1.0 release and we took the opportunity to fix the API now rather than to go through a lengthy and difficult deprecation strategy (which we'll use once 1.0 is out). We apologize to all our early users for this. This is probably the last time such a major change happens before the 1.0 release.

Implemented issues

jira.codehaus.org (25 issues)					
T	Key	Res	Summary	Assignee	Reporter
	CARGO-235	FIXED	Add support for EJB deployments	Vincent Massol	Vincent Massol
	CARGO-234	FIXED	Add support for EJB deployables	Vincent Massol	Vincent Massol
	CARGO-233	FIXED	Replace <war> and <ear> config element by a generic <deployable> one in the Cargo Ant task	Vincent Massol	Vincent Massol
	CARGO-232	FIXED	ClassCastException when trying to deploy an EAR in Tomcat 4.x or 5.x	Vincent Massol	Vincent Massol
	CARGO-228	FIXED	Error while performing a remote	Vincent Massol	Aleksander Blomskøld

			deployment with Tomcat		
	CARGO-226	WON'T FIX	.project & .classpat files for eclipse are in the repo, should be in svn:ignore property	Vincent Massol	Bill Dudney
	CARGO-225	FIXED	DefaultConfigurationFactory fails when creating runtime configurations	Vincent Massol	Aleksander Blomskøld
	CARGO-223	WON'T FIX	no way to specify the manager to use in the tomcat impl	Vincent Massol	Bill Dudney
	CARGO-220	FIXED	Rename all m2 groupIds to org.codehaus.cargo	Vincent Massol	Vincent Massol
	CARGO-219	FIXED	Create a core/uberjar project to generate an aggregated core jar (uberjar)	Vincent Massol	Vincent Massol
	CARGO-218	FIXED	Removed creation of default configurations when creating a container using a ContainerFactory	Vincent Massol	Vincent Massol
	CARGO-215	FIXED	NPE if no container output file is defined by the user	Vincent Massol	Vincent Massol
	CARGO-213	FIXED	Rename homeDir property in home in LocalContainer and Installer to align it with LocalConfiguration.	Vincent Massol	Vincent Massol
	CARGO-212	DUPLICATE	Websphere support	Vincent Massol	Malcolm Wong Ho
	CARGO-211	FIXED	API for implementing merging	Magnus Grimsell	Nigel Magnay
	CARGO-209	FIXED	Base classes for wrapping Elements	Magnus Grimsell	Nigel Magnay
	CARGO-207	DUPLICATE	Improve War file support to allow merging of files other than	Nigel Magnay	Nigel Magnay

	CARGO-206	FIXED	web.xml Incorrect Home URL passed to JBoss 3.x	I Nyoman Winardi	Binil Thomas
	CARGO-205	FIXED	Add API to check if a given deployer has been registered, when using the generic API	Vincent Massol	Vincent Massol
	CARGO-204	FIXED	Create Tomcat classes for supporting remote deployments	Vincent Massol	Vincent Massol
	CARGO-203	FIXED	NPE when deploying tomcat on ant	Vincent Massol	Steve Loughran
	CARGO-202	FIXED	Add notion of Deployer type to represent local and remote deployers	Vincent Massol	Vincent Massol
	CARGO-201	FIXED	Add getType() method to the Configuration interface	Vincent Massol	Vincent Massol
	CARGO-200	FIXED	Add getType() method to the Deployable interface	Vincent Massol	Vincent Massol
	CARGO-199	FIXED	Add getType() method to the Container interface	Vincent Massol	Vincent Massol

Source code changes

Changes detected by Clirr between `cargo-0.6.jar` and `cargo-0.7.jar`.

Changes to the core:

```
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public
org.codehaus.cargo.container.configuration.Configuration getConfiguration()' has been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public java.lang.String[]
getExtraClasspath()' has been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public java.io.File getHomeDir()'
has been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public java.io.File getOutput()'
has been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public java.util.Map
getSystemProperties()' has been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public long getTimeout()' has been
removed
ERROR: 7012: org.codehaus.cargo.container.Container: Method 'public
org.codehaus.cargo.container.ContainerType getType()' has been added to an interface
```

```
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public boolean isAppend()' has
been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public void setAppend(boolean)'
has been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public void
setConfiguration(org.codehaus.cargo.container.configuration.Configuration)' has been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public void
setExtraClasspath(java.lang.String[])' has been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public void
setHomeDir(java.io.File)' has been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public void
setHomeDir(java.lang.String)' has been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public void
setOutput(java.io.File)' has been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public void
setSystemProperties(java.util.Map)' has been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public void setTimeout(long)' has
been removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public void start()' has been
removed
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public void stop()' has been
removed
ERROR: 7012: org.codehaus.cargo.container.ContainerCapability: Method 'public boolean
supportsDeployableType(org.codehaus.cargo.container.deployable.DeployableType)' has been added
to an interface
ERROR: 7002: org.codehaus.cargo.container.ContainerCapability: Method 'public boolean
supportsEar()' has been removed
ERROR: 7002: org.codehaus.cargo.container.ContainerCapability: Method 'public boolean
supportsWar()' has been removed
INFO: 8000: org.codehaus.cargo.container.ContainerType: Class
org.codehaus.cargo.container.ContainerType added
INFO: 8000: org.codehaus.cargo.container.LocalContainer: Class
org.codehaus.cargo.container.LocalContainer added
INFO: 8000: org.codehaus.cargo.container.RemoteContainer: Class
org.codehaus.cargo.container.RemoteContainer added
ERROR: 7002: org.codehaus.cargo.container.configuration.Configuration: Method 'public void
addDeployable(org.codehaus.cargo.container.deployable.Deployable)' has been removed
ERROR: 7002: org.codehaus.cargo.container.configuration.Configuration: Method 'public
java.util.List getDeployables()' has been removed
ERROR: 7002: org.codehaus.cargo.container.configuration.Configuration: Method 'public
java.io.File getDir()' has been removed
ERROR: 7012: org.codehaus.cargo.container.configuration.Configuration: Method 'public
org.codehaus.cargo.container.configuration.ConfigurationType getType()' has been added to an
interface
INFO: 8000: org.codehaus.cargo.container.configuration.ConfigurationType: Class
org.codehaus.cargo.container.configuration.ConfigurationType added
INFO: 8000: org.codehaus.cargo.container.configuration.ExistingLocalConfiguration: Class
org.codehaus.cargo.container.configuration.ExistingLocalConfiguration added
INFO: 8000: org.codehaus.cargo.container.configuration.LocalConfiguration: Class
org.codehaus.cargo.container.configuration.LocalConfiguration added
INFO: 8000: org.codehaus.cargo.container.configuration.RuntimeConfiguration: Class
org.codehaus.cargo.container.configuration.RuntimeConfiguration added
INFO: 8000: org.codehaus.cargo.container.configuration.StandaloneLocalConfiguration: Class
org.codehaus.cargo.container.configuration.StandaloneLocalConfiguration added
ERROR: 7012: org.codehaus.cargo.container.deployable.Deployable: Method 'public
org.codehaus.cargo.container.deployable.DeployableType getType()' has been added to an
interface
INFO: 8000: org.codehaus.cargo.container.deployable.DeployableType: Class
org.codehaus.cargo.container.deployable.DeployableType added
INFO: 7011: org.codehaus.cargo.container.deployable.EAR: Method 'public
org.codehaus.cargo.container.deployable.DeployableType getType()' has been added
INFO: 8000: org.codehaus.cargo.container.deployable.EJB: Class
org.codehaus.cargo.container.deployable.EJB added
INFO: 7011: org.codehaus.cargo.container.deployable.WAR: Method 'public
org.codehaus.cargo.container.deployable.DeployableType getType()' has been added
ERROR: 8001: org.codehaus.cargo.container.deployable.jboss.JBossWAR: Class
org.codehaus.cargo.container.deployable.jboss.JBossWAR removed
ERROR: 8001: org.codehaus.cargo.container.deployable.tomcat.TomcatWAR: Class
org.codehaus.cargo.container.deployable.tomcat.TomcatWAR removed
ERROR: 7012: org.codehaus.cargo.container.deployer.Deployer: Method 'public
org.codehaus.cargo.container.deployer.DeployerType getType()' has been added to an interface
ERROR: 7012: org.codehaus.cargo.container.deployer.Deployer: Method 'public void
redeploy(org.codehaus.cargo.container.deployable.Deployable)' has been added to an interface
INFO: 8000: org.codehaus.cargo.container.deployer.DeployerType: Class
org.codehaus.cargo.container.deployer.DeployerType added
ERROR: 7012: org.codehaus.cargo.container.installer.Installer: Method 'public java.io.File
getHome()' has been added to an interface
ERROR: 7002: org.codehaus.cargo.container.installer.Installer: Method 'public java.io.File
```

```
getHomeDir()' has been removed
INFO: 7011: org.codehaus.cargo.container.installer.ZipURLInstaller: Method 'public java.io.File
getHome()' has been added
ERROR: 7002: org.codehaus.cargo.container.installer.ZipURLInstaller: Method 'public
java.io.File getHomeDir()' has been removed
INFO: 7011: org.codehaus.cargo.container.internal.J2EEContainerCapability: Method 'public
boolean supportsDeployableType(org.codehaus.cargo.container.deployable.DeployableType)' has
been added
ERROR: 7002: org.codehaus.cargo.container.internal.J2EEContainerCapability: Method 'public
boolean supportsEar()' has been removed
ERROR: 7002: org.codehaus.cargo.container.internal.J2EEContainerCapability: Method 'public
boolean supportsWar()' has been removed
INFO: 8000: org.codehaus.cargo.container.internal.RunnableContainer: Class
org.codehaus.cargo.container.internal.RunnableContainer added
INFO: 7011: org.codehaus.cargo.container.internal.ServletContainerCapability: Method 'public
boolean supportsDeployableType(org.codehaus.cargo.container.deployable.DeployableType)' has
been added
ERROR: 7002: org.codehaus.cargo.container.internal.ServletContainerCapability: Method 'public
boolean supportsEar()' has been removed
ERROR: 7002: org.codehaus.cargo.container.internal.ServletContainerCapability: Method 'public
boolean supportsWar()' has been removed
ERROR: 8001: org.codehaus.cargo.container.internal.jetty.JettyExecutorThread: Class
org.codehaus.cargo.container.internal.jetty.JettyExecutorThread removed
ERROR: 8001:
org.codehaus.cargo.container.internal.jetty.JettyStandaloneConfigurationCapability: Class
org.codehaus.cargo.container.internal.jetty.JettyStandaloneConfigurationCapability removed
ERROR: 8001: org.codehaus.cargo.container.internal.jo.Jo1xStandaloneConfigurationCapability:
Class org.codehaus.cargo.container.internal.jo.Jo1xStandaloneConfigurationCapability removed
ERROR: 8001:
org.codehaus.cargo.container.internal.orion.OrionStandaloneConfigurationCapability: Class
org.codehaus.cargo.container.internal.orion.OrionStandaloneConfigurationCapability removed
ERROR: 8001: org.codehaus.cargo.container.internal.resin.ResinRun: Class
org.codehaus.cargo.container.internal.resin.ResinRun removed
ERROR: 8001:
org.codehaus.cargo.container.internal.resin.ResinStandaloneConfigurationCapability: Class
org.codehaus.cargo.container.internal.resin.ResinStandaloneConfigurationCapability removed
ERROR: 8001: org.codehaus.cargo.container.internal.resin.ResinUtil: Class
org.codehaus.cargo.container.internal.resin.ResinUtil removed
ERROR: 8001:
org.codehaus.cargo.container.internal.tomcat.TomcatStandaloneConfigurationCapability: Class
org.codehaus.cargo.container.internal.tomcat.TomcatStandaloneConfigurationCapability removed
INFO: 7011: org.codehaus.cargo.container.internal.util.AntUtils: Method 'public
org.apache.tools.ant.types.Environment$Variable createSysProperty(java.lang.String,
java.net.URI)' has been added
ERROR: 8001:
org.codehaus.cargo.container.internal.weblogic.WebLogicStandaloneConfigurationCapability: Class
org.codehaus.cargo.container.internal.weblogic.WebLogicStandaloneConfigurationCapability
removed
INFO: 8000: org.codehaus.cargo.container.jboss.JBoss3xLocalContainer: Class
org.codehaus.cargo.container.jboss.JBoss3xLocalContainer added
INFO: 8000: org.codehaus.cargo.container.jboss.JBoss4xLocalContainer: Class
org.codehaus.cargo.container.jboss.JBoss4xLocalContainer added
INFO: 8000: org.codehaus.cargo.container.jboss.JBossDeployer: Class
org.codehaus.cargo.container.jboss.JBossDeployer added
INFO: 8000: org.codehaus.cargo.container.jboss.JBossExistingLocalConfiguration: Class
org.codehaus.cargo.container.jboss.JBossExistingLocalConfiguration added
INFO: 8000: org.codehaus.cargo.container.jboss.JBossPropertySet: Class
org.codehaus.cargo.container.jboss.JBossPropertySet added
INFO: 8000: org.codehaus.cargo.container.jboss.JBossStandaloneLocalConfiguration: Class
org.codehaus.cargo.container.jboss.JBossStandaloneLocalConfiguration added
INFO: 8000: org.codehaus.cargo.container.jboss.JBossWAR: Class
org.codehaus.cargo.container.jboss.JBossWAR added
INFO: 8000: org.codehaus.cargo.container.jboss.internal.AbstractJBossLocalContainer: Class
org.codehaus.cargo.container.jboss.internal.AbstractJBossLocalContainer added
INFO: 8000: org.codehaus.cargo.container.jboss.internal.JBossContainerCapability: Class
org.codehaus.cargo.container.jboss.internal.JBossContainerCapability added
INFO: 8000:
org.codehaus.cargo.container.jboss.internal.JBossStandaloneLocalConfigurationCapability: Class
org.codehaus.cargo.container.jboss.internal.JBossStandaloneLocalConfigurationCapability added
ERROR: 8001: org.codehaus.cargo.container.jetty.Jetty4xEmbeddedContainer: Class
org.codehaus.cargo.container.jetty.Jetty4xEmbeddedContainer removed
INFO: 8000: org.codehaus.cargo.container.jetty.Jetty4xEmbeddedLocalContainer: Class
org.codehaus.cargo.container.jetty.Jetty4xEmbeddedLocalContainer added
ERROR: 7005: org.codehaus.cargo.container.jetty.JettyDeployer: Parameter 1 of 'public
JettyDeployer(org.codehaus.cargo.container.Container)' has changed its type to
org.codehaus.cargo.container.LocalContainer
INFO: 7011: org.codehaus.cargo.container.jetty.JettyDeployer: Method 'public
org.codehaus.cargo.container.deployer.DeployerType getType()' has been added
```

```
INFO: 7011: org.codehaus.cargo.container.jetty.JettyDeployer: Method 'public void
redeploy(org.codehaus.cargo.container.deployable.Deployable)' has been added
ERROR: 8001: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Class
org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration removed
INFO: 8000: org.codehaus.cargo.container.jetty.JettyStandaloneLocalConfiguration: Class
org.codehaus.cargo.container.jetty.JettyStandaloneLocalConfiguration added
INFO: 8000: org.codehaus.cargo.container.jetty.internal.JettyExecutorThread: Class
org.codehaus.cargo.container.jetty.internal.JettyExecutorThread added
INFO: 8000:
org.codehaus.cargo.container.jetty.internal.JettyStandaloneLocalConfigurationCapability: Class
org.codehaus.cargo.container.jetty.internal.JettyStandaloneLocalConfigurationCapability added
ERROR: 8001: org.codehaus.cargo.container.jo.Jo1xContainer: Class
org.codehaus.cargo.container.jo.Jo1xContainer removed
ERROR: 5001: org.codehaus.cargo.container.jo.Jo1xDeployer: Removed
org.codehaus.cargo.container.spi.AbstractCopyingDeployer from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.jo.Jo1xDeployer: Added
org.codehaus.cargo.container.spi.deployer.AbstractCopyingDeployer to the list of superclasses
INFO: 5000: org.codehaus.cargo.container.jo.Jo1xDeployer: Added
org.codehaus.cargo.container.spi.deployer.AbstractDeployer to the list of superclasses
INFO: 5000: org.codehaus.cargo.container.jo.Jo1xDeployer: Added
org.codehaus.cargo.container.spi.deployer.AbstractLocalDeployer to the list of superclasses
ERROR: 7005: org.codehaus.cargo.container.jo.Jo1xDeployer: Parameter 1 of 'public
Jo1xDeployer(org.codehaus.cargo.container.Container)' has changed its type to
org.codehaus.cargo.container.LocalContainer
INFO: 8000: org.codehaus.cargo.container.jo.Jo1xLocalContainer: Class
org.codehaus.cargo.container.jo.Jo1xLocalContainer added
ERROR: 8001: org.codehaus.cargo.container.jo.Jo1xStandaloneConfiguration: Class
org.codehaus.cargo.container.jo.Jo1xStandaloneConfiguration removed
INFO: 8000: org.codehaus.cargo.container.jo.Jo1xStandaloneLocalConfiguration: Class
org.codehaus.cargo.container.jo.Jo1xStandaloneLocalConfiguration added
INFO: 8000:
org.codehaus.cargo.container.jo.internal.Jo1xStandaloneLocalConfigurationCapability: Class
org.codehaus.cargo.container.jo.internal.Jo1xStandaloneLocalConfigurationCapability added
INFO: 8000: org.codehaus.cargo.container.jsr88.GenericJSR88Configuration: Class
org.codehaus.cargo.container.jsr88.GenericJSR88Configuration added
INFO: 8000: org.codehaus.cargo.container.jsr88.JSR88ConfigurationCapability: Class
org.codehaus.cargo.container.jsr88.JSR88ConfigurationCapability added
INFO: 8000: org.codehaus.cargo.container.jsr88.JSR88Deployer: Class
org.codehaus.cargo.container.jsr88.JSR88Deployer added
ERROR: 8001: org.codehaus.cargo.container.orion.AbstractOrionContainer: Class
org.codehaus.cargo.container.orion.AbstractOrionContainer removed
ERROR: 8001: org.codehaus.cargo.container.orion.Oc4j9xContainer: Class
org.codehaus.cargo.container.orion.Oc4j9xContainer removed
INFO: 8000: org.codehaus.cargo.container.orion.Oc4j9xLocalContainer: Class
org.codehaus.cargo.container.orion.Oc4j9xLocalContainer added
ERROR: 8001: org.codehaus.cargo.container.orion.Orion1xContainer: Class
org.codehaus.cargo.container.orion.Orion1xContainer removed
INFO: 8000: org.codehaus.cargo.container.orion.Orion1xLocalContainer: Class
org.codehaus.cargo.container.orion.Orion1xLocalContainer added
ERROR: 8001: org.codehaus.cargo.container.orion.Orion2xContainer: Class
org.codehaus.cargo.container.orion.Orion2xContainer removed
INFO: 8000: org.codehaus.cargo.container.orion.Orion2xLocalContainer: Class
org.codehaus.cargo.container.orion.Orion2xLocalContainer added
ERROR: 8001: org.codehaus.cargo.container.orion.OrionStandaloneConfiguration: Class
org.codehaus.cargo.container.orion.OrionStandaloneConfiguration removed
INFO: 8000: org.codehaus.cargo.container.orion.OrionStandaloneLocalConfiguration: Class
org.codehaus.cargo.container.orion.OrionStandaloneLocalConfiguration added
INFO: 8000: org.codehaus.cargo.container.orion.internal.AbstractOrionLocalContainer: Class
org.codehaus.cargo.container.orion.internal.AbstractOrionLocalContainer added
INFO: 8000:
org.codehaus.cargo.container.orion.internal.OrionStandaloneLocalConfigurationCapability: Class
org.codehaus.cargo.container.orion.internal.OrionStandaloneLocalConfigurationCapability added
INFO: 6000: org.codehaus.cargo.container.property.GeneralPropertySet: Added public field
PROTOCOL
INFO: 8000: org.codehaus.cargo.container.property.JSR88PropertySet: Class
org.codehaus.cargo.container.property.JSR88PropertySet added
INFO: 8000: org.codehaus.cargo.container.property.RemotePropertySet: Class
org.codehaus.cargo.container.property.RemotePropertySet added
ERROR: 8001: org.codehaus.cargo.container.resin.AbstractResinContainer: Class
org.codehaus.cargo.container.resin.AbstractResinContainer removed
ERROR: 8001: org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration: Class
org.codehaus.cargo.container.resin.AbstractResinStandaloneConfiguration removed
ERROR: 8001: org.codehaus.cargo.container.resin.Resin2xContainer: Class
org.codehaus.cargo.container.resin.Resin2xContainer removed
INFO: 8000: org.codehaus.cargo.container.resin.Resin2xLocalContainer: Class
org.codehaus.cargo.container.resin.Resin2xLocalContainer added
ERROR: 8001: org.codehaus.cargo.container.resin.Resin2xStandaloneConfiguration: Class
org.codehaus.cargo.container.resin.Resin2xStandaloneConfiguration removed
```

```
INFO: 8000: org.codehaus.cargo.container.resin.Resin2xStandaloneLocalConfiguration: Class
org.codehaus.cargo.container.resin.Resin2xStandaloneLocalConfiguration added
ERROR: 8001: org.codehaus.cargo.container.resin.Resin3xContainer: Class
org.codehaus.cargo.container.resin.Resin3xContainer removed
INFO: 8000: org.codehaus.cargo.container.resin.Resin3xLocalContainer: Class
org.codehaus.cargo.container.resin.Resin3xLocalContainer added
ERROR: 8001: org.codehaus.cargo.container.resin.Resin3xStandaloneConfiguration: Class
org.codehaus.cargo.container.resin.Resin3xStandaloneConfiguration removed
INFO: 8000: org.codehaus.cargo.container.resin.Resin3xStandaloneLocalConfiguration: Class
org.codehaus.cargo.container.resin.Resin3xStandaloneLocalConfiguration added
ERROR: 5001: org.codehaus.cargo.container.resin.ResinDeployer: Removed
org.codehaus.cargo.container.spi.AbstractCopyingDeployer from the list of superclasses
INFO: 5000: org.codehaus.cargo.container.resin.ResinDeployer: Added
org.codehaus.cargo.container.spi.deployer.AbstractCopyingDeployer to the list of superclasses
INFO: 5000: org.codehaus.cargo.container.resin.ResinDeployer: Added
org.codehaus.cargo.container.spi.deployer.AbstractDeployer to the list of superclasses
INFO: 5000: org.codehaus.cargo.container.resin.ResinDeployer: Added
org.codehaus.cargo.container.spi.deployer.AbstractLocalDeployer to the list of superclasses
ERROR: 7005: org.codehaus.cargo.container.resin.ResinDeployer: Parameter 1 of 'public
ResinDeployer(org.codehaus.cargo.container.Container)' has changed its type to
org.codehaus.cargo.container.LocalContainer
ERROR: 8001: org.codehaus.cargo.container.resin.ResinExistingConfiguration: Class
org.codehaus.cargo.container.resin.ResinExistingConfiguration removed
INFO: 8000: org.codehaus.cargo.container.resin.ResinExistingLocalConfiguration: Class
org.codehaus.cargo.container.resin.ResinExistingLocalConfiguration added
INFO: 8000: org.codehaus.cargo.container.resin.internal.AbstractResinLocalContainer: Class
org.codehaus.cargo.container.resin.internal.AbstractResinLocalContainer added
INFO: 8000:
org.codehaus.cargo.container.resin.internal.AbstractResinStandaloneLocalConfiguration: Class
org.codehaus.cargo.container.resin.internal.AbstractResinStandaloneLocalConfiguration added
INFO: 8000: org.codehaus.cargo.container.resin.internal.ResinRun: Class
org.codehaus.cargo.container.resin.internal.ResinRun added
INFO: 8000:
org.codehaus.cargo.container.resin.internal.ResinStandaloneLocalConfigurationCapability: Class
org.codehaus.cargo.container.resin.internal.ResinStandaloneLocalConfigurationCapability added
INFO: 8000: org.codehaus.cargo.container.resin.internal.ResinUtil: Class
org.codehaus.cargo.container.resin.internal.ResinUtil added
ERROR: 8001: org.codehaus.cargo.container.spi.AbstractConfiguration: Class
org.codehaus.cargo.container.spi.AbstractConfiguration removed
ERROR: 7004: org.codehaus.cargo.container.spi.AbstractContainer: In method 'public
AbstractContainer(org.codehaus.cargo.container.configuration.Configuration)' the number of
arguments has changed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'protected void
addToolsJarToClasspath(org.apache.tools.ant.types.Path)' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'protected void
doStart(org.apache.tools.ant.taskdefs.Java)' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'protected void
doStop(org.apache.tools.ant.taskdefs.Java)' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'protected
org.codehaus.cargo.container.internal.util.AntUtils getAntUtils()' has been removed
ERROR: 7013: org.codehaus.cargo.container.spi.AbstractContainer: Abstract method 'public
org.codehaus.cargo.container.ContainerCapability getCapability()' has been added
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public
org.codehaus.cargo.container.configuration.Configuration getConfiguration()' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public
java.lang.String[] getExtraClasspath()' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'protected
org.codehaus.cargo.util.FileUtils getFileUtils()' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public java.io.File
getHomeDir()' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'protected
org.codehaus.cargo.container.internal.util.HttpUtils getHttpUtils()' has been removed
ERROR: 7013: org.codehaus.cargo.container.spi.AbstractContainer: Abstract method 'public
java.lang.String getId()' has been added
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'protected
org.codehaus.cargo.container.internal.util.JdkUtils getJdkUtils()' has been removed
ERROR: 7013: org.codehaus.cargo.container.spi.AbstractContainer: Abstract method 'public
java.lang.String getName()' has been added
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public java.io.File
getOutput()' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'protected
org.codehaus.cargo.container.internal.util.ResourceUtils getResourceUtils()' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public java.util.Map
getSystemProperties()' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public long
getTimeout()' has been removed
ERROR: 7013: org.codehaus.cargo.container.spi.AbstractContainer: Abstract method 'public
org.codehaus.cargo.container.ContainerType getType()' has been added
```

```
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public boolean
isAppend()' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public void
setAppend(boolean)' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public void
setConfiguration(org.codehaus.cargo.container.configuration.Configuration)' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public void
setExtraClasspath(java.lang.String[])' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public void
setHomeDir(java.io.File)' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public void
setHomeDir(java.lang.String)' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public void
setOutput(java.io.File)' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'protected void
setState(org.codehaus.cargo.container.State)' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public void
setSystemProperties(java.util.Map)' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public void
setTimeout(long)' has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public void start()'
has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'public void stop()'
has been removed
ERROR: 7002: org.codehaus.cargo.container.spi.AbstractContainer: Method 'protected void
verifyHomeDir()' has been removed
ERROR: 8001: org.codehaus.cargo.container.spi.AbstractCopyingDeployer: Class
org.codehaus.cargo.container.spi.AbstractCopyingDeployer removed
INFO: 8000: org.codehaus.cargo.container.spi.AbstractLocalContainer: Class
org.codehaus.cargo.container.spi.AbstractLocalContainer added
INFO: 8000: org.codehaus.cargo.container.spi.AbstractRemoteContainer: Class
org.codehaus.cargo.container.spi.AbstractRemoteContainer added
ERROR: 8001: org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration: Class
org.codehaus.cargo.container.spi.AbstractStandaloneConfiguration removed
ERROR: 8001: org.codehaus.cargo.container.spi.AbstractStandaloneConfigurationCapability: Class
org.codehaus.cargo.container.spi.AbstractStandaloneConfigurationCapability removed
ERROR: 8001: org.codehaus.cargo.container.spi.ContainerConfiguration: Class
org.codehaus.cargo.container.spi.ContainerConfiguration removed
ERROR: 8001: org.codehaus.cargo.container.spi.DefaultServerRun: Class
org.codehaus.cargo.container.spi.DefaultServerRun removed
ERROR: 8001: org.codehaus.cargo.container.spi.DeployerWatchdog: Class
org.codehaus.cargo.container.spi.DeployerWatchdog removed
INFO: 8000: org.codehaus.cargo.container.spi.configuration.AbstractConfiguration: Class
org.codehaus.cargo.container.spi.configuration.AbstractConfiguration added
INFO: 8000: org.codehaus.cargo.container.spi.configuration.AbstractExistingLocalConfiguration:
Class org.codehaus.cargo.container.spi.configuration.AbstractExistingLocalConfiguration added
INFO: 8000: org.codehaus.cargo.container.spi.configuration.AbstractLocalConfiguration: Class
org.codehaus.cargo.container.spi.configuration.AbstractLocalConfiguration added
INFO: 8000: org.codehaus.cargo.container.spi.configuration.AbstractRuntimeConfiguration: Class
org.codehaus.cargo.container.spi.configuration.AbstractRuntimeConfiguration added
INFO: 8000:
org.codehaus.cargo.container.spi.configuration.AbstractRuntimeConfigurationCapability: Class
org.codehaus.cargo.container.spi.configuration.AbstractRuntimeConfigurationCapability added
INFO: 8000:
org.codehaus.cargo.container.spi.configuration.AbstractStandaloneLocalConfiguration: Class
org.codehaus.cargo.container.spi.configuration.AbstractStandaloneLocalConfiguration added
INFO: 8000:
org.codehaus.cargo.container.spi.configuration.AbstractStandaloneLocalConfigurationCapability:
Class
org.codehaus.cargo.container.spi.configuration.AbstractStandaloneLocalConfigurationCapability
added
INFO: 8000: org.codehaus.cargo.container.spi.configuration.ContainerConfiguration: Class
org.codehaus.cargo.container.spi.configuration.ContainerConfiguration added
INFO: 8000: org.codehaus.cargo.container.spi.deployer.AbstractCopyingDeployer: Class
org.codehaus.cargo.container.spi.deployer.AbstractCopyingDeployer added
INFO: 8000: org.codehaus.cargo.container.spi.deployer.AbstractDeployer: Class
org.codehaus.cargo.container.spi.deployer.AbstractDeployer added
INFO: 8000: org.codehaus.cargo.container.spi.deployer.AbstractLocalDeployer: Class
org.codehaus.cargo.container.spi.deployer.AbstractLocalDeployer added
INFO: 8000: org.codehaus.cargo.container.spi.deployer.AbstractRemoteDeployer: Class
org.codehaus.cargo.container.spi.deployer.AbstractRemoteDeployer added
INFO: 8000: org.codehaus.cargo.container.spi.deployer.DeployerWatchdog: Class
org.codehaus.cargo.container.spi.deployer.DeployerWatchdog added
INFO: 8000: org.codehaus.cargo.container.spi.util.ContainerUtils: Class
org.codehaus.cargo.container.spi.util.ContainerUtils added
INFO: 8000: org.codehaus.cargo.container.spi.util.DefaultServerRun: Class
org.codehaus.cargo.container.spi.util.DefaultServerRun added
ERROR: 8001: org.codehaus.cargo.container.tomcat.AbstractCatalinaContainer: Class
```

```
   org.codehaus.cargo.container.tomcat.AbstractCatalinaContainer removed
   ERROR: 8001: org.codehaus.cargo.container.tomcat.AbstractTomcatContainer: Class
   org.codehaus.cargo.container.tomcat.AbstractTomcatContainer removed
   INFO: 8000: org.codehaus.cargo.container.tomcat.AbstractTomcatRemoteContainer: Class
   org.codehaus.cargo.container.tomcat.AbstractTomcatRemoteContainer added
   ERROR: 8001: org.codehaus.cargo.container.tomcat.AbstractTomcatStandaloneConfiguration: Class
   org.codehaus.cargo.container.tomcat.AbstractTomcatStandaloneConfiguration removed
   ERROR: 8001: org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration: Class
   org.codehaus.cargo.container.tomcat.CatalinaStandaloneConfiguration removed
   ERROR: 8001: org.codehaus.cargo.container.tomcat.Tomcat3xContainer: Class
   org.codehaus.cargo.container.tomcat.Tomcat3xContainer removed
   INFO: 8000: org.codehaus.cargo.container.tomcat.Tomcat3xLocalContainer: Class
   org.codehaus.cargo.container.tomcat.Tomcat3xLocalContainer added
   INFO: 8000: org.codehaus.cargo.container.tomcat.Tomcat3xStandaloneLocalConfiguration: Class
   org.codehaus.cargo.container.tomcat.Tomcat3xStandaloneLocalConfiguration added
   ERROR: 8001: org.codehaus.cargo.container.tomcat.Tomcat4xContainer: Class
   org.codehaus.cargo.container.tomcat.Tomcat4xContainer removed
   INFO: 8000: org.codehaus.cargo.container.tomcat.Tomcat4xLocalContainer: Class
   org.codehaus.cargo.container.tomcat.Tomcat4xLocalContainer added
   INFO: 8000: org.codehaus.cargo.container.tomcat.Tomcat4xRemoteContainer: Class
   org.codehaus.cargo.container.tomcat.Tomcat4xRemoteContainer added
   INFO: 8000: org.codehaus.cargo.container.tomcat.Tomcat4xStandaloneLocalConfiguration: Class
   org.codehaus.cargo.container.tomcat.Tomcat4xStandaloneLocalConfiguration added
   ERROR: 8001: org.codehaus.cargo.container.tomcat.Tomcat5xContainer: Class
   org.codehaus.cargo.container.tomcat.Tomcat5xContainer removed
   INFO: 8000: org.codehaus.cargo.container.tomcat.Tomcat5xLocalContainer: Class
   org.codehaus.cargo.container.tomcat.Tomcat5xLocalContainer added
   INFO: 8000: org.codehaus.cargo.container.tomcat.Tomcat5xRemoteContainer: Class
   org.codehaus.cargo.container.tomcat.Tomcat5xRemoteContainer added
   INFO: 8000: org.codehaus.cargo.container.tomcat.Tomcat5xStandaloneLocalConfiguration: Class
   org.codehaus.cargo.container.tomcat.Tomcat5xStandaloneLocalConfiguration added
   INFO: 8000: org.codehaus.cargo.container.tomcat.TomcatCopyingLocalDeployer: Class
   org.codehaus.cargo.container.tomcat.TomcatCopyingLocalDeployer added
   INFO: 8000: org.codehaus.cargo.container.tomcat.TomcatExistingLocalConfiguration: Class
   org.codehaus.cargo.container.tomcat.TomcatExistingLocalConfiguration added
   INFO: 8000: org.codehaus.cargo.container.tomcat.TomcatLocalDeployer: Class
   org.codehaus.cargo.container.tomcat.TomcatLocalDeployer added
   INFO: 6000: org.codehaus.cargo.container.tomcat.TomcatPropertySet: Added public field
   MANAGER_PASSWORD
   INFO: 6000: org.codehaus.cargo.container.tomcat.TomcatPropertySet: Added public field
   MANAGER_URL
   INFO: 6000: org.codehaus.cargo.container.tomcat.TomcatPropertySet: Added public field
   MANAGER_USERNAME
   INFO: 8000: org.codehaus.cargo.container.tomcat.TomcatRemoteDeployer: Class
   org.codehaus.cargo.container.tomcat.TomcatRemoteDeployer added
   INFO: 8000: org.codehaus.cargo.container.tomcat.TomcatRuntimeConfiguration: Class
   org.codehaus.cargo.container.tomcat.TomcatRuntimeConfiguration added
   ERROR: 8001: org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration: Class
   org.codehaus.cargo.container.tomcat.TomcatStandaloneConfiguration removed
   INFO: 8000: org.codehaus.cargo.container.tomcat.TomcatWAR: Class
   org.codehaus.cargo.container.tomcat.TomcatWAR added
   INFO: 8000: org.codehaus.cargo.container.tomcat.internal.AbstractCatalinaLocalContainer: Class
   org.codehaus.cargo.container.tomcat.internal.AbstractCatalinaLocalContainer added
   INFO: 8000:
   org.codehaus.cargo.container.tomcat.internal.AbstractCatalinaStandaloneLocalConfiguration:
   Class org.codehaus.cargo.container.tomcat.internal.AbstractCatalinaStandaloneLocalConfiguration
   added
   INFO: 8000: org.codehaus.cargo.container.tomcat.internal.AbstractTomcatDeployer: Class
   org.codehaus.cargo.container.tomcat.internal.AbstractTomcatDeployer added
   INFO: 8000: org.codehaus.cargo.container.tomcat.internal.AbstractTomcatLocalContainer: Class
   org.codehaus.cargo.container.tomcat.internal.AbstractTomcatLocalContainer added
   INFO: 8000:
   org.codehaus.cargo.container.tomcat.internal.AbstractTomcatStandaloneLocalConfiguration: Class
   org.codehaus.cargo.container.tomcat.internal.AbstractTomcatStandaloneLocalConfiguration added
   INFO: 8000: org.codehaus.cargo.container.tomcat.internal.TomcatManager: Class
   org.codehaus.cargo.container.tomcat.internal.TomcatManager added
   INFO: 8000: org.codehaus.cargo.container.tomcat.internal.TomcatManagerException: Class
   org.codehaus.cargo.container.tomcat.internal.TomcatManagerException added
   INFO: 8000: org.codehaus.cargo.container.tomcat.internal.TomcatRuntimeConfigurationCapability:
   Class org.codehaus.cargo.container.tomcat.internal.TomcatRuntimeConfigurationCapability added
   INFO: 8000:
   org.codehaus.cargo.container.tomcat.internal.TomcatStandaloneLocalConfigurationCapability:
   Class org.codehaus.cargo.container.tomcat.internal.TomcatStandaloneLocalConfigurationCapability
   added
   ERROR: 8001: org.codehaus.cargo.container.weblogic.AbstractWebLogicContainer: Class
   org.codehaus.cargo.container.weblogic.AbstractWebLogicContainer removed
   ERROR: 8001: org.codehaus.cargo.container.weblogic.WebLogic8xContainer: Class
   org.codehaus.cargo.container.weblogic.WebLogic8xContainer removed
```

```
INFO: 8000: org.codehaus.cargo.container.weblogic.WebLogic8xLocalContainer: Class
org.codehaus.cargo.container.weblogic.WebLogic8xLocalContainer added
INFO: 8000: org.codehaus.cargo.container.weblogic.WebLogicExistingLocalConfiguration: Class
org.codehaus.cargo.container.weblogic.WebLogicExistingLocalConfiguration added
INFO: 8000: org.codehaus.cargo.container.weblogic.WebLogicPropertySet: Class
org.codehaus.cargo.container.weblogic.WebLogicPropertySet added
ERROR: 8001: org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration: Class
org.codehaus.cargo.container.weblogic.WebLogicStandaloneConfiguration removed
INFO: 8000: org.codehaus.cargo.container.weblogic.WebLogicStandaloneLocalConfiguration: Class
org.codehaus.cargo.container.weblogic.WebLogicStandaloneLocalConfiguration added
INFO: 8000: org.codehaus.cargo.container.weblogic.internal.AbstractWebLogicLocalContainer:
Class org.codehaus.cargo.container.weblogic.internal.AbstractWebLogicLocalContainer added
INFO: 8000:
org.codehaus.cargo.container.weblogic.internal.WebLogicExistingLocalConfigurationCapability:
Class
org.codehaus.cargo.container.weblogic.internal.WebLogicExistingLocalConfigurationCapability
added
INFO: 8000:
org.codehaus.cargo.container.weblogic.internal.WebLogicStandaloneLocalConfigurationCapability:
Class
org.codehaus.cargo.container.weblogic.internal.WebLogicStandaloneLocalConfigurationCapability
added
ERROR: 7005: org.codehaus.cargo.generic.ContainerFactory: Parameter 2 of 'public
org.codehaus.cargo.container.Container createContainer(java.lang.String,
org.codehaus.cargo.generic.configuration.ConfigurationType, java.io.File)' has changed its type
to org.codehaus.cargo.container.ContainerType
ERROR: 7005: org.codehaus.cargo.generic.ContainerFactory: Parameter 3 of 'public
org.codehaus.cargo.container.Container createContainer(java.lang.String,
org.codehaus.cargo.generic.configuration.ConfigurationType, java.io.File)' has changed its type
to org.codehaus.cargo.container.configuration.Configuration
ERROR: 7002: org.codehaus.cargo.generic.ContainerFactory: Method 'public
org.codehaus.cargo.container.Container createContainer(java.lang.String,
org.codehaus.cargo.generic.configuration.ConfigurationType)' has been removed
ERROR: 7002: org.codehaus.cargo.generic.ContainerFactory: Method 'public
org.codehaus.cargo.container.Container createContainer(java.lang.String)' has been removed
ERROR: 7012: org.codehaus.cargo.generic.ContainerFactory: Method 'public boolean
isContainerRegistered(java.lang.String, org.codehaus.cargo.container.ContainerType)' has been
added to an interface
ERROR: 7004: org.codehaus.cargo.generic.ContainerFactory: In method 'public void
registerContainer(java.lang.String, java.lang.Class)' the number of arguments has changed
INFO: 4000: org.codehaus.cargo.generic.DefaultContainerFactory: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
INFO: 5000: org.codehaus.cargo.generic.DefaultContainerFactory: Added
org.codehaus.cargo.generic.spi.AbstractGenericHintFactory to the list of superclasses
INFO: 5000: org.codehaus.cargo.generic.DefaultContainerFactory: Added
org.codehaus.cargo.generic.spi.AbstractIntrospectionGenericHintFactory to the list of
superclasses
INFO: 5000: org.codehaus.cargo.generic.DefaultContainerFactory: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 7005: org.codehaus.cargo.generic.DefaultContainerFactory: Parameter 2 of 'public
org.codehaus.cargo.container.Container createContainer(java.lang.String,
org.codehaus.cargo.generic.configuration.ConfigurationType, java.io.File)' has changed its type
to org.codehaus.cargo.container.ContainerType
ERROR: 7005: org.codehaus.cargo.generic.DefaultContainerFactory: Parameter 3 of 'public
org.codehaus.cargo.container.Container createContainer(java.lang.String,
org.codehaus.cargo.generic.configuration.ConfigurationType, java.io.File)' has changed its type
to org.codehaus.cargo.container.configuration.Configuration
ERROR: 7002: org.codehaus.cargo.generic.DefaultContainerFactory: Method 'public
org.codehaus.cargo.container.Container createContainer(java.lang.String,
org.codehaus.cargo.generic.configuration.ConfigurationType)' has been removed
ERROR: 7002: org.codehaus.cargo.generic.DefaultContainerFactory: Method 'public
org.codehaus.cargo.container.Container createContainer(java.lang.String)' has been removed
INFO: 7011: org.codehaus.cargo.generic.DefaultContainerFactory: Method 'protected
java.lang.Object createInstance(java.lang.reflect.Constructor, java.lang.String,
org.codehaus.cargo.generic.spi.AbstractGenericHintFactory$GenericParameters)' has been added
INFO: 7011: org.codehaus.cargo.generic.DefaultContainerFactory: Method 'protected
java.lang.reflect.Constructor getConstructor(java.lang.Class, java.lang.String,
org.codehaus.cargo.generic.spi.AbstractGenericHintFactory$GenericParameters)' has been added
INFO: 7011: org.codehaus.cargo.generic.DefaultContainerFactory: Method 'public boolean
isContainerRegistered(java.lang.String, org.codehaus.cargo.container.ContainerType)' has been
added
ERROR: 7004: org.codehaus.cargo.generic.DefaultContainerFactory: In method 'public void
registerContainer(java.lang.String, java.lang.Class)' the number of arguments has changed
INFO: 7011: org.codehaus.cargo.generic.DefaultContainerFactory: Method 'public void
registerContainer(java.lang.String, java.lang.String,
org.codehaus.cargo.container.ContainerType)' has been added
ERROR: 7002: org.codehaus.cargo.generic.DefaultContainerFactory: Method 'public void
setConfigurationFactory(org.codehaus.cargo.generic.configuration.ConfigurationFactory)' has
```

```
been removed
ERROR: 7005: org.codehaus.cargo.generic.configuration.ConfigurationFactory: Parameter 2 of
'public org.codehaus.cargo.container.configuration.Configuration
createConfiguration(java.lang.String,
org.codehaus.cargo.generic.configuration.ConfigurationType)' has changed its type to
org.codehaus.cargo.container.configuration.ConfigurationType
ERROR: 7005: org.codehaus.cargo.generic.configuration.ConfigurationFactory: Parameter 2 of
'public org.codehaus.cargo.container.configuration.Configuration
createConfiguration(java.lang.String,
org.codehaus.cargo.generic.configuration.ConfigurationType, java.io.File)' has changed its type
to org.codehaus.cargo.container.configuration.ConfigurationType
ERROR: 7005: org.codehaus.cargo.generic.configuration.ConfigurationFactory: Parameter 2 of
'public boolean isConfigurationRegistered(java.lang.String,
org.codehaus.cargo.generic.configuration.ConfigurationType)' has changed its type to
org.codehaus.cargo.container.configuration.ConfigurationType
ERROR: 7005: org.codehaus.cargo.generic.configuration.ConfigurationFactory: Parameter 2 of
'public void registerConfiguration(java.lang.String,
org.codehaus.cargo.generic.configuration.ConfigurationType, java.lang.Class)' has changed its
type to org.codehaus.cargo.container.configuration.ConfigurationType
ERROR: 8001: org.codehaus.cargo.generic.configuration.ConfigurationType: Class
org.codehaus.cargo.generic.configuration.ConfigurationType removed
INFO: 4000: org.codehaus.cargo.generic.configuration.DefaultConfigurationFactory: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
INFO: 5000: org.codehaus.cargo.generic.configuration.DefaultConfigurationFactory: Added
org.codehaus.cargo.generic.spi.AbstractIntrospectionGenericHintFactory to the list of
superclasses
INFO: 5000: org.codehaus.cargo.generic.configuration.DefaultConfigurationFactory: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 7005: org.codehaus.cargo.generic.configuration.DefaultConfigurationFactory: Parameter 2
of 'public org.codehaus.cargo.container.configuration.Configuration
createConfiguration(java.lang.String,
org.codehaus.cargo.generic.configuration.ConfigurationType)' has changed its type to
org.codehaus.cargo.container.configuration.ConfigurationType
ERROR: 7005: org.codehaus.cargo.generic.configuration.DefaultConfigurationFactory: Parameter 2
of 'public org.codehaus.cargo.container.configuration.Configuration
createConfiguration(java.lang.String,
org.codehaus.cargo.generic.configuration.ConfigurationType, java.io.File)' has changed its type
to org.codehaus.cargo.container.configuration.ConfigurationType
ERROR: 7004: org.codehaus.cargo.generic.configuration.DefaultConfigurationFactory: In method
'protected java.lang.Object createInstance(java.lang.reflect.Constructor,
org.codehaus.cargo.generic.spi.AbstractGenericHintFactory$GenericParameters)' the number of
arguments has changed
ERROR: 7004: org.codehaus.cargo.generic.configuration.DefaultConfigurationFactory: In method
'protected java.lang.reflect.Constructor getConstructor(java.lang.Class,
org.codehaus.cargo.generic.spi.AbstractGenericHintFactory$GenericParameters)' the number of
arguments has changed
ERROR: 7005: org.codehaus.cargo.generic.configuration.DefaultConfigurationFactory: Parameter 2
of 'public boolean isConfigurationRegistered(java.lang.String,
org.codehaus.cargo.generic.configuration.ConfigurationType)' has changed its type to
org.codehaus.cargo.container.configuration.ConfigurationType
ERROR: 7005: org.codehaus.cargo.generic.configuration.DefaultConfigurationFactory: Parameter 2
of 'public void registerConfiguration(java.lang.String,
org.codehaus.cargo.generic.configuration.ConfigurationType, java.lang.Class)' has changed its
type to org.codehaus.cargo.container.configuration.ConfigurationType
INFO: 7011: org.codehaus.cargo.generic.configuration.DefaultConfigurationFactory: Method
'public void registerConfiguration(java.lang.String, java.lang.String,
org.codehaus.cargo.container.configuration.ConfigurationType)' has been added
INFO: 4000: org.codehaus.cargo.generic.deployable.DefaultDeployableFactory: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
INFO: 5000: org.codehaus.cargo.generic.deployable.DefaultDeployableFactory: Added
org.codehaus.cargo.generic.spi.AbstractGenericHintFactory to the list of superclasses
INFO: 5000: org.codehaus.cargo.generic.deployable.DefaultDeployableFactory: Added
org.codehaus.cargo.generic.spi.AbstractIntrospectionGenericHintFactory to the list of
superclasses
INFO: 5000: org.codehaus.cargo.generic.deployable.DefaultDeployableFactory: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 7005: org.codehaus.cargo.generic.deployable.DefaultDeployableFactory: Parameter 3 of
'public org.codehaus.cargo.container.deployable.Deployable createDeployable(java.lang.String,
java.lang.String, org.codehaus.cargo.generic.deployable.DeployableType)' has changed its type
to org.codehaus.cargo.container.deployable.DeployableType
INFO: 7011: org.codehaus.cargo.generic.deployable.DefaultDeployableFactory: Method 'protected
java.lang.Object createInstance(java.lang.reflect.Constructor, java.lang.String,
org.codehaus.cargo.generic.spi.AbstractGenericHintFactory$GenericParameters)' has been added
INFO: 7011: org.codehaus.cargo.generic.deployable.DefaultDeployableFactory: Method 'protected
java.lang.reflect.Constructor getConstructor(java.lang.Class, java.lang.String,
org.codehaus.cargo.generic.spi.AbstractGenericHintFactory$GenericParameters)' has been added
INFO: 7011: org.codehaus.cargo.generic.deployable.DefaultDeployableFactory: Method 'public
boolean isDeployableRegistered(java.lang.String,
```

```
org.codehaus.cargo.container.deployable.DeployableType)' has been added
INFO: 7011: org.codehaus.cargo.generic.deployable.DefaultDeployableFactory: Method 'public void
registerDeployable(java.lang.String, org.codehaus.cargo.container.deployable.DeployableType,
java.lang.Class)' has been added
INFO: 7011: org.codehaus.cargo.generic.deployable.DefaultDeployableFactory: Method 'public void
registerDeployable(java.lang.String, java.lang.String,
org.codehaus.cargo.container.deployable.DeployableType)' has been added
ERROR: 7002: org.codehaus.cargo.generic.deployable.DefaultDeployableFactory: Method 'public
void registerImplementation(java.lang.String, java.util.Map)' has been removed
ERROR: 7005: org.codehaus.cargo.generic.deployable.DeployableFactory: Parameter 3 of 'public
org.codehaus.cargo.container.deployable.Deployable createDeployable(java.lang.String,
java.lang.String, org.codehaus.cargo.generic.deployable.DeployableType)' has changed its type
to org.codehaus.cargo.container.deployable.DeployableType
ERROR: 7012: org.codehaus.cargo.generic.deployable.DeployableFactory: Method 'public boolean
isDeployableRegistered(java.lang.String,
org.codehaus.cargo.container.deployable.DeployableType)' has been added to an interface
ERROR: 7012: org.codehaus.cargo.generic.deployable.DeployableFactory: Method 'public void
registerDeployable(java.lang.String, org.codehaus.cargo.container.deployable.DeployableType,
java.lang.Class)' has been added to an interface
ERROR: 8001: org.codehaus.cargo.generic.deployable.DeployableType: Class
org.codehaus.cargo.generic.deployable.DeployableType removed
INFO: 4000: org.codehaus.cargo.generic.deployer.DefaultDeployerFactory: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
INFO: 5000: org.codehaus.cargo.generic.deployer.DefaultDeployerFactory: Added
org.codehaus.cargo.generic.spi.AbstractIntrospectionGenericHintFactory to the list of
superclasses
INFO: 5000: org.codehaus.cargo.generic.deployer.DefaultDeployerFactory: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 7005: org.codehaus.cargo.generic.deployer.DefaultDeployerFactory: Parameter 2 of 'public
org.codehaus.cargo.container.deployer.Deployer
createDeployer(org.codehaus.cargo.container.Container, java.lang.String)' has changed its type
to org.codehaus.cargo.container.deployer.DeployerType
ERROR: 7004: org.codehaus.cargo.generic.deployer.DefaultDeployerFactory: In method 'protected
java.lang.Object createInstance(java.lang.reflect.Constructor,
org.codehaus.cargo.generic.spi.AbstractGenericHintFactory$GenericParameters)' the number of
arguments has changed
ERROR: 7004: org.codehaus.cargo.generic.deployer.DefaultDeployerFactory: In method 'protected
java.lang.reflect.Constructor getConstructor(java.lang.Class,
org.codehaus.cargo.generic.spi.AbstractGenericHintFactory$GenericParameters)' the number of
arguments has changed
INFO: 7011: org.codehaus.cargo.generic.deployer.DefaultDeployerFactory: Method 'public boolean
isDeployerRegistered(java.lang.String, org.codehaus.cargo.container.deployer.DeployerType)' has
been added
ERROR: 7005: org.codehaus.cargo.generic.deployer.DefaultDeployerFactory: Parameter 2 of 'public
void registerDeployer(java.lang.String, java.lang.String, java.lang.Class)' has changed its
type to org.codehaus.cargo.container.deployer.DeployerType
INFO: 7011: org.codehaus.cargo.generic.deployer.DefaultDeployerFactory: Method 'public void
registerDeployer(java.lang.String, java.lang.String,
org.codehaus.cargo.container.deployer.DeployerType)' has been added
ERROR: 6011: org.codehaus.cargo.generic.deployer.DeployerFactory: Field DEFAULT has been
removed, but it was previously a constant
ERROR: 7005: org.codehaus.cargo.generic.deployer.DeployerFactory: Parameter 2 of 'public
org.codehaus.cargo.container.deployer.Deployer
createDeployer(org.codehaus.cargo.container.Container, java.lang.String)' has changed its type
to org.codehaus.cargo.container.deployer.DeployerType
ERROR: 7012: org.codehaus.cargo.generic.deployer.DeployerFactory: Method 'public boolean
isDeployerRegistered(java.lang.String, org.codehaus.cargo.container.deployer.DeployerType)' has
been added to an interface
ERROR: 7005: org.codehaus.cargo.generic.deployer.DeployerFactory: Parameter 2 of 'public void
registerDeployer(java.lang.String, java.lang.String, java.lang.Class)' has changed its type to
org.codehaus.cargo.container.deployer.DeployerType
INFO: 4000: org.codehaus.cargo.generic.spi.AbstractGenericHintFactory: Added
org.codehaus.cargo.util.monitor.Monitorable to the set of implemented interfaces
INFO: 5000: org.codehaus.cargo.generic.spi.AbstractGenericHintFactory: Added
org.codehaus.cargo.util.monitor.MonitoredObject to the list of superclasses
ERROR: 7004: org.codehaus.cargo.generic.spi.AbstractGenericHintFactory: In method 'protected
java.lang.Object createInstance(java.lang.reflect.Constructor,
org.codehaus.cargo.generic.spi.AbstractGenericHintFactory$GenericParameters)' the number of
arguments has changed
ERROR: 7004: org.codehaus.cargo.generic.spi.AbstractGenericHintFactory: In method 'protected
java.lang.reflect.Constructor getConstructor(java.lang.Class,
org.codehaus.cargo.generic.spi.AbstractGenericHintFactory$GenericParameters)' the number of
arguments has changed
INFO: 8000: org.codehaus.cargo.generic.spi.AbstractIntrospectionGenericHintFactory: Class
org.codehaus.cargo.generic.spi.AbstractIntrospectionGenericHintFactory added
INFO: 8000: org.codehaus.cargo.module.AbstractDescriptor: Class
org.codehaus.cargo.module.AbstractDescriptor added
INFO: 8000: org.codehaus.cargo.module.AbstractDescriptorIo: Class
```

```
   org.codehaus.cargo.module.AbstractDescriptorIo added
   INFO: 8000: org.codehaus.cargo.module.AbstractDescriptorTag: Class
   org.codehaus.cargo.module.AbstractDescriptorTag added
   INFO: 8000: org.codehaus.cargo.module.J2eeDescriptor: Class
   org.codehaus.cargo.module.J2eeDescriptor added
   ERROR: 7012: org.codehaus.cargo.module.application.ApplicationXml: Method 'public void
   addEjbModule(java.lang.String)' has been added to an interface
   INFO: 5000: org.codehaus.cargo.module.application.ApplicationXmlTag: Added
   org.codehaus.cargo.module.AbstractDescriptorTag to the list of superclasses
   ERROR: 5001: org.codehaus.cargo.module.application.ApplicationXmlTag: Removed
   org.codehaus.cargo.module.webapp.AbstractDescriptorTag from the list of superclasses
   INFO: 5000: org.codehaus.cargo.module.application.DefaultApplicationXml: Added
   org.codehaus.cargo.module.AbstractDescriptor to the list of superclasses
   ERROR: 5001: org.codehaus.cargo.module.application.DefaultApplicationXml: Removed
   org.codehaus.cargo.module.webapp.AbstractDescriptor from the list of superclasses
   INFO: 7011: org.codehaus.cargo.module.application.DefaultApplicationXml: Method 'public void
   addEjbModule(java.lang.String)' has been added
   INFO: 4000: org.codehaus.cargo.module.ejb.EjbJarXml: Added org.codehaus.cargo.module.Descriptor
   to the set of implemented interfaces
   INFO: 4000: org.codehaus.cargo.module.ejb.EjbJarXml: Added
   org.codehaus.cargo.module.J2eeDescriptor to the set of implemented interfaces
   INFO: 5000: org.codehaus.cargo.module.ejb.EjbJarXml: Added
   org.codehaus.cargo.module.AbstractDescriptor to the list of superclasses
   ERROR: 5001: org.codehaus.cargo.module.ejb.EjbJarXml: Removed
   org.codehaus.cargo.module.webapp.AbstractDescriptor from the list of superclasses
   INFO: 7011: org.codehaus.cargo.module.ejb.EjbJarXml: Method 'public java.lang.String
   getFileName()' has been added
   ERROR: 7006: org.codehaus.cargo.module.ejb.EjbJarXml: Return type of method 'public
   org.codehaus.cargo.module.ejb.VendorEjbDescriptor getVendorDescriptor()' has been changed to
   org.codehaus.cargo.module.Descriptor
   INFO: 5000: org.codehaus.cargo.module.ejb.EjbJarXmlIo: Added
   org.codehaus.cargo.module.AbstractDescriptorIo to the list of superclasses
   ERROR: 5001: org.codehaus.cargo.module.ejb.EjbJarXmlIo: Removed
   org.codehaus.cargo.module.webapp.AbstractDescriptorIo from the list of superclasses
   INFO: 5000: org.codehaus.cargo.module.ejb.EjbJarXmlTag: Added
   org.codehaus.cargo.module.AbstractDescriptorTag to the list of superclasses
   ERROR: 5001: org.codehaus.cargo.module.ejb.EjbJarXmlTag: Removed
   org.codehaus.cargo.module.webapp.AbstractDescriptorTag from the list of superclasses
   INFO: 4000: org.codehaus.cargo.module.ejb.VendorEjbDescriptor: Added
   org.codehaus.cargo.module.Descriptor to the set of implemented interfaces
   INFO: 4000: org.codehaus.cargo.module.ejb.orion.OrionEjbJarXml: Added
   org.codehaus.cargo.module.Descriptor to the set of implemented interfaces
   INFO: 5000: org.codehaus.cargo.module.ejb.orion.OrionEjbJarXml: Added
   org.codehaus.cargo.module.AbstractDescriptor to the list of superclasses
   ERROR: 5001: org.codehaus.cargo.module.ejb.orion.OrionEjbJarXml: Removed
   org.codehaus.cargo.module.webapp.AbstractDescriptor from the list of superclasses
   INFO: 7011: org.codehaus.cargo.module.ejb.orion.OrionEjbJarXml: Method 'public java.lang.String
   getFileName()' has been added
   INFO: 5000: org.codehaus.cargo.module.ejb.orion.OrionEjbJarXmlIo: Added
   org.codehaus.cargo.module.AbstractDescriptorIo to the list of superclasses
   ERROR: 5001: org.codehaus.cargo.module.ejb.orion.OrionEjbJarXmlIo: Removed
   org.codehaus.cargo.module.webapp.AbstractDescriptorIo from the list of superclasses
   INFO: 4000: org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXml: Added
   org.codehaus.cargo.module.Descriptor to the set of implemented interfaces
   INFO: 5000: org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXml: Added
   org.codehaus.cargo.module.AbstractDescriptor to the list of superclasses
   ERROR: 5001: org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXml: Removed
   org.codehaus.cargo.module.webapp.AbstractDescriptor from the list of superclasses
   INFO: 7011: org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXml: Method 'public void
   addDispatchPolicy(org.codehaus.cargo.module.ejb.EjbDef, java.lang.String)' has been added
   INFO: 7011: org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXml: Method 'public
   java.lang.String getDispatchPolicy(org.codehaus.cargo.module.ejb.EjbDef)' has been added
   INFO: 7011: org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXml: Method 'public
   java.lang.String getFileName()' has been added
   INFO: 5000: org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXmlIo: Added
   org.codehaus.cargo.module.AbstractDescriptorIo to the list of superclasses
   ERROR: 5001: org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXmlIo: Removed
   org.codehaus.cargo.module.webapp.AbstractDescriptorIo from the list of superclasses
   INFO: 5000: org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXmlTag: Added
   org.codehaus.cargo.module.AbstractDescriptorTag to the list of superclasses
   ERROR: 5001: org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXmlTag: Removed
   org.codehaus.cargo.module.webapp.AbstractDescriptorTag from the list of superclasses
   INFO: 6000: org.codehaus.cargo.module.ejb.weblogic.WeblogicEjbJarXmlTag: Added public field
   DISPATCH_POLICY
   INFO: 4000: org.codehaus.cargo.module.ejb.websphere.IbmEjbJarBndXmi: Added
   org.codehaus.cargo.module.Descriptor to the set of implemented interfaces
   INFO: 5000: org.codehaus.cargo.module.ejb.websphere.IbmEjbJarBndXmi: Added
   org.codehaus.cargo.module.AbstractDescriptor to the list of superclasses
```

```
ERROR: 5001: org.codehaus.cargo.module.ejb.websphere.IbmEjbJarBndXmi: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptor from the list of superclasses
INFO: 7011: org.codehaus.cargo.module.ejb.websphere.IbmEjbJarBndXmi: Method 'public
java.lang.String getFileName()' has been added
INFO: 5000: org.codehaus.cargo.module.ejb.websphere.IbmEjbJarBndXmiIo: Added
org.codehaus.cargo.module.AbstractDescriptorIo to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.ejb.websphere.IbmEjbJarBndXmiIo: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptorIo from the list of superclasses
INFO: 8000: org.codehaus.cargo.module.internal.util.xml.AbstractElement: Class
org.codehaus.cargo.module.internal.util.xml.AbstractElement added
INFO: 8000: org.codehaus.cargo.module.internal.util.xml.AbstractNode: Class
org.codehaus.cargo.module.internal.util.xml.AbstractNode added
INFO: 8000: org.codehaus.cargo.module.internal.util.xml.AbstractNodeList: Class
org.codehaus.cargo.module.internal.util.xml.AbstractNodeList added
INFO: 8000: org.codehaus.cargo.module.merge.AbstractMergeSet: Class
org.codehaus.cargo.module.merge.AbstractMergeSet added
INFO: 8000: org.codehaus.cargo.module.merge.MergeElement: Class
org.codehaus.cargo.module.merge.MergeElement added
INFO: 8000: org.codehaus.cargo.module.merge.MergeException: Class
org.codehaus.cargo.module.merge.MergeException added
INFO: 8000: org.codehaus.cargo.module.merge.MergeNodeList: Class
org.codehaus.cargo.module.merge.MergeNodeList added
INFO: 8000: org.codehaus.cargo.module.merge.MergePair: Class
org.codehaus.cargo.module.merge.MergePair added
INFO: 8000: org.codehaus.cargo.module.merge.MergeProcessor: Class
org.codehaus.cargo.module.merge.MergeProcessor added
INFO: 8000: org.codehaus.cargo.module.merge.MergeStrategy: Class
org.codehaus.cargo.module.merge.MergeStrategy added
ERROR: 8001: org.codehaus.cargo.module.webapp.AbstractDescriptor: Class
org.codehaus.cargo.module.webapp.AbstractDescriptor removed
ERROR: 8001: org.codehaus.cargo.module.webapp.AbstractDescriptorIo: Class
org.codehaus.cargo.module.webapp.AbstractDescriptorIo removed
ERROR: 8001: org.codehaus.cargo.module.webapp.AbstractDescriptorTag: Class
org.codehaus.cargo.module.webapp.AbstractDescriptorTag removed
INFO: 4000: org.codehaus.cargo.module.webapp.WebXml: Added org.codehaus.cargo.module.Descriptor
to the set of implemented interfaces
INFO: 4000: org.codehaus.cargo.module.webapp.WebXml: Added
org.codehaus.cargo.module.J2eeDescriptor to the set of implemented interfaces
INFO: 5000: org.codehaus.cargo.module.webapp.WebXml: Added
org.codehaus.cargo.module.AbstractDescriptor to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.WebXml: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptor from the list of superclasses
ERROR: 7005: org.codehaus.cargo.module.webapp.WebXml: Parameter 1 of 'public void
addRootElement(org.codehaus.cargo.module.webapp.AbstractDescriptorTag, org.w3c.dom.Element)'
has changed its type to org.codehaus.cargo.module.AbstractDescriptorTag
INFO: 7011: org.codehaus.cargo.module.webapp.WebXml: Method 'public java.lang.String
getFileName()' has been added
ERROR: 7006: org.codehaus.cargo.module.webapp.WebXml: Return type of method 'public
org.codehaus.cargo.module.webapp.VendorWebAppDescriptor getVendorDescriptor()' has been changed
to org.codehaus.cargo.module.Descriptor
ERROR: 7005: org.codehaus.cargo.module.webapp.WebXml: Parameter 1 of 'public void
replaceRootElement(org.codehaus.cargo.module.webapp.AbstractDescriptorTag,
org.w3c.dom.Element)' has changed its type to org.codehaus.cargo.module.AbstractDescriptorTag
INFO: 5000: org.codehaus.cargo.module.webapp.WebXmlIo: Added
org.codehaus.cargo.module.AbstractDescriptorIo to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.WebXmlIo: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptorIo from the list of superclasses
ERROR: 7002: org.codehaus.cargo.module.webapp.WebXmlIo: Method 'public java.io.File[]
writeAll(org.codehaus.cargo.module.webapp.WebXml, java.io.File)' has been removed
ERROR: 7002: org.codehaus.cargo.module.webapp.WebXmlIo: Method 'public void
writeWebXml(org.codehaus.cargo.module.webapp.WebXml, java.io.File)' has been removed
ERROR: 7002: org.codehaus.cargo.module.webapp.WebXmlIo: Method 'public void
writeWebXml(org.codehaus.cargo.module.webapp.WebXml, java.io.File, java.lang.String)' has been
removed
ERROR: 7002: org.codehaus.cargo.module.webapp.WebXmlIo: Method 'public void
writeWebXml(org.codehaus.cargo.module.webapp.WebXml, java.io.File, java.lang.String, boolean)'
has been removed
ERROR: 7002: org.codehaus.cargo.module.webapp.WebXmlIo: Method 'public void
writeWebXml(org.codehaus.cargo.module.webapp.WebXml, java.io.OutputStream, java.lang.String,
boolean)' has been removed
INFO: 5000: org.codehaus.cargo.module.webapp.WebXmlTag: Added
org.codehaus.cargo.module.AbstractDescriptorTag to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.WebXmlTag: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptorTag from the list of superclasses
INFO: 5000: org.codehaus.cargo.module.webapp.jboss.JBossWebXml: Added
org.codehaus.cargo.module.AbstractDescriptor to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.jboss.JBossWebXml: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptor from the list of superclasses
```

```
INFO: 5000: org.codehaus.cargo.module.webapp.jboss.JBossWebXmlIo: Added
org.codehaus.cargo.module.AbstractDescriptorIo to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.jboss.JBossWebXmlIo: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptorIo from the list of superclasses
INFO: 5000: org.codehaus.cargo.module.webapp.jboss.JBossWebXmlTag: Added
org.codehaus.cargo.module.AbstractDescriptorTag to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.jboss.JBossWebXmlTag: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptorTag from the list of superclasses
INFO: 5000: org.codehaus.cargo.module.webapp.orion.OrionWebXml: Added
org.codehaus.cargo.module.AbstractDescriptor to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.orion.OrionWebXml: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptor from the list of superclasses
INFO: 5000: org.codehaus.cargo.module.webapp.orion.OrionWebXmlIo: Added
org.codehaus.cargo.module.AbstractDescriptorIo to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.orion.OrionWebXmlIo: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptorIo from the list of superclasses
INFO: 5000: org.codehaus.cargo.module.webapp.tomcat.TomcatContextXml: Added
org.codehaus.cargo.module.AbstractDescriptor to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.tomcat.TomcatContextXml: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptor from the list of superclasses
INFO: 5000: org.codehaus.cargo.module.webapp.tomcat.TomcatContextXmlIo: Added
org.codehaus.cargo.module.AbstractDescriptorIo to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.tomcat.TomcatContextXmlIo: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptorIo from the list of superclasses
INFO: 5000: org.codehaus.cargo.module.webapp.tomcat.TomcatContextXmlTag: Added
org.codehaus.cargo.module.AbstractDescriptorTag to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.tomcat.TomcatContextXmlTag: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptorTag from the list of superclasses
INFO: 5000: org.codehaus.cargo.module.webapp.weblogic.WeblogicXml: Added
org.codehaus.cargo.module.AbstractDescriptor to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.weblogic.WeblogicXml: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptor from the list of superclasses
INFO: 5000: org.codehaus.cargo.module.webapp.weblogic.WeblogicXmlIo: Added
org.codehaus.cargo.module.AbstractDescriptorIo to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.weblogic.WeblogicXmlIo: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptorIo from the list of superclasses
INFO: 5000: org.codehaus.cargo.module.webapp.weblogic.WeblogicXmlTag: Added
org.codehaus.cargo.module.AbstractDescriptorTag to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.weblogic.WeblogicXmlTag: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptorTag from the list of superclasses
INFO: 5000: org.codehaus.cargo.module.webapp.websphere.IbmWebBndXmi: Added
org.codehaus.cargo.module.AbstractDescriptor to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.websphere.IbmWebBndXmi: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptor from the list of superclasses
INFO: 5000: org.codehaus.cargo.module.webapp.websphere.IbmWebBndXmiIo: Added
org.codehaus.cargo.module.AbstractDescriptorIo to the list of superclasses
ERROR: 5001: org.codehaus.cargo.module.webapp.websphere.IbmWebBndXmiIo: Removed
org.codehaus.cargo.module.webapp.AbstractDescriptorIo from the list of superclasses
INFO: 8000: org.codehaus.cargo.util.Base64: Class org.codehaus.cargo.util.Base64 added
INFO: 7011: org.codehaus.cargo.util.FileUtils: Method 'public java.io.File
createDirectory(java.net.URI, java.lang.String)' has been added
```

Changes to the Ant API:

```
INFO: 7011: org.codehaus.cargo.ant.CargoTask: Method 'protected java.lang.String getHome()' has
been added
ERROR: 7002: org.codehaus.cargo.ant.CargoTask: Method 'protected java.io.File getHomeDir()' has
been removed
INFO: 7011: org.codehaus.cargo.ant.CargoTask: Method 'public void setHint(java.lang.String)'
has been added
INFO: 7011: org.codehaus.cargo.ant.CargoTask: Method 'public void setHome(java.lang.String)'
has been added
ERROR: 7002: org.codehaus.cargo.ant.CargoTask: Method 'public void setHomeDir(java.io.File)'
has been removed
INFO: 7011: org.codehaus.cargo.ant.CargoTask: Method 'public void
setType(org.codehaus.cargo.container.ContainerType)' has been added
INFO: 7011: org.codehaus.cargo.ant.CargoTask: Method 'protected void setupHome()' has been
added
ERROR: 7002: org.codehaus.cargo.ant.CargoTask: Method 'protected void setupHomeDir()' has been
removed
INFO: 7011: org.codehaus.cargo.ant.ConfigurationElement: Method 'public void
addConfiguredDeployable(org.codehaus.cargo.ant.DeployableElement)' has been added
ERROR: 7002: org.codehaus.cargo.ant.ConfigurationElement: Method 'public void
addConfiguredEar(org.codehaus.cargo.ant.EARElement)' has been removed
```

```
ERROR: 7002: org.codehaus.cargo.ant.ConfigurationElement: Method 'public void
addConfiguredWar(org.codehaus.cargo.ant.WARElement)' has been removed
INFO: 7011: org.codehaus.cargo.ant.ConfigurationElement: Method 'protected java.util.List
getDeployables()' has been added
ERROR: 7002: org.codehaus.cargo.ant.ConfigurationElement: Method 'protected java.util.List
getEars()' has been removed
ERROR: 7006: org.codehaus.cargo.ant.ConfigurationElement: Return type of method 'public
org.codehaus.cargo.generic.configuration.ConfigurationType getType()' has been changed to
org.codehaus.cargo.container.configuration.ConfigurationType
ERROR: 7002: org.codehaus.cargo.ant.ConfigurationElement: Method 'protected java.util.List
getWars()' has been removed
INFO: 8000: org.codehaus.cargo.ant.DeployableElement: Class
org.codehaus.cargo.ant.DeployableElement added
ERROR: 8001: org.codehaus.cargo.ant.EARElement: Class org.codehaus.cargo.ant.EARElement removed
ERROR: 8001: org.codehaus.cargo.ant.WARElement: Class org.codehaus.cargo.ant.WARElement removed
ERROR: 8001: org.codehaus.cargo.ant.jo.Jo1xCargoTask: Class
org.codehaus.cargo.ant.jo.Jo1xCargoTask removed
ERROR: 8001: org.codehaus.cargo.ant.orion.Oc4j9xCargoTask: Class
org.codehaus.cargo.ant.orion.Oc4j9xCargoTask removed
ERROR: 8001: org.codehaus.cargo.ant.orion.Orion1xCargoTask: Class
org.codehaus.cargo.ant.orion.Orion1xCargoTask removed
ERROR: 8001: org.codehaus.cargo.ant.orion.Orion2xCargoTask: Class
org.codehaus.cargo.ant.orion.Orion2xCargoTask removed
ERROR: 8001: org.codehaus.cargo.ant.resin.Resin2xCargoTask: Class
org.codehaus.cargo.ant.resin.Resin2xCargoTask removed
ERROR: 8001: org.codehaus.cargo.ant.resin.Resin3xCargoTask: Class
org.codehaus.cargo.ant.resin.Resin3xCargoTask removed
ERROR: 8001: org.codehaus.cargo.ant.tomcat.Tomcat3xCargoTask: Class
org.codehaus.cargo.ant.tomcat.Tomcat3xCargoTask removed
ERROR: 8001: org.codehaus.cargo.ant.tomcat.Tomcat4xCargoTask: Class
org.codehaus.cargo.ant.tomcat.Tomcat4xCargoTask removed
ERROR: 8001: org.codehaus.cargo.ant.tomcat.Tomcat5xCargoTask: Class
org.codehaus.cargo.ant.tomcat.Tomcat5xCargoTask removed
ERROR: 8001: org.codehaus.cargo.ant.weblogic.WebLogic8xCargoTask: Class
org.codehaus.cargo.ant.weblogic.WebLogic8xCargoTask removed
```

Release notes for IntelliJ IDEA Plugin 0.1

This page last changed on Dec 30, 2005 by vmassol.

Release notes for IntelliJ IDEA Plugin 0.1

This is the initial release of the plugin. It comes bundled with Cargo 0.6 and supports all its containers.

Installation

You may install this plugin by unzipping the zip file into your IntelliJ IDEA plugins folder and restarting the IDE. However, usually you are much better off by using the built-in plugin manager like this:

File -> Settings -> Plugins -> Available -> Right Click on Cargo -> Download and Install Plugin

Requirements

- IntelliJ IDEA 4.5.4

Implemented issues

jira.codehaus.org (3 issues)					
T	Key	Res	Summary	Assignee	Reporter
	CARGO-188	FIXED	org.codehaus.cargo.idea.CargoConfigTest.testSetH fails if no floppy disc is inserted in the disk drive	Hendrik Schreiber	Magnus Grimsell
	CARGO-186	FIXED	Testcases testSetConfigDir and testSetHomeDir fail on windows	Hendrik Schreiber	Arnaud Heritier
	CARGO-179	FIXED	Create Cargo plugin for IntelliJIdea 4.5.4	Hendrik Schreiber	Hendrik Schreiber

Known Issues

- When stopping a container an Exception will be shown in the console. This is a bug in the version of Ant that comes bundled with IntelliJ IDEA 4.5.4.
- This version of the plugin will not work with any other version of IntelliJ IDEA. The next release will be for IntelliJ IDEA 5.x.

This page last changed on Dec 30, 2005 by vmassol.

Mission

Cargo is a thin wrapper around existing containers (e.g. J2EE containers). It provides different APIs to easily manipulate containers.

Cargo provides the following APIs:

- A Java to start/stop/configure Java Containers and deploy modules into them. We also offer Ant tasks, Maven 1, Maven 2, Intellij IDEA and Netbeans plugins.
- A Java API to parse/create/merge J2EE Modules

Status

Version status (click in the status column to get release notes):

Version	Status	Comments
0.1	(/)	Released on 11/09/04
0.2	(/)	Released on 03/10/04
0.3	(/)	Released on 30/10/04
0.4	(/)	Released on 26/11/04
0.5	(/)	Released on 30/04/05
0.6	(/)	Released on 21/07/05
0.7	(/)	Released on 30/12/05

Architecture

(view as slideshow)

High level Cargo architecture

Different ways of using Cargo

Cargo offers differents ways of using it at different levels:

- **Module Java API**: A Java API to parse/create/merge J2EE Modules (WAR, EAR, etc)
- **Container Java API**: A Java API to start/stop/configure Java Containers and deploy modules into them.
- **Generic Java API**: A Java API that sits on top of the Container API but allows writing generic code that works with any container. It consists mostly in a set of Factory classes to instantiate Container API objects by name.
- **Build plugins**
  - **Ant tasks**: A set of Ant tasks that wrap the Generic Java API
  - **Maven 1**: A Maven 1 plugin that wraps the Ant tasks
  - **Maven 2**: A Maven 2 plugin
- **IDE plugins**
  - **Netbeans**
  - **IntelliJ IDEA**

The main Container API objects are:

- The **Container** is the top level interface wrapping a real physical container. Cargo supports local and remote containers. A Container is composed of a Configuration.
- A **Configuration** tells Cargo how the container is to be configured (whether it should create a standalone setup, whether it should be based on an existing configuration, etc). A Configuration can be configured to install Deployables before the Container is started.
- You can use a **Deployer** to deploy Deployables dynamically (i.e. after the Container is started).
- **Deployables** are archives to be deployed in the Container. They are WAR, EAR, etc.

Feature list

Some top-level features (the full feature list can be found here):

- Configuration — A Configuration specifies how the container is configured
- Container — A top level interface wrapping a real physical container
- Debugging — Explain how to perform debugging when something doesn't work in Cargo
- Deployment — How to deploy components to a container
- Extensions — Extensions are additions to the Cargo core Java API such as build tool plugins, IDE plugins, etc
- Module API — API to manipulate J2EE archives, including vendor-specific deployment descriptors

Container support

List of supported containers and the extensions that are implemented for each container (Java API, Ant tasks and Maven plugins). The specified version is the Cargo version where the feature was first made available. Click on a container's name to see a detailed list of features it supports.

Container	Java API(version)	Ant tasks(version)	Maven 1 plugin(version)	Maven 2 plugin(version)
JBoss 3.x	0.7	0.7	0.7	0.7
JBoss 4.x	0.7	0.7	0.7	0.7
Jetty 4.x	0.1	???	???	???

jo! 1.x	0.5	0.5	0.5	0.7
OC4J 9.x	0.3	0.3	0.5	0.7
Orion 1.x	0.1	0.1	0.5	0.7
Orion 2.x	0.1	0.1	0.5	0.7
Resin 2.x	0.1	0.1	0.5	0.7
Resin 3.x	0.1	0.1	0.5	0.7
Tomcat 3.x	0.1	0.1	0.5	0.7
Tomcat 4.x	0.1	0.1	0.5	0.7
Tomcat 5.x	0.1	0.1	0.5	0.7
WebLogic 8.x	0.3	0.3	0.5	0.7

We also encourage you to report success and failures on different versions of those containers in the Tested on section.

Quick Start

The following examples demonstrate how to configure Resin 3.0.15 to start in `target/resin3x` and deploy a WAR located in `path/to/simple.war`. The default port is 8080. Please note that the `container.start()` and `container.stop()` methods wait until the container is fully started and fully stopped before continuing. Thus, for any action you are executing after, you are assured the container is completely operational.

Static deployment

Static deployment means that the Deployabe is deployed before the container is started. Here's an example using the strongly type Java API:

```
Deployable war = new WAR("path/to/simple.war");

Configuration configuration =
    new Resin3xStandaloneLocalConfiguration("target/myresin3x"));
configuration.addDeployable(war);

LocalContainer container = new Resin3xLocalContainer(configuration);
container.setHome("c:/apps/resin-3.0.15");

container.start();
// Here you are assured the container is started.

container.stop();
// Here you are assured the container is stopped.
```

Here's the same example using the generic untyped API:

```
Deployable war = new DefaultDeployableFactory().createDeployable(
    "resin3x", "path/to/simple.war", DeployableType.WAR);
```

```
LocalConfiguration configuration =
    (LocalConfigutation) new DefaultConfigurationFactory(
        "resin3x", ConfigurationType.STANDALONE);
configuration.addDeployable(war);

LocalContainer container =
    (LocalContainer) new DefaultContainerFactory().createContainer(
        "resin3x", configuration);
container.setHome("c:/apps/resin-3.0.15");

container.start();
// Here you are assured the container is started.

container.stop();
// Here you are assured the container is stopped.
```

Dynamic deployment

Dynamic deployment means that the Deployable is deployed after the container is started.

```
LocalContainer container = new Resin3xLocalContainer(
    new Resin3xStandaloneLocalConfiguration("target/myresin3x"));
container.setHome("c:/apps/resin-3.0.15");

container.start();

// Here you are assured the container is started.

Deployable war = new WAR("path/to/simple.war");
Deployer deployer = new ResinDeployer(container);
deployer.deploy(war)

// Here you are NOT sure the WAR has finished deploying. To be sure you
// need to use a DeployableMonitor to monitor the deployment. For example
// the following code deploys the WAR and wait until it is available to
// serve requests (the URL should point to a resource inside your WAR):
deployer.deploy(war, new URLDeployableMonitor("http://server:port/some/url"));

container.stop();

// Here you are assured the container is stopped.
```