


## Space Details

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

### Available Pages

- Home 
- Navigation
- Tested on
- SVN
- Using from Java
- Using from Ant
- Credits
- Documentation Archives
- Properties
- Resin 3.x
- Resin 2.x
- Orion 2.x
- Orion 1.x
- Tomcat 3.x
- Tomcat 4.x
- Tomcat 5.x
- Jetty 4.x
- JBoss 3.x
- Weblogic 7.x
- Debugging
- News

## Mission




### Cargo provides a Java API to start/stop and configure Java containers

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)

## Status

### Version status:































Version	Status	Comments
<a href="#">0.1</a>		Released on 11/09/04
<a href="#">0.2</a>		Released on 03/10/04
0.3		Not released yet

As glitches may happen even after a container is released for the first time, e.g. if a new feature is added to the framework, but not supported by all containers, we encourage you to report your success/failures in the [Tested on](#) section.

## Feature list

- Provides a [Java API](#) to:
  - Start containers
  - Stop containers
  - In-place startup (i.e. ability to start the container in any user-specified directory)
  - Wait for containers to be started, ensuring that the container is fully started after the start call returns
  - Wait for containers to be stopped, ensuring that the container is fully stopped after the start call returns
  - Supports static WAR deployments including support for expanded WARs (directories pointing to a WAR structure)

- Supports static EAR deployments
- Configurable [container properties](#)
- Provides [Ant tasks](#) that wraps the Java API.
- Supports the following containers:

Container	Java API/version	Ant API/version	Maven API/version
<a href="#">JBoss 3.x</a>	 ???	 ???	 N/A
<a href="#">Jetty 4.x</a>	 0.1	 ???	 N/A
<a href="#">Orion 1.x</a>	 0.1	 0.1	 N/A
<a href="#">Orion 2.x</a>	 0.1	 0.1	 N/A
<a href="#">Resin 2.x</a>	 0.1	 0.1	 N/A
<a href="#">Resin 3.x</a>	 0.1	 0.1	 N/A
<a href="#">Tomcat 3.x</a>	 0.1	 0.1	 N/A
<a href="#">Tomcat 4.x</a>	 0.1	 0.1	 N/A
<a href="#">Tomcat 5.x</a>	 0.1	 0.1	 N/A
<a href="#">WebLogic 7.x</a>	 ???	 ???	 N/A

In addition the Cargo project also offers a Java API to manipulate J2EE descriptors (currently `web.xml` and `application.xml`). Most notably the API allows merging two `web.xml` files.

## Quick Start

The following piece of code demonstrates how to configure Resin 3.0.8 to start in `target/resin3x` and deploy a WAR located in `src/testinput/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.

**WARNING: Do not point the working dir to an existing directory. Everything in this directory will be deleted. This is a dangerous bug in version 0.1 that is fixed in version 0.2**

```
Container container = new Resin3xContainer();
container.setHomeDir("c:/apps/resin-3.0.8");
container.setWorkingDir("target/resin3x");
```

```
Deployable deployable =
container.getDeployableFactory().createWAR("src/testinput/simple.war");

container.addDeployable(deployable);

container.start();

// At this point you are assured the container is started.

container.stop();

// At this point you are assured the container is stopped.
```

## Navigation

---

This page last changed on Oct 03, 2004 by [vmassol](#).

## Cargo 0.2 doc

- [Home](#)
- [News](#)
- [Using from Java](#)
- [Using from Ant](#)
- [Properties](#)
- [Debugging](#)
- [Javadoc](#)
- [License](#)

## Archives

- [Doc Archives](#)
- [Cargo 0.1 doc](#)

## Download

- [Cargo 0.1](#)

## Containers

- [JBoss 3.x](#)
- [Jetty 4.x](#)
- [Orion 1.x](#)
- [Orion 2.x](#)
- [Resin 2.x](#)
- [Resin 3.x](#)
- [Tomcat 3.x](#)
- [Tomcat 4.x](#)
- [Tomcat 5.x](#)
- [Weblogic 7.x](#)

## Support

- [Issues](#)
- [Roadmap](#)
- [Change log](#)

## Community

- [Mailing Lists](#)
- [Who we are](#)

## Developers

- [SVN](#)
- [Wiki](#)
- [Maven site](#)
- [Credits](#)

## Tested on

---

This page last changed on Sep 19, 2004 by [vmassol](#).

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

This page will then contain results of testing the framework in real world configurations.

Add your own experiences to the section matching your framework version, using the following format:

- Tomcat
  - 4.1.27 (J2EE 1.2 and J2EE 1.3) – [Vincent Massol](#), on 10th of September 3001
  - 4.1.28 (J2EE 1.3) **failed** – [jerome@coffeebreaks.org](mailto:jerome@coffeebreaks.org), on 11th of September 3001

### Cargo 0.2

- Resin
- Tomcat
- Orion
- Jetty

### Cargo 0.1

- Resin
  - 3.0.8 (J2EE 1.3) – [Vincent Massol](#), on 17 Aug 2004
- Tomcat
  - 3.3.2 (J2EE 1.3) – [Vincent Massol](#), on 5 Sep 2004
  - 4.1.30 (J2EE 1.3) – [Vincent Massol](#), on 5 Sep 2004
  - 5.0.25 (J2EE 1.3) – [Vincent Massol](#), on 5 Sep 2004
  - 5.0.28 (J2EE 1.3) – [Vincent Massol](#), on 5 Sep 2004
- Orion
  - 1.6.0b (J2EE 1.3) – [Vincent Massol](#), on 17 Aug 2004
  - 2.0.3 (J2EE 1.3) – [Vincent Massol](#), on 17 Aug 2004
- Jetty
  - 4.1.20 (J2EE 1.3) – [Vincent Massol](#), on 18 Aug 2004
  - 4.2.17 (J2EE 1.3) – [Vincent Massol](#), on 19 Sep 2004

## SVN

---

This page last changed on Aug 20, 2004 by [vmassol](#).

For general information see the [SVN page on Codehaus](#).

### Web Access

<http://svn.cargo.codehaus.org>

### Anonymous SVN Access

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

### Developer SVN Access via SSH

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

### SVN Access behind a firewall

Currently Codehaus does not support WebDAV access.



# Instantiating a container

There are 2 solutions to instantiate a container:

- by explicitly creating a new instance of the container itself. For example to instantiate a Resin 3.x container:

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

- by using the ContainerFactory class. The advantage is then that you can instantiate 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 ContainerFactory();  
Container container = factory.createContainer("resin3x");
```

Note: You can also pass the full container class name (that's useful if you wish to instantiate a custom container you have developed):

```
Container container =  
factory.createContainer("org.codehaus.cargo.container.resin.Resin3xContainer");
```

# Examples using the Cargo Java API

**WARNING: Do not point the working dir to an existing directory. Everything in this directory will be deleted. This is a dangerous bug in version 0.1 that is fixed in version 0.2**

## Starting Resin 3.x with no deployables

Note: The homeDir and workingDir property are mandatory.

```
Container container = new Resin3xContainer();  
container.setHomeDir("c:/apps/resin-3.0.8");  
container.setWorkingDir("target/resin3x");  
container.start();
```

## Starting Orion 2.x with an EAR to deploy

```

Container container = new Orion2xContainer();
container.setHomeDir("c:/apps/orion-2.0.3");
container.setWorkingDir("target/orion2x");

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

container.start();

```

## Starting Jetty 4.x with a WAR to deploy

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 homeDir property has not effect.

```

Container container = new Jetty4xContainer();
container.setWorkingDir("target/jetty4x");

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

container.start();

```

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

```

Container container = new Tomcat4xContainer();
container.setHomeDir("c:/apps/jakarta-tomcat-4.1.30");
container.setWorkingDir("target/tomcat4x");

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.

## Starting Tomcat 5.x on a specific port

```

Container container = new Tomcat5xContainer();
container.setHomeDir("c:/apps/jakarta-tomcat-5.0.25");
container.setWorkingDir("target/tomcat5x");

container.setProperty(ServletPropertySet.PORT, "8888");

```

```
container.start();
```

## Starting Orion 1.x with some additional classpath entries

This can be useful if you need to add some jars to the container classpath. For example if you have instrumented your source code with Clover you'll need to add the Clover jar to the classpath.

```
Container container = new Orion1xContainer();
container.setHomeDir("c:/apps/orion-1.6.0b");
container.setWorkingDir("target/orion1x");

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

container.start();
```

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

```
Container container = new Tomcat3xContainer();
container.setHomeDir("c:/apps/jakarta-tomcat-3.3.2");
container.setWorkingDir("target/tomcat3x");

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

container.start();
```

# Examples using Cargo with Ant

**WARNING: Do not point the working dir to an existing directory. Everything in this directory will be deleted. This is a dangerous bug in version 0.1 that is fixed in version 0.2**

Before being able to use the Cargo tasks you need to register them against Ant. This is done by using the Ant `<taskdef>` element:

```
<taskdef resource="cargo.tasks">
  <classpath>
    <pathelement location="{cargo.jar}"/>
  </classpath>
</taskdef>
```

## Starting Resin 3.x with no deployables

Note: The `homeDir` and `workingDir` properties are mandatory.

```
<cargo-resin3x homeDir="c:/apps/resin-3.0.8" workingDir="target/resin3x"
action="start"/>
```

## Starting Orion 2.x with an EAR to deploy

```
<cargo-orion2x homeDir="c:/apps/orion-2.0.3" workingDir="target/orion2x"
action="start">
  <ear earFile="src/data/some.ear"/>
</cargo-orion2x>
```

## Starting Orion 1.x with a WAR to deploy

```
<cargo-orion1x homeDir="c:/apps/orion-1.6.0b" workingDir="target/orion1x"
action="start">
  <war warFile="src/data/some.war"/>
</cargo-orion1x>
```

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

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

Use the `append="true|false"` attribute to decide whether the log file is recreated or whether it is appended to, keeping the previous execution logs.

## Starting Tomcat 5.x on a specific port

```
<cargo-tomcat5x homeDir="c:/apps/jakarta-tomcat-5.0.25" workingDir="target/tomcat5x"
action="start">
  <property name="cargo.servlet.port" value="8888"/>
</cargo-tomcat5x>
```

## Starting Orion 1.x with some additional classpath entries

This can be useful if you need to add some jars to the container classpath. For example if you have instrumented your source code with Clover you'll need to add the Clover jar to the classpath.

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

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

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

### Comments

---

I'd like to use Cargo with <http://webtest.canoo.com> . Usage would be much easier with:

- if the container type would be an attribute, instead of an element name, we could switch the container by setting a property. (I guess a factory can do the job).

- it would be great if `<cargo-xxx>` could be a `TaskContainer` that calls the nested elements after having started the container, and stopping it afterwards. This would allow to get around the error-prone `<parallel>` constructs that one need to do nowadays.

keep up the good work

**Mittie**

Posted by [mittie](#) at Sep 23, 2004.

---

## Credits

---

This page last changed on Sep 04, 2004 by [vmassol](#).

The following persons deserve credit for Cargo:

- Apache and The Jakarta cactus project: Cargo started as a refactoring of the Cactus Ant integration subproject
- [Vincent Massol](#): Lead developer of Cargo (and of Cactus)
- Christopher Lenz: Has developed most of the Cactus Ant integration code that has eventually found its way in Cargo
- Desire Atanga: Implementatoin of Tomcat support for the Java API
- Jerome Lacoste: General ideas and discussions about Cargo

## Documentation Archives

---

This page last changed on Oct 03, 2004 by [vmassol](#).

This web site contains the documentation for the next version of Cargo.

Available documentation PDFs:

- [Cargo 0.1 documentation](#)



## Properties

This page last changed on Sep 21, 2004 by [vmassol](#).

It is possible to set container configuration properties using the Cargo API.

Using Java you would write:

```
// Generic:
container.setProperty(propertyName, propertyValue);

// Example:
container.setProperty(ServletPropertySet.PORT, "8081");
```

Using Ant you would write:

```
<!-- Generic -->
<cargo-xxx [...]>
  <property name="propertyName" value="propertyValue"/>
</cargo-xxx>






<!-- Example -->
<cargo-resin3x homeDir="c:/apps/resin-3.0.8" workingDir="target/resin3x"
action="start"/>
  <property name="cargo.servlet.port" value="8081"/>
</cargo-resin3x>
```

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	"8081"
cargo.hostname	GeneralPropertySet.HOSTNAME	String	Host name on which the container will listen to	"myserver"
cargo.logging	GeneralPropertySet.LOGGING	"info", "warn" or "error"	Logging level	"error"

## Resin 3.x

This page last changed on Sep 26, 2004 by [vmassol](#).






Feature category	Feature name	Supported	Description
<b>Java API</b>	Start		Start the container
	Wait for start completion		Wait for start completion before returning
	Stop		Stops the container
	Wait for stop completion		Wait for stop completion before returning
	In-place startup		Ability to start the container in any user-specified directory
	Static deployment of WAR		Ability to deploy WARs when the container starts
	Static deployment of expanded WARs		Ability to deploy expanded WARs when the container starts
	Static deployment of EAR	N/A	Ability to deploy EARs when the container starts
	Standalone mode		The container has to be installed on the hard drive
	Embedded mode		Ability to start the container in embedded mode, i.e. without having to install it (requires the container jars to be in the classpath)

			though)
<b>Properties</b>	ServletPropertySet.PC		Port on which the Servlet/JSP container listens to
	GeneralPropertySet.H	 TNAME	Host name on which the container listens to
	GeneralPropertySet.L	 GING	Logging level
<b>Ant</b>	Generic Ant support		Support for all the Java API features
<b>Maven</b>	Generic Maven support		Support for all the Java API features

## Resin 2.x


This page last changed on Sep 26, 2004 by [vmassol](#).






Feature category	Feature name	Supported	Description
<b>Java API</b>	Start		Start the container
	Wait for start completion		Wait for start completion before returning
	Stop		Stops the container
	Wait for stop completion		Wait for stop completion before returning
	In-place startup		Ability to start the container in any user-specified directory
	Static deployment of WAR		Ability to deploy WARs when the container starts
	Static deployment of expanded WARs		Ability to deploy expanded WARs when the container starts
	Static deployment of EAR	N/A	Ability to deploy EARs when the container starts
	Standalone mode		The container has to be installed on the hard drive
	Embedded mode		Ability to start the container in embedded mode, i.e. without having to install it (requires the container jars to be in the classpath)

			though)
<b>Properties</b>	ServletPropertySet.PC		Port on which the Servlet/JSP container listens to
	GeneralPropertySet.H	 TNAME	Host name on which the container listens to
	GeneralPropertySet.L	 GING	Logging level
<b>Ant</b>	Generic Ant support		Support for all the Java API features
<b>Maven</b>	Generic Maven support		Support for all the Java API features

## Orion 2.x

This page last changed on Sep 26, 2004 by [vmassol](#).

Feature category	Feature name	Supported	Description
<b>Java API</b>	Start		Start the container
	Wait for start completion		Wait for start completion before returning
	Stop		Stops the container
	Wait for stop completion		Wait for stop completion before returning
	In-place startup		Ability to start the container in any user-specified directory
	Static deployment of WAR		Ability to deploy WARs when the container starts
	Static deployment of expanded WARs		Ability to deploy expanded WARs when the container starts
	Static deployment of EAR		Ability to deploy EARs when the container starts
	Standalone mode		The container has to be installed on the hard drive
	Embedded mode		Ability to start the container in embedded mode, i.e. without having to install it (requires the container jars to be in the classpath)






			though)
<b>Properties</b>	ServletPropertySet.PORT		Port on which the Servlet/JSP container listens to
	GeneralPropertySet.HOSTNAME		Host name on which the container listens to
	GeneralPropertySet.LOGGING		Logging level
<b>Ant</b>	Generic Ant support		Support for all the Java API features
<b>Maven</b>	Generic Maven support		Support for all the Java API features

## Orion 1.x

This page last changed on Sep 26, 2004 by [vmassol](#).

Feature category	Feature name	Supported	Description
<b>Java API</b>	Start		Start the container
	Wait for start completion		Wait for start completion before returning
	Stop		Stops the container
	Wait for stop completion		Wait for stop completion before returning
	In-place startup		Ability to start the container in any user-specified directory
	Static deployment of WAR		Ability to deploy WARs when the container starts
	Static deployment of expanded WARs		Ability to deploy expanded WARs when the container starts
	Static deployment of EAR		Ability to deploy EARs when the container starts
	Standalone mode		The container has to be installed on the hard drive
	Embedded mode		Ability to start the container in embedded mode, i.e. without having to install it (requires the container jars to be in the classpath)








			though)
<b>Properties</b>	ServletPropertySet.PORT		Port on which the Servlet/JSP container listens to
	GeneralPropertySet.HOSTNAME		Host name on which the container listens to
	GeneralPropertySet.LOGGING		Logging level
<b>Ant</b>	Generic Ant support		Support for all the Java API features
<b>Maven</b>	Generic Maven support		Support for all the Java API features

## Tomcat 3.x








This page last changed on Sep 26, 2004 by [vmassol](#).






Feature category	Feature name	Supported	Description
<b>Java API</b>	Start		Start the container
	Wait for start completion		Wait for start completion before returning
	Stop		Stops the container
	Wait for stop completion		Wait for stop completion before returning
	In-place startup		Ability to start the container in any user-specified directory
	Static deployment of WAR		Ability to deploy WARs when the container starts
	Static deployment of expanded WARs		Ability to deploy expanded WARs when the container starts
	Static deployment of EAR	N/A	Ability to deploy EARs when the container starts
	Standalone mode		The container has to be installed on the hard drive
	Embedded mode		Ability to start the container in embedded mode, i.e. without having to install it (requires the container jars to be in the classpath)

			though)
<b>Properties</b>	ServletPropertySet.PC		Port on which the Servlet/JSP container listens to
	GeneralPropertySet.H	 TNAME	Host name on which the container listens to
	GeneralPropertySet.L	 GING	Logging level
<b>Ant</b>	Generic Ant support		Support for all the Java API features
<b>Maven</b>	Generic Maven support		Support for all the Java API features

## Tomcat 4.x

This page last changed on Sep 26, 2004 by [vmassol](#).







Feature category	Feature name	Supported	Description
<b>Java API</b>	Start		Start the container
	Wait for start completion		Wait for start completion before returning
	Stop		Stops the container
	Wait for stop completion		Wait for stop completion before returning
	In-place startup		Ability to start the container in any user-specified directory
	Static deployment of WAR		Ability to deploy WARs when the container starts
	Static deployment of expanded WARs		Ability to deploy expanded WARs when the container starts
	Static deployment of EAR		Ability to deploy EARs when the container starts
	Standalone mode		The container has to be installed on the hard drive
	Embedded mode		Ability to start the container in embedded mode, i.e. without having to install it (requires the container jars to be in the classpath)

			though)
<b>Properties</b>	ServletPropertySet.PC		Port on which the Servlet/JSP container listens to
	GeneralPropertySet.H	 TNAME	Host name on which the container listens to
	GeneralPropertySet.L	 GING	Logging level
<b>Ant</b>	Generic Ant support		Support for all the Java API features
<b>Maven</b>	Generic Maven support		Support for all the Java API features

## Tomcat 5.x

This page last changed on Oct 03, 2004 by [vmassol](#).

Feature category	Feature name	Supported	Description
<b>Java API</b>	Start		Start the container
	Wait for start completion		Wait for start completion before returning
	Stop		Stops the container
	Wait for stop completion		Wait for stop completion before returning
	In-place startup		Ability to start the container in any user-specified directory
	Static deployment of WAR		Ability to deploy WARs when the container starts
	Static deployment of expanded WARs		Ability to deploy expanded WARs when the container starts
	Static deployment of EAR	N/A	Ability to deploy EARs when the container starts
	Standalone mode		The container has to be installed on the hard drive
	Embedded mode		Ability to start the container in embedded mode, i.e. without having to install it (requires the container jars to be in the classpath)






			though)
	Context file support		Ability to bundle context.xml files in WARs
<b>Properties</b>	ServletPropertySet.PC		Port on which the Servlet/JSP container listens to
	GeneralPropertySet.H	 TNAME	Host name on which the container listens to
	GeneralPropertySet.L	 GING	Logging level
<b>Ant</b>	Generic Ant support		Support for all the Java API features
<b>Maven</b>	Generic Maven support		Support for all the Java API features











## Jetty 4.x






This page last changed on Sep 26, 2004 by [vmassol](#).

Feature category	Feature name	Supported	Description
Java API	Start		Start the container
	Wait for start completion		Wait for start completion before returning
	Stop		Stops the container
	Wait for stop completion		Wait for stop completion before returning
	In-place startup		Ability to start the container in any user-specified directory
	Static deployment of WAR		Ability to deploy WARs when the container starts
	Static deployment of expanded WARs		Ability to deploy expanded WARs when the container starts
	Static deployment of EAR		Ability to deploy EARs when the container starts
	Standalone mode		The container has to be installed on the hard drive
	Embedded mode		Ability to start the container in embedded mode, i.e. without having to install it (requires the container jars to be in the classpath)













			though)
<b>Properties</b>	ServletPropertySet.PC		Port on which the Servlet/JSP container listens to
	GeneralPropertySet.H	 TNAME	Host name on which the container listens to
	GeneralPropertySet.L	 GING	Logging level
<b>Ant</b>	Generic Ant support		Support for all the Java API features
<b>Maven</b>	Generic Maven support		Support for all the Java API features






Feature category	Feature name	Supported	Description
<b>Java API</b>	Start		Start the container
	Wait for start completion		Wait for start completion before returning
	Stop		Stops the container
	Wait for stop completion		Wait for stop completion before returning
	In-place startup		Ability to start the container in any user-specified directory
	Static deployment of WAR		Ability to deploy WARs when the container starts
	Static deployment of expanded WARs		Ability to deploy expanded WARs when the container starts
	Static deployment of EAR		Ability to deploy EARs when the container starts
	Standalone mode		The container has to be installed on the hard drive
	Embedded mode		Ability to start the container in embedded mode, i.e. without having to install it (requires the container jars to be in the classpath)

			though)
<b>Properties</b>	ServletPropertySet.POR		Port on which the Servlet/JSP container listens to
	GeneralPropertySet.HO	 TNAME	Host name on which the container listens to
	GeneralPropertySet.L	 GING	Logging level
<b>Ant</b>	Generic Ant support		Support for all the Java API features
<b>Maven</b>	Generic Maven support		Support for all the Java API features

## Weblogic 7.x

This page last changed on Sep 26, 2004 by [vmassol](#).

Feature category	Feature name	Supported	Description
<b>Java API</b>	Start		Start the container
	Wait for start completion		Wait for start completion before returning
	Stop		Stops the container
	Wait for stop completion		Wait for stop completion before returning
	In-place startup		Ability to start the container in any user-specified directory
	Static deployment of WAR		Ability to deploy WARs when the container starts
	Static deployment of expanded WARs		Ability to deploy expanded WARs when the container starts
	Static deployment of EAR		Ability to deploy EARs when the container starts
	Standalone mode		The container has to be installed on the hard drive
	Embedded mode		Ability to start the container in embedded mode, i.e. without having to install it (requires the container jars to be in the classpath)

			though)
<b>Properties</b>	ServletPropertySet.POR		Port on which the Servlet/JSP container listens to
	GeneralPropertySet.HO	 TNAME	Host name on which the container listens to
	GeneralPropertySet.L	 GING	Logging level
<b>Ant</b>	Generic Ant support		Support for all the Java API features
<b>Maven</b>	Generic Maven support		Support for all the Java API features

## Debugging

---

This page last changed on Sep 23, 2004 by [vmassol](#).

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 redirect the container console (stdout) to a file. This is the first file you should check in case of problem.

## Getting some 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)

## Turning on container logs

Cargo is able to configure containers to generate various levels logs. There are 3 levels defined: "info", "warning" and "error". You can turn container logging by using the following API:

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

The generated log files will then be found in the Working directory you have specified

on the container (through the `container.setWorkingDir()` call).

## News

---

This page last changed on Oct 03, 2004 by [vmassol](#).

Add Cargo news here using the blog feature.