DataSource and Resource Support

Container support

Here are the configurations that currently support DataSource or Resource configuration:

> ⚠ This page / section has been automatically generated by Cargo's build. Do not edit it directly as it'll be overwritten next time it's generated again.

Container	Configuration	Resource	DataSource	Transactional DataSource	XA DataSource
Geronimo 2.x	`o.c.c.c.geronimo.Geronimo2xStandaloneLocalConfiguration`	❌	✅	✅	❌
Geronimo 3.x	`o.c.c.c.geronimo.Geronimo3xStandaloneLocalConfiguration`	❌	✅	✅	❌
Glassfish 3.x	`o.c.c.c.glassfish.GlassFish3xStandaloneLocalConfiguration`	❌	✅	✅	❌
Glassfish 4.x	`o.c.c.c.glassfish.GlassFish4xStandaloneLocalConfiguration`	❌	✅	✅	❌
JBoss 3.x	`o.c.c.c.jboss.JBossStandaloneLocalConfiguration`	❌	✅	✅	❌
JBoss 4.x	`o.c.c.c.jboss.JBossStandaloneLocalConfiguration`	❌	✅	✅	❌

JBoss 4.2.x	`o.c.c.c.jboss.JBoss42xStandaloneLocalConfiguration`	❌	✅	✅	❌
JBoss 5.x	`o.c.c.c.jboss.JBoss5xStandaloneLocalConfiguration`	❌	✅	✅	❌
JBoss 5.1.x	`o.c.c.c.jboss.JBoss51xStandaloneLocalConfiguration`	❌	✅	✅	❌
JBoss 6.x	`o.c.c.c.jboss.JBoss6xStandaloneLocalConfiguration`	❌	✅	✅	❌
JBoss 6.1.x	`o.c.c.c.jboss.JBoss6xStandaloneLocalConfiguration`	❌	✅	✅	❌
JBoss 7.x	`o.c.c.c.jboss.JBoss7xStandaloneLocalConfiguration`	❌	✅	✅	❌
JBoss 7.1.x	`o.c.c.c.jboss.JBoss71xStandaloneLocalConfiguration`	❌	✅	✅	❌
Jetty 7.x	`o.c.c.c.jetty.Jetty7xStandaloneLocalConfiguration`	❌	✅	❌	❌

Jetty 8.x	`o.c.c.c.jetty.Jetty8xStandaloneLocalConfiguration`	❌	✅	❌	❌
Jetty 9.x	`o.c.c.c.jetty.Jetty9xStandaloneLocalConfiguration`	❌	✅	❌	❌
JOnAS 4.x	`o.c.c.c.jonas.Jonas4xStandaloneLocalConfiguration`	❌	✅	✅	❌
JOnAS 5.x	`o.c.c.c.jonas.Jonas5xStandaloneLocalConfiguration`	❌	✅	✅	❌
JRun 4.x	`o.c.c.c.jrun.JRun4xExistingLocalConfiguration`	✅	✅	✅	✅
Oc4J 9.x	`o.c.c.c.orion.Oc4j9xStandaloneLocalConfiguration`	❌	✅	✅	✅
Oc4j 10.x	`o.c.c.c.orion.Oc4j10xExistingLocalConfiguration`	❌	✅	✅	✅
Resin 2.x	`o.c.c.c.resin.Resin2xStandaloneLocalConfiguration`	✅	✅	❌	❌
Resin 3.x	`o.c.c.c.resin.Resin3xStandaloneLocalConfiguration`	✅	✅	❌	❌

Resin 3.1.x	`o.c.c.c.resin.Resin31xStandaloneLocalConfiguration`	✅	✅	❌	❌
Resin 4.x	`o.c.c.c.resin.Resin4xStandaloneLocalConfiguration`	✅	✅	❌	❌
Tomcat 4.x	`o.c.c.c.tomcat.Tomcat4xStandaloneLocalConfiguration`	✅	✅	❌	❌
Tomcat 5.x	`o.c.c.c.tomcat.Tomcat5xStandaloneLocalConfiguration`	✅	✅	❌	❌
Tomcat 6.x	`o.c.c.c.tomcat.Tomcat6xStandaloneLocalConfiguration`	✅	✅	❌	❌
Tomcat 7.x	`o.c.c.c.tomcat.Tomcat7xStandaloneLocalConfiguration`	✅	✅	❌	❌
WebLogic 8.x	`o.c.c.c.weblogic.WebLogic8xStandaloneLocalConfiguration`	❌	✅	✅	✅
WebLogic 9.x	`o.c.c.c.weblogic.WebLogic9xStandaloneLocalConfiguration`	❌	✅	✅	✅

WebLogic 10.x	`o.c.c.c.weblogic.WebLogic10xStandaloneLocalConfiguration`	❌	✅	✅	✅
WebLogic 10.3.x	`o.c.c.c.weblogic.WebLogic103xStandaloneLocalConfiguration`	❌	✅	✅	✅
WebLogic 12.x	`o.c.c.c.weblogic.WebLogic12xStandaloneLocalConfiguration`	❌	✅	✅	✅

Notes:

- Datasources on the Jetty container use the c3p0 datasource pool.
- JOnAS handles transactions for datasources using its `jtm` service.

DataSource properties

DataSources are added through pipe-delimited configuration properties that starts with `cargo.datasource.datasource`. Example:

```
cargo.datasource.datasource1=cargo.datasource.url=jdbc:mydriver:userdb\|cargo.datasource.driver=org.database.Driver\|...
cargo.datasource.datasource2=cargo.datasource.url=jdbc:mydriver:referencedb\|cargo.datasource.driver=org.database.Driver\|...
```

Here are the properties that are valid for this:

- Note that c.d means cargo.datasource
- Note that if you specify a property marked **do not set** you will get a CargoException.

Property	Purpose	Valid Values	DataSource	Transactional DataSource	XA DataSource

`c.d.jndi`	The path to this datasource in JNDI	Any jndi path, like `jdbc/ds`	mandatory	mandatory	mandatory
`c.d.driver`	The implementation class	ex. `my.Driver`	mandatory: must implement `java.sql.Driver`	mandatory: must implement `java.sql.Driver`	mandatory: must implement `javax.sql.XADataSource`
`c.d.properties`	Properties to pass to the driver	Semi-colon delimited string	optional	optional	mandatory
`c.d.url`	URL for the `java.sql.Driver`	ex. `jdbc:host:port/mydb`	mandatory	mandatory	optional
`c.d.type`	Determines the type of the driver	Defaults to `java.sql.Driver`, only set if you want to use a `javax.sql.XADataSource`	**do not set**	**do not set**	`javax.sql.XADataSource`
`c.d.transactionsupport`	Determines transaction support type	`LOCAL_TRANSACTION` or `XA_TRANSACTION`	**do not set**	mandatory	unset defaults to only valid option: `XA_TRANSACTION`
`c.d.id`	Identifier used in configuration files to reference this datasource	Must contain no path-like characters	optional	optional	optional
`c.d.username`	Username to connect to the database	String	optional	optional	optional
`c.d.password`	Password to connect to the database	String	optional	optional	optional

Resource properties

Resources are added through pipe-delimited configuration properties that starts with `cargo.resource.resource`. Example:

```
cargo.resource.resource1=cargo.resource.name
=resource/apple\|cargo.resource.class=org.my
company.Apple\|...
cargo.resource.resource2=cargo.resource.name
=resource/pear\|cargo.resource.driver=org.my
company.Pear\|...
```

Here are the properties that are valid for this:

- Note that `c.r` means `cargo.resource`

Property	Purpose	Valid Values	Mandatory?
`c.r.name`	The path to this datasource in JNDI	Any jndi path, like `resource/apple`	mandatory
`c.r.type`	Interface of the object	Any valid interface	mandatory
`c.r.class`	The implementation class	Any valid class implementing the interface	mandatory
`c.r.parameters`	Properties to to populate the class with	Semi-colon delimited string; must correspond to setters	optional

Examples

Users of the Java API can take a look at the following classes as example:

- `DataSourceOnStandaloneConfigurationTest`: datasource definition.
- `TransactionEmulationDataSourceOnStandaloneConfigurationTest`: datasource definition with transaction emulation.
- `XATransactionDataSourceOnStandaloneConfigurationTest`: XA datasource definition.
- `ResourceOnStandaloneConfigurationTest`: resource definition, showing for example `mailsession`.

Users of the Maven2/Maven3 plugin can use the Maven2 Archetype showing datasource support. Please read: Data source Definition Archetype.