

# **Deployit JBoss Application Server Plugin Manual**

**Version 3.5.0**

# Table of Content

Preface	3
Overview	3
Features	3
Requirements	3
Usage in Deployment Packages	3
Using the deployables and deployed	4
Deployable vs. Container table	4
Deployed Actions Table	4
Deploying applications	5
Extension points	5
Extending the JBoss Plugin	5
Discovery	6
CI Reference	6
Configuration Item Overview	6
Deployable Configuration Items	6
Deployed Configuration Items	7
Topology Configuration Items	7
Virtual Deployed Configuration Items	7
Virtual Topology Configuration Items	7
Configuration Item Details	8
jboss-v4.Server	8
jboss-v5.Server	9
jboss-v6.Server	11
jboss.BaseServer	13
jboss.ConfigurationFile	15
jboss.ConfigurationFolder	15
jboss.Datasource	15
jboss.DatasourceSpec	17
jboss.DeployedConfigurationFile	18
jboss.DeployedConfigurationFolder	19
jboss.DeployedLibrary	21
jboss.Ear	22
jboss.EarModule	23
jboss.Library	24
jboss.NonTransactionalDatasource	25
jboss.NonTransactionalDatasourceSpec	27
jboss.Queue	28
jboss.QueueSpec	30
jboss.Resource	31
jboss.Topic	32
jboss.TopicSpec	34
jboss.TransactionaDatasource	35
jboss.TransactionaDatasourceSpec	36
jboss.TransactionaXADatasource	37
jboss.TransactionaXADatasourceSpec	39
jboss.War	40
jboss.WarModule	40

## Preface

This document describes the functionality provided by the JBoss application server (AS) plugin.

See the **Deployit Reference Manual** for background information on Deployit and deployment concepts.

## Overview

The JBoss plugin is a Deployit plugin that adds the capability to manage deployments and resources on JBoss application server. It works out of the box for deploying/ undeploying application artifacts, datasource and other JMS resources (see the *Features* section below) , and can easily be extended to support more deployment options or management of new artifacts/resources on JBoss AS.

## Features

- Application artifacts
  - Enterprise application (EAR)
  - Web application (WAR)
- Resources
  - Datasource
  - JMS Queue
  - JMS Topic
- Discovery

## Requirements

- **Deployit requirements**
  - **Deployit:** version 3.5+
  - **JBoss AS versions:** 4.x, 5.x and 6.x
  - **Other Deployit Plugins:** None
- **Infrastructural requirements**
  - **User credentials** for accessing the Host running the JBoss application Server.

## Usage in Deployment Packages

The plugin works with the standard deployment package of DAR format. Please see the *Packaging Manual* for more details about the DAR format and the ways to compose one.

The following is a sample MANIFEST.MF file that can be used to create a JBoss AS specific deployment package. It contain declarations for an [Ear](#), a [datasource](#) and a couple of JMS resources.

```
Manifest-Version: 1.0
Deployit-Package-Format-Version: 1.3
CI-Application: SampleApp
CI-Version: 1.0

Name: PetClinic-1.0.ear
CI-Name: PetClinic
CI-Type: jee.Ear

Name: testDatasource
CI-Type: jboss.TransactionaDatasourceSpec
CI-jndiName: jdbc/sampleDatasource
CI-connectionUrl: jdbc:mysql://localhost/test
CI-driverClass: com.mysql.jdbc.Driver
CI-username: {{DATABASE_USERNAME}}
```

CI-password: {{DATABASE_PASSWORD}}
Name: testQueue
CI-Type: jboss.QueueSpec
CI-jndiName: jms/testQueue
Name: testTopic
CI-Type: jboss.TopicSpec
CI-jndiName: jms/testTopic
Name: testConfigFile.txt
CI-Name: testConfigFiles
CI-Type: jboss.ConfigurationFile
Name: testConfigFolder
CI-Name: testConfigFolder
CI-Type: jboss.ConfigurationFolder

## Using the deployables and deployments

The following table describes which deployable/container combinations are possible.

### Deployable vs. Container table

Deployable	Container	Generated deployed
Application artifact: jee.Ear jee.War	Server	jboss.EarModule jboss.WarModule
jboss.TransactionDataSourceSpec jboss.TransactionXADataSourceSpec jboss.NonTransactionalDataSourceSpec	Server	jboss.TransactionDataSource jboss.TransactionXADataSource jboss.NonTransactionalDataSource
jboss.QueueSpec	Server	jboss.Queue
jboss.TopicSpec	Server	jboss.Topic
jboss.ConfigurationFile	Server	jboss.DeployedConfigurationFile
jboss.ConfigurationFolder	Server	jboss.DeployedConfigurationFolder

The following table describes the effect a deployed has on it's container

### Deployed Actions Table

Deployed	Actions performed for operations		
	Create	Destroy	Modify
jboss.EarModule jboss.WarModule	<ul style="list-style-type: none"> <li>stop Server</li> <li>deploy application</li> <li>start Server</li> </ul>	<ul style="list-style-type: none"> <li>stop Server</li> <li>undeploy old application version</li> <li>start Server</li> </ul>	<ul style="list-style-type: none"> <li>stop Server</li> <li>undeploy old application version</li> <li>deploy new application version</li> <li>start Server</li> </ul>
jboss.TransactionDataSource jboss.TransactionXADataSource jboss.NonTransactionalDataSource	<ul style="list-style-type: none"> <li>create datasource</li> </ul>	<ul style="list-style-type: none"> <li>destroy datasource</li> </ul>	<ul style="list-style-type: none"> <li>destroy datasource</li> <li>create modified datasource</li> </ul>
jboss.Queue	<ul style="list-style-type: none"> <li>stop Server</li> <li>create Queue</li> <li>start Server</li> </ul>	<ul style="list-style-type: none"> <li>stop Server</li> <li>destroy Queue</li> <li>start Server</li> </ul>	<ul style="list-style-type: none"> <li>stop Server</li> <li>destroy Queue</li> <li>create modified Queue</li> <li>start Server</li> </ul>

jboss.Topic	<ul style="list-style-type: none"> <li>• stop Server</li> <li>• create Topic</li> <li>• start Server</li> </ul>	<ul style="list-style-type: none"> <li>• stop Server</li> <li>• destroy Topic</li> <li>• start Server</li> </ul>	<ul style="list-style-type: none"> <li>• stop Server</li> <li>• destroy Topic</li> <li>• create modified Topic</li> <li>• start Server</li> </ul>
jboss.DeployedConfigurationFile	<ul style="list-style-type: none"> <li>• stop Server</li> <li>• deploying config files</li> <li>• start Server</li> </ul>	<ul style="list-style-type: none"> <li>• stop Server</li> <li>• undeploy config files</li> <li>• start Server</li> </ul>	<ul style="list-style-type: none"> <li>• stop Server</li> <li>• undeploy config files</li> <li>• deploy new config files</li> <li>• start Server</li> </ul>
jboss.DeployedConfigurationFolder	<ul style="list-style-type: none"> <li>• stop Server</li> <li>• deploying configuration folder</li> <li>• start Server</li> </ul>	<ul style="list-style-type: none"> <li>• stop Server</li> <li>• undeploy configuration folder</li> <li>• start Server</li> </ul>	<ul style="list-style-type: none"> <li>• stop Server</li> <li>• undeploy configuration folder</li> <li>• deploy new configuration folder</li> <li>• start Server</li> </ul>

## Deploying applications

By default, Deployit deploys the application artifacts and resource specifications (datasource, queues, topics etc) to the 'deploy' directory in the server's configuration. So if the server configuration is set to 'default' (which is the default value of server name), the artifact is copied to \${JBOSS\_HOME}/server/default/deploy. Also, the server is stopped before copying the artifact and then started again. But all these configurations can be customized to suit more specific scenarios.

## Extension points

The JBoss plugin is designed to be extended through Deployit's Plugin API type system. Also, since the JBoss plugin is built on top of generic-plugin, support for new types can be added using the generic plugin patterns. Refer to *Generic Plugin Manual* for more details. Also, refer to *Customization Manual* for an explanation of the type system.

The next section describes extensibility by examples:

## Extending the JBoss Plugin

### Making existing property hidden/visible or changing the default value

The following synthetic.xml snippet shows how the *restartRequired* property can be made visible and the *targetDirectory* property can be given a default value of '/home/deployer/install-files' for *jboss.EarModule*.

```
<type-modification type="jboss.EarModule">
  <!-- make it visible so that I can control whether to restart a Server or not from UI-->
  <property name="restartRequired" kind="boolean" default="true" hidden="false"/>

  <!-- custom deploy directory for my jboss applications -->
  <property name="targetDirectory" default="/home/deployer/install-files" hidden="true"/>
</type-modification>
```

### Adding a new property to a deployed/deployable

The following synthetic.xml snippet shows how a new property 'blocking-timeout-millis' can be added to *jboss.TransactionaDataSource*

```
<type-modification type="jboss.TransactionaDataSource">
  <!-- adding new property -->
  <property name="blockingTimeoutMillis" kind="integer" default="3000" description="maximum time in milliseconds to block
while waiting for a connection before throwing an exception"/>
</type-modification>
```

Note that while adding a new property in JBoss plugin, *the configuration property must be specified in lower camel-case with the hyphens removed from it*. Thus the property

'blocking-timeout-millis' has to be specified as *blockingTimeoutMillis*. Similarly 'idle-timeout-minutes' becomes *idleTimeoutMinutes* in synthetic.xml.

## Adding a new type

New types can be added in JBoss plugin using the `Generic Plugin` patterns. For example, the following synthetic.xml snippet defines a new type `jboss.EjbJarModule`

```
<type type="jboss.EjbJarModule" extends="generic.CopiedArtifact" deployable-type="jee.EjbJar"
      container-type="jboss.BaseServer">
  <generate-deployable type="jboss.EjbJar" extends="jee.EjbJar"/>
  <property name="targetDirectory" default="${deployed.container.home}/server/${deployed.container.serverName}/deploy"
            hidden="true"/>
  <property name="targetFile" default="${deployed.deployable.name}.jar" hidden="true"/>
  <property name="createOrder" kind="integer" default="50" hidden="true"/>
  <property name="destroyOrder" kind="integer" default="40" hidden="true"/>
  <property name="restartRequired" kind="boolean" default="true" hidden="true"/>
</type>
```

# Discovery

Once the JBoss server's home location and the host on which JBoss server is running are specified, the following properties of JBoss server can be discovered

- JBoss version
- Control port
- Http port
- Ajp port

Here is an example CLI script which discovers a sample JBoss server:

```
host = repository.create(factory.configurationItem('Infrastructure/jboss-51-host', 'overthere.SshHost',
  {'connectionType':'SFTP','address': 'jboss-51','username': 'root','password':'centos','os':'UNIX'}))
jboss = factory.configurationItem('Infrastructure/jboss-51-host/jboss-51', 'jboss-v5.Server',
  {'home':'/opt/jboss/5.1.0.GA', 'host':'Infrastructure/jboss-51-host'})
cis = deployit.discover(jboss)
print cis

#discovery just discovers the topology and keeps the configuration items in memory. Save them in Deployit repository
repository.create(cis)
```

## CI Reference

## Configuration Item Overview

### Deployable Configuration Items

CI	Description
<a href="#">jboss.ConfigurationFile</a>	File that may contain application configurations which is deployed to the server conf directory
<a href="#">jboss.ConfigurationFolder</a>	Folder that may contain files for application configuration which are deployed to the server conf directory
<a href="#">jboss.DatasourceSpec</a>	Base type for a JBoss datasource (deployable)
<a href="#">jboss.Ear</a>	A JEE EAR archive
<a href="#">jboss.Library</a>	The Library as deployed on the jboss server
<a href="#">jboss.NonTransactionalDatasourceSpec</a>	A JBoss No Transaction Datasource (deployable)
<a href="#">jboss.QueueSpec</a>	Specification for a JBoss queue
<a href="#">jboss.TopicSpec</a>	Specification for a JBoss topic
<a href="#">jboss.TransactionalDatasourceSpec</a>	A JBoss Local Transaction Datasource (deployable)
<a href="#">jboss.TransactionalXADatasourceSpec</a>	A JBoss XA Datasource (deployable)
<a href="#">jboss.War</a>	A JEE WAR archive

## Deployed Configuration Items

CI	Description
<a href="#">jboss.DeployedConfigurationFile</a>	Deployed configuration file
<a href="#">jboss.DeployedConfigurationFolder</a>	Deployed configuration folder
<a href="#">jboss.DeployedLibrary</a>	The Library as deployed on the jboss server
<a href="#">jboss.EarModule</a>	Ear with values configured for a deployment
<a href="#">jboss.NonTransactionalDatasource</a>	A JBoss No Transaction Datasource
<a href="#">jboss.Queue</a>	A JBoss Queue
<a href="#">jboss.Topic</a>	A JBoss topic
<a href="#">jboss.TransactionalDatasource</a>	A JBoss Local Transaction Datasource
<a href="#">jboss.TransactionalXADatasource</a>	A JBoss XA Datasource
<a href="#">jboss.WarModule</a>	War with values configured for a deployment

## Topology Configuration Items

CI	Description
<a href="#">jboss-v4.Server</a>	Jboss 4
<a href="#">jboss-v5.Server</a>	Jboss 5
<a href="#">jboss-v6.Server</a>	JBoss 6

## Virtual Deployed Configuration Items

CI	Description
<a href="#">jboss.Datasource</a>	Base type for a JBoss datasource
<a href="#">jboss.Resource</a>	A template deployed to a generic container

## Virtual Topology Configuration Items

CI	Description
<a href="#">jboss.BaseServer</a>	Base JBoss Server

## Configuration Item Details

### jboss-v4.Server

**Hierarchy** [jboss.BaseServer](#) >> generic.Container >> udm.BaseContainer >> udm.BaseConfigurationItem

**Interfaces** udm.ConfigurationItem, udm.Container

JBoss 4.x server

Public Properties
<b>ajpPort</b> * : INTEGER = 8009
AJP connector port
<b>controlPort</b> * : INTEGER = 1099
Control Port
<b>home</b> * : STRING
JBoss application server installation directory. e.g. /opt/jboss/5.1.0.GA
<b>host</b> * : CI<overthere.Host>
Host upon which the container resides
<b>httpPort</b> * : INTEGER = 8080
HTTP/1.1 connector port
<b>serverName</b> * : STRING = default
Server configuration name e.g. default or minimal



**Hidden Properties****inspectTemplate** : *STRING* = *jboss/v4/container/inspect.sh.ftl*

Inspect Template

**jmsResourceFileSuffix** : *STRING* = *service*

Jms Resource File Suffix

**queryTemplate** : *STRING* = *jboss/v4/container/query.sh.ftl*

Query Template

**restartOrder** : *INTEGER* = *90*

The order of the restart container step in the step list.

**restartWaitTime** : *INTEGER* = *0*

The time to wait in seconds for a container restart action.

**startOrder** : *INTEGER* = *90*

The order of the start container step in the step list.

**startScript** : *STRING* = *jboss/v4/container/start.sh*

Start Script

**startWaitTime** : *INTEGER* = *30*

Start Wait Time

**stopOrder** : *INTEGER* = *10*

The order of the stop container step in the step list.

**stopScript** : *STRING* = *jboss/v4/container/stop.sh*

Stop Script

**stopWaitTime** : *INTEGER* = *10*

Stop Wait Time

**version** : *INTEGER* = *4*

Version

**restartScript** : *STRING*

Classpath to the script used to restart the generic container.

**jboss-v5.Server****Hierarchy** [jboss.BaseServer](#) >> generic.Container >> udm.BaseContainer >> udm.BaseConfigurationItem**Interfaces** udm.ConfigurationItem, udm.Container

Jboss 5.x server

## Public Properties

**ajpPort**\* : **INTEGER** = *8009*

AJP connector port

**controlPort**\* : **INTEGER** = *1099*

Control Port

**home**\* : **STRING**

JBoss application server installation directory. e.g. /opt/jboss/5.1.0.GA

**host**\* : **CI<overthere.Host>**

Host upon which the container resides

**httpPort**\* : **INTEGER** = *8080*

HTTP/1.1 connector port

**restartWaitTime**\* : **INTEGER** = *20*

Restart Wait Time

**serverName**\* : **STRING** = *default*

Server configuration name e.g. default or minimal

**version**\* : **INTEGER** = *5*

Version

## Hidden Properties

**inspectTemplate\*** : *STRING* = *jboss/v5/container/inspect.sh.ftl*

Inspect Template

**jmsResourceFileSuffix\*** : *STRING* = *service*

Jms Resource File Suffix

**queryTemplate\*** : *STRING* = *jboss/v5/container/query.sh.ftl*

Query Template

**restartOrder\*** : *INTEGER* = *90*

The order of the restart container step in the step list.

**startOrder\*** : *INTEGER* = *90*

The order of the start container step in the step list.

**startScript\*** : *STRING* = *jboss/v5/container/start.sh*

Start Script

**startWaitTime\*** : *INTEGER* = *30*

Start Wait Time

**stopOrder\*** : *INTEGER* = *10*

The order of the stop container step in the step list.

**stopScript\*** : *STRING* = *jboss/v5/container/stop.sh*

Stop Script

**stopWaitTime\*** : *INTEGER* = *10*

Stop Wait Time

**restartScript** : *STRING*

Classpath to the script used to restart the generic container.

## jboss-v6.Server

**Hierarchy** [jboss.BaseServer](#) >> generic.Container >> udm.BaseContainer >> udm.BaseConfigurationItem

**Interfaces** udm.ConfigurationItem, udm.Container

JBoss 6.x Server

## Public Properties

**ajpPort**\* : **INTEGER** = *8009*

AJP connector port

**controlPort**\* : **INTEGER** = *1090*

Control Port

**home**\* : **STRING**

JBoss application server installation directory. e.g. /opt/jboss/5.1.0.GA

**host**\* : **CI<overthere.Host>**

Host upon which the container resides

**httpPort**\* : **INTEGER** = *8080*

HTTP/1.1 connector port

**serverName**\* : **STRING** = *default*

Server configuration name e.g. default or minimal

**version**\* : **INTEGER** = *6*

Version

## Hidden Properties

**inspectTemplate** : *STRING* = *jboss/v6/container/inspect.sh.ftl*

Inspect Template

**jmsResourceFileSuffix** : *STRING* = *hornetq-jms*

Jms Resource File Suffix

**queryTemplate** : *STRING* = *jboss/v6/container/query.sh.ftl*

Query Template

**resourceFileSuffix** : *STRING* = *hornetq-jms.xml*

Resource File Suffix

**restartOrder** : *INTEGER* = *90*

The order of the restart container step in the step list.

**restartWaitTime** : *INTEGER* = *0*

The time to wait in seconds for a container restart action.

**startOrder** : *INTEGER* = *90*

The order of the start container step in the step list.

**startScript** : *STRING* = *jboss/v6/container/start.sh*

Start Script

**startWaitTime** : *INTEGER* = *30*

Start Wait Time

**stopOrder** : *INTEGER* = *10*

The order of the stop container step in the step list.

**stopScript** : *STRING* = *jboss/v6/container/stop.sh*

Stop Script

**stopWaitTime** : *INTEGER* = *10*

Stop Wait Time

**restartScript** : *STRING*

Classpath to the script used to restart the generic container.

## jboss.BaseServer

**Hierarchy** generic.Container >> udm.BaseContainer >> udm.BaseConfigurationItem

**Interfaces** udm.ConfigurationItem, udm.Container

Base JBoss Server

## Public Properties

**ajpPort** : **INTEGER** = *8009*

AJP connector port

**controlPort** : **INTEGER** = *1099*

Connector control port

**home** : **STRING**

JBoss application server installation directory. e.g. /opt/jboss/5.1.0.GA

**host** : **CI<overthere.Host>**

Host upon which the container resides

**httpPort** : **INTEGER** = *8080*

HTTP/1.1 connector port

**serverName** : **STRING** = *default*

Server configuration name e.g. default or minimal

**version** : **INTEGER**

JBoss application server version

## Hidden Properties

**jmsResourceFileSuffix** : **STRING** = *service*

Jms Resource File Suffix

**restartOrder** : **INTEGER** = *90*

The order of the restart container step in the step list.

**restartWaitTime** : **INTEGER** = *0*

The time to wait in seconds for a container restart action.

**startOrder** : **INTEGER** = *90*

The order of the start container step in the step list.

**startWaitTime** : **INTEGER** = *0*

The time to wait in seconds for a container start action.

**stopOrder** : **INTEGER** = *10*

The order of the stop container step in the step list.

**stopWaitTime** : **INTEGER** = *0*

The time to wait in seconds for a container stop action.

**restartScript** : **STRING**

Classpath to the script used to restart the generic container.

**startScript** : **STRING**

Classpath to the script used to start the generic container.

**stopScript** : **STRING**

Classpath to the script used to stop the generic container.

## jboss.ConfigurationFile

**Hierarchy** generic.File >> udm.BaseDeployableFileArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem

**Interfaces** udm.Deployable, udm.SourceArtifact, udm.Artifact, udm.DeployableArtifact, udm.ConfigurationItem, udm.FileArtifact

File that may contain application configurations which is deployed to the server conf directory

### Public Properties

**placeholders** : SET\_OF\_STRING

Placeholders detected in this artifact

**targetFile** : STRING

Name of the artifact on the generic server.

### Hidden Properties

**textFileNamesRegex**\* : STRING = *.\+\. (cfg | conf | config | ini | properties | props | txt | asp | aspx | htm | html | jsf | jsp | xht | xhtml | sql | xml | xsd | xsl | xslt)*

Regular expression that matches file names of text files

## jboss.ConfigurationFolder

**Hierarchy** generic.Folder >> udm.BaseDeployableFolderArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem

**Interfaces** udm.Deployable, udm.SourceArtifact, udm.Artifact, udm.DeployableArtifact, udm.ConfigurationItem, udm.FolderArtifact

Folder that may contain files for application configuration which are deployed to the server conf directory

### Public Properties

**placeholders** : SET\_OF\_STRING

Placeholders detected in this artifact

**targetFile** : STRING

Name of the artifact on the generic server.

### Hidden Properties

**textFileNamesRegex**\* : STRING = *.\+\. (cfg | conf | config | ini | properties | props | txt | asp | aspx | htm | html | jsf | jsp | xht | xhtml | sql | xml | xsd | xsl | xslt)*

Regular expression that matches file names of text files

## jboss.Datasource

**Hierarchy** [jboss.Resource](#) >> generic.ProcessedTemplate >> generic.AbstractDeployedArtifact >> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

**Interfaces** udm.Deployed, udm.ConfigurationItem

Base type for a JBoss datasource

## Public Properties

**connectionUrl**\* : *STRING*

Connection Url

**container**\* : *Cl<udm.Container>*

The container on which this deployed runs.

**driverClass**\* : *STRING*

Driver Class

**jndiName**\* : *STRING*

Jndi Name

**maxPoolSize**\* : *INTEGER = 20*

Max Pool Size

**minPoolSize**\* : *INTEGER = 0*

Min Pool Size

**password**\* : *STRING*

Password

**targetFile**\* : *STRING = \${deployed.deployable.name}-ds.xml*

Target File

**userName**\* : *STRING*

User Name

**deployable** : *Cl<udm.Deployable>*

The deployable that this deployed is derived from.

**useJavaContext** : *BOOLEAN = true*

Use Java Context



## Hidden Properties

**createOrder** : INTEGER = 50

Create Order

**createVerb** : STRING = Create

Create Verb

**destroyOrder** : INTEGER = 40

Destroy Order

**destroyVerb** : STRING = Destroy

Destroy Verb

**inspectTemplate** : STRING = *jboss/v\${deployed.container.version}/datasource/inspect.sh.ftl*

Inspect Template

**modifyOrder** : INTEGER = 50

The order of the step in the step list for the modify operation.

**modifyVerb** : STRING = Modify

Modify Verb

**noopOrder** : INTEGER = 50

The order of the step in the step list for the noop operation.

**noopVerb** : STRING = Modify

Noop Verb

**targetDirectory** : STRING = *\${deployed.container.home}/server/\${deployed.container.serverName}/deploy*

Target Directory

**template** : STRING = *jboss/v\${deployed.container.version}/datasource/template.xml.ftl*

Template

**createTargetDirectory** : BOOLEAN = false

Create the target directory on the generic server if it does not exist.

**restartRequired** : BOOLEAN = false

Restart Required

**restartRequiredForNoop** : BOOLEAN = false

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : BOOLEAN = true

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## jboss.DatasourceSpec

**Hierarchy** generic.Resource >> udm.BaseDeployable >> udm.BaseConfigurationItem

**Interfaces** udm.Deployable, udm.ConfigurationItem

Base type for a JBoss datasource (deployable)

**Public Properties****connectionUrl** : *STRING*

Connection Url

**driverClass** : *STRING*

Driver Class

**jndiName** : *STRING*

Jndi Name

**maxPoolSize** : *STRING*

Max Pool Size

**minPoolSize** : *STRING*

Min Pool Size

**password** : *STRING*

Password

**targetFile** : *STRING*

Target File

**useJavaContext** : *STRING*

Use Java Context

**userName** : *STRING*

User Name

**jboss.DeployedConfigurationFile**

**Hierarchy** generic.CopiedArtifact >> generic.AbstractDeployedArtifact >> generic.AbstractDeployed >>  
udm.BaseDeployed >> udm.BaseConfigurationItem

**Interfaces** udm.Artifact, udm.Deployed, udm.ConfigurationItem, udm.DerivedArtifact

Deployed configuration file

**Public Properties****container** : *CI<udm.Container>*

The container on which this deployed runs.

**deployable** : *CI<udm.Deployable>*

The deployable that this deployed is derived from.

**placeholders** : *MAP\_STRING\_STRING*

A Map containing all the placeholders mapped to their values. Special values are &lt;ignore&gt; or &lt;empty&gt;

**targetFile** : *STRING*

Name of the artifact on the generic server.

**Hidden Properties****createOrder** : INTEGER = 50

Create Order

**createVerb** : STRING = Create

Create Verb

**destroyOrder** : INTEGER = 40

Destroy Order

**destroyVerb** : STRING = Destroy

Destroy Verb

**modifyOrder** : INTEGER = 50

The order of the step in the step list for the modify operation.

**modifyVerb** : STRING = Modify

Modify Verb

**noopOrder** : INTEGER = 50

The order of the step in the step list for the noop operation.

**noopVerb** : STRING = Modify

Noop Verb

**targetDirectory** : STRING = \${deployed.container.home}/server/\${deployed.container.serverName}/conf

Target Directory

**createTargetDirectory** : BOOLEAN = false

Create the target directory on the generic server if it does not exist.

**restartRequired** : BOOLEAN = true

Restart Required

**restartRequiredForNoop** : BOOLEAN = false

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : BOOLEAN = true

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

**jboss.DeployedConfigurationFolder****Hierarchy** generic.CopiedArtifact >> generic.AbstractDeployedArtifact >> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem**Interfaces** udm.Artifact, udm.Deployed, udm.ConfigurationItem, udm.DerivedArtifact

Deployed configuration folder

## Public Properties

**container**\* : `CI<udm.Container>`

The container on which this deployed runs.

**deployable** : `CI<udm.Deployable>`

The deployable that this deployed is derived from.

**placeholders** : `MAP_STRING_STRING`

A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>

**targetFile** : `STRING`

Name of the artifact on the generic server.

## Hidden Properties

**createOrder**\* : `INTEGER = 50`

Create Order

**createVerb**\* : `STRING = Create`

Create Verb

**destroyOrder**\* : `INTEGER = 40`

Destroy Order

**destroyVerb**\* : `STRING = Destroy`

Destroy Verb

**modifyOrder**\* : `INTEGER = 50`

The order of the step in the step list for the modify operation.

**modifyVerb**\* : `STRING = Modify`

Modify Verb

**noopOrder**\* : `INTEGER = 50`

The order of the step in the step list for the noop operation.

**noopVerb**\* : `STRING = Modify`

Noop Verb

**targetDirectory**\* : `STRING = ${deployed.container.home}/server/${deployed.container.serverName}/conf`

Target Directory

**createTargetDirectory** : `BOOLEAN = false`

Create the target directory on the generic server if it does not exist.

**restartRequired** : `BOOLEAN = true`

Restart Required

**restartRequiredForNoop** : `BOOLEAN = false`

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : `BOOLEAN = true`

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## jboss.DeployedLibrary

**Hierarchy** generic.CopiedArtifact >> generic.AbstractDeployedArtifact >> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

**Interfaces** udm.Artifact, udm.Deployed, udm.ConfigurationItem, udm.DerivedArtifact

The Library as deployed on the jboss server.

### Public Properties

**container\*** : [CI<udm.Container>](#)

The container on which this deployed runs.

**deployable** : [CI<udm.Deployable>](#)

The deployable that this deployed is derived from.

**placeholders** : [MAP\\_STRING\\_STRING](#)

A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>

**targetFile** : [STRING](#)

Name of the artifact on the generic server.

## Hidden Properties

**createOrder** : *INTEGER = 50*

Create Order

**createVerb** : *STRING = Create*

Create Verb

**destroyOrder** : *INTEGER = 40*

Destroy Order

**destroyVerb** : *STRING = Destroy*

Destroy Verb

**modifyOrder** : *INTEGER = 50*

The order of the step in the step list for the modify operation.

**modifyVerb** : *STRING = Modify*

Modify Verb

**noopOrder** : *INTEGER = 50*

The order of the step in the step list for the noop operation.

**noopVerb** : *STRING = Modify*

Noop Verb

**targetDirectory** : *STRING = \${deployed.container.home}/server/\${deployed.container.serverName}/lib*

Target Directory

**createTargetDirectory** : *BOOLEAN = false*

Create the target directory on the generic server if it does not exist.

**restartRequired** : *BOOLEAN = true*

Restart Required

**restartRequiredForNoop** : *BOOLEAN = false*

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : *BOOLEAN = true*

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## jboss.Ear

**Hierarchy** jee.Ear >> udm.BaseDeployableArchiveArtifact >> udm.BaseDeployableFileArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem

**Interfaces** udm.Deployable, udm.SourceArtifact, udm.ArchiveArtifact, udm.Artifact, udm.DeployableArtifact, udm.ConfigurationItem, udm.FileArtifact

A JEE EAR archive

## Public Properties

**placeholders** : *SET\_OF\_STRING*

Placeholders detected in this artifact

### Hidden Properties

**textFileNamesRegex**\* : `STRING` = `.\+\. (cfg | conf | config | ini | properties | props | txt | asp | aspx | htm | html | jsf | jsp | xht | xhtml | sql | xml | xsd | xsl | xslt)`

Regular expression that matches file names of text files

## jboss.EarModule

**Hierarchy** generic.CopiedArtifact >> generic.AbstractDeployedArtifact >> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

**Interfaces** udm.Artifact, udm.Deployed, udm.ConfigurationItem, udm.DerivedArtifact

Ear with values configured for a deployment

### Public Properties

**container**\* : `CI<udm.Container>`

The container on which this deployed runs.

**deployable** : `CI<udm.Deployable>`

The deployable that this deployed is derived from.

**placeholders** : `MAP_STRING_STRING`

A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>

## Hidden Properties

**createOrder** : *INTEGER = 50*

Create Order

**createVerb** : *STRING = Create*

Create Verb

**destroyOrder** : *INTEGER = 40*

Destroy Order

**destroyVerb** : *STRING = Destroy*

Destroy Verb

**modifyOrder** : *INTEGER = 50*

The order of the step in the step list for the modify operation.

**modifyVerb** : *STRING = Modify*

Modify Verb

**noopOrder** : *INTEGER = 50*

The order of the step in the step list for the noop operation.

**noopVerb** : *STRING = Modify*

Noop Verb

**targetDirectory** : *STRING = \${deployed.container.home}/server/\${deployed.container.serverName}/deploy*

Target Directory

**targetFile** : *STRING = \${deployed.deployable.name}.ear*

Target File

**createTargetDirectory** : *BOOLEAN = false*

Create the target directory on the generic server if it does not exist.

**restartRequired** : *BOOLEAN = true*

Restart Required

**restartRequiredForNoop** : *BOOLEAN = false*

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : *BOOLEAN = true*

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## jboss.Library

**Hierarchy** generic.File >> udm.BaseDeployableFileArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem

**Interfaces** udm.Deployable, udm.SourceArtifact, udm.Artifact, udm.DeployableArtifact, udm.ConfigurationItem, udm.FileArtifact

The Library as deployed on the jboss server. (deployable)



**Public Properties****placeholders** : SET\_OF\_STRING

Placeholders detected in this artifact

**targetFile** : STRING

Name of the artifact on the generic server.

**Hidden Properties****textFileNamesRegex\*** : STRING = *.\+\. (cfg | conf | config | ini | properties | props | txt | asp | aspx | htm | html | jsf | jsp | xht | xhtml | sql | xml | xsd | xsl | xslt)*

Regular expression that matches file names of text files

**jboss.NonTransactionalDataSource****Hierarchy** [jboss.DataSource](#) >> [jboss.Resource](#) >> generic.ProcessedTemplate >> generic.AbstractDeployedArtifact >> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem**Interfaces** udm.Deployed, udm.ConfigurationItem

A JBoss No Transaction Datasource

**Public Properties****connectionUrl**\* : *STRING*

Connection Url

**container**\* : *Cl<udm.Container>*

The container on which this deployed runs.

**driverClass**\* : *STRING*

Driver Class

**jndiName**\* : *STRING*

Jndi Name

**maxPoolSize**\* : *INTEGER = 20*

Max Pool Size

**minPoolSize**\* : *INTEGER = 0*

Min Pool Size

**password**\* : *STRING*

Password

**targetFile**\* : *STRING = \${deployed.deployable.name}-ds.xml*

Target File

**userName**\* : *STRING*

User Name

**deployable** : *Cl<udm.Deployable>*

The deployable that this deployed is derived from.

**useJavaContext** : *BOOLEAN = true*

Use Java Context

## Hidden Properties

**createOrder** : INTEGER = 50

Create Order

**createVerb** : STRING = *Create*

Create Verb

**destroyOrder** : INTEGER = 40

Destroy Order

**destroyVerb** : STRING = *Destroy*

Destroy Verb

**dsType** : STRING = *no-tx-datasource*

Ds Type

**inspectTemplate** : STRING = *jboss/v\${deployed.container.version}/datasource/inspect.sh.ftl*

Inspect Template

**modifyOrder** : INTEGER = 50

The order of the step in the step list for the modify operation.

**modifyVerb** : STRING = *Modify*

Modify Verb

**noopOrder** : INTEGER = 50

The order of the step in the step list for the noop operation.

**noopVerb** : STRING = *Modify*

Noop Verb

**targetDirectory** : STRING = *\${deployed.container.home}/server/\${deployed.container.serverName}/deploy*

Target Directory

**template** : STRING = *jboss/v\${deployed.container.version}/datasource/template.xml.ftl*

Template

**createTargetDirectory** : BOOLEAN = *false*

Create the target directory on the generic server if it does not exist.

**restartRequired** : BOOLEAN = *false*

Restart Required

**restartRequiredForNoop** : BOOLEAN = *false*

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : BOOLEAN = *true*

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## jboss.NonTransactionalDatasourceSpec

**Hierarchy** generic.Resource >> udm.BaseDeployable >> udm.BaseConfigurationItem

**Interfaces** udm.Deployable, udm.ConfigurationItem

## A JBoss No Transaction Datasource (deployable)

Public Properties	
<b>connectionUrl</b> :	STRING
Connection Url	
<b>driverClass</b> :	STRING
Driver Class	
<b>jndiName</b> :	STRING
Jndi Name	
<b>maxPoolSize</b> :	STRING
Max Pool Size	
<b>minPoolSize</b> :	STRING
Min Pool Size	
<b>password</b> :	STRING
Password	
<b>targetFile</b> :	STRING
Target File	
<b>useJavaContext</b> :	STRING
Use Java Context	
<b>userName</b> :	STRING
User Name	

## jboss.Queue

**Hierarchy** [jboss.Resource](#) >> generic.ProcessedTemplate >> generic.AbstractDeployedArtifact >> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

**Interfaces** udm.Deployed, udm.ConfigurationItem

A JBoss Queue

## Public Properties

**container\*** : `CI<udm.Container>`

The container on which this deployed runs.

**jndiName\*** : `STRING`

Jndi Name

**maxDepth\*** : `INTEGER = 0`

Max Depth

**targetFile\*** : `STRING = ${deployed.deployable.name}-${deployed.container.jmsResourceFileSuffix}.xml`

Target File

**deployable** : `CI<udm.Deployable>`

The deployable that this deployed is derived from.

## Hidden Properties

**createOrder** : INTEGER = 50

Create Order

**createVerb** : STRING = Create

Create Verb

**destroyOrder** : INTEGER = 40

Destroy Order

**destroyVerb** : STRING = Destroy

Destroy Verb

**inspectTemplate** : STRING = *jboss/v\${deployed.container.version}/queue/inspect.sh.ftl*

Inspect Template

**modifyOrder** : INTEGER = 50

The order of the step in the step list for the modify operation.

**modifyVerb** : STRING = Modify

Modify Verb

**noopOrder** : INTEGER = 50

The order of the step in the step list for the noop operation.

**noopVerb** : STRING = Modify

Noop Verb

**targetDirectory** : STRING = *\${deployed.container.home}/server/\${deployed.container.serverName}/deploy*

Target Directory

**template** : STRING = *jboss/v\${deployed.container.version}/queue/template.xml.ftl*

Template

**createTargetDirectory** : BOOLEAN = false

Create the target directory on the generic server if it does not exist.

**restartRequired** : BOOLEAN = true

Restart Required

**restartRequiredForNoop** : BOOLEAN = false

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : BOOLEAN = true

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## jboss.QueueSpec

**Hierarchy** generic.Resource >> udm.BaseDeployable >> udm.BaseConfigurationItem

**Interfaces** udm.Deployable, udm.ConfigurationItem

Specification for a JBoss queue

**Public Properties****jndiName** : *STRING*

Jndi Name

**maxDepth** : *STRING*

Max Depth

**targetFile** : *STRING*

Target File

**jboss.Resource**

**Hierarchy** generic.ProcessedTemplate >> generic.AbstractDeployedArtifact >> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

**Interfaces** udm.Deployed, udm.ConfigurationItem

A template deployed to a generic container

**Public Properties****container**\* : *CI<udm.Container>*

The container on which this deployed runs.

**deployable** : *CI<udm.Deployable>*

The deployable that this deployed is derived from.

**targetFile** : *STRING*

Name of the artifact on the generic server.

## Hidden Properties

**createOrder** : *INTEGER = 50*

The order of the step in the step list for the create operation.

**createVerb** : *STRING = Create*

Create Verb

**destroyOrder** : *INTEGER = 40*

The order of the step in the step list for the destroy operation.

**destroyVerb** : *STRING = Destroy*

Destroy Verb

**modifyOrder** : *INTEGER = 50*

The order of the step in the step list for the modify operation.

**modifyVerb** : *STRING = Modify*

Modify Verb

**noopOrder** : *INTEGER = 50*

The order of the step in the step list for the noop operation.

**noopVerb** : *STRING = Modify*

Noop Verb

**targetDirectory** : *STRING*

Path to which artifact must be copied to on the generic server.

**template** : *STRING*

Classpath to the freemarker template used to generate the content of the final text base artifact.

**createTargetDirectory** : *BOOLEAN = false*

Create the target directory on the generic server if it does not exist.

**restartRequired** : *BOOLEAN = false*

The generic container requires a restart for the action performed by this deployed.

**restartRequiredForNoop** : *BOOLEAN = false*

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : *BOOLEAN = true*

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## jboss.Topic

**Hierarchy** [jboss.Resource](#) >> generic.ProcessedTemplate >> generic.AbstractDeployedArtifact >> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

**Interfaces** udm.Deployed, udm.ConfigurationItem

A JBoss topic



## Public Properties

**container\*** : `CI<udm.Container>`

The container on which this deployed runs.

**jndiName\*** : `STRING`

Jndi Name

**maxDepth\*** : `INTEGER = 0`

Max Depth

**targetFile\*** : `STRING = ${deployed.deployable.name}-${deployed.container.jmsResourceFileSuffix}.xml`

Target File

**deployable** : `CI<udm.Deployable>`

The deployable that this deployed is derived from.

## Hidden Properties

**createOrder** : INTEGER = 50

Create Order

**createVerb** : STRING = Create

Create Verb

**destroyOrder** : INTEGER = 40

Destroy Order

**destroyVerb** : STRING = Destroy

Destroy Verb

**inspectTemplate** : STRING = *jboss/v\${deployed.container.version}/topic/inspect.sh.ftl*

Inspect Template

**modifyOrder** : INTEGER = 50

The order of the step in the step list for the modify operation.

**modifyVerb** : STRING = Modify

Modify Verb

**noopOrder** : INTEGER = 50

The order of the step in the step list for the noop operation.

**noopVerb** : STRING = Modify

Noop Verb

**targetDirectory** : STRING = *\${deployed.container.home}/server/\${deployed.container.serverName}/deploy*

Target Directory

**template** : STRING = *jboss/v\${deployed.container.version}/topic/template.xml.ftl*

Template

**createTargetDirectory** : BOOLEAN = false

Create the target directory on the generic server if it does not exist.

**restartRequired** : BOOLEAN = true

Restart Required

**restartRequiredForNoop** : BOOLEAN = false

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : BOOLEAN = true

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## jboss.TopicSpec

**Hierarchy** generic.Resource >> udm.BaseDeployable >> udm.BaseConfigurationItem

**Interfaces** udm.Deployable, udm.ConfigurationItem

Specification for a JBoss topic

**Public Properties****jndiName** : *STRING*

Jndi Name

**maxDepth** : *STRING*

Max Depth

**targetFile** : *STRING*

Target File

**jboss.TransactionDataSource**

**Hierarchy** [jboss.Datasource](#) >> [jboss.Resource](#) >> generic.ProcessedTemplate >> generic.AbstractDeployedArtifact  
 >> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

**Interfaces** udm.Deployed, udm.ConfigurationItem

A JBoss Local Transaction Datasource

**Public Properties****connectionUrl**\* : *STRING*

Connection Url

**container**\* : *Cl<udm.Container>*

The container on which this deployed runs.

**driverClass**\* : *STRING*

Driver Class

**jndiName**\* : *STRING*

Jndi Name

**maxPoolSize**\* : *INTEGER = 20*

Max Pool Size

**minPoolSize**\* : *INTEGER = 0*

Min Pool Size

**password**\* : *STRING*

Password

**targetFile**\* : *STRING = \${deployed.deployable.name}-ds.xml*

Target File

**userName**\* : *STRING*

User Name

**deployable** : *Cl<udm.Deployable>*

The deployable that this deployed is derived from.

**useJavaContext** : *BOOLEAN = true*

Use Java Context

## Hidden Properties

**createOrder** : INTEGER = 50

Create Order

**createVerb** : STRING = Create

Create Verb

**destroyOrder** : INTEGER = 40

Destroy Order

**destroyVerb** : STRING = Destroy

Destroy Verb

**dsType** : STRING = local-tx-datasource

Ds Type

**inspectTemplate** : STRING = jboss/v\${deployed.container.version}/datasource/inspect.sh.ftl

Inspect Template

**modifyOrder** : INTEGER = 50

The order of the step in the step list for the modify operation.

**modifyVerb** : STRING = Modify

Modify Verb

**noopOrder** : INTEGER = 50

The order of the step in the step list for the noop operation.

**noopVerb** : STRING = Modify

Noop Verb

**targetDirectory** : STRING = \${deployed.container.home}/server/\${deployed.container.serverName}/deploy

Target Directory

**template** : STRING = jboss/v\${deployed.container.version}/datasource/template.xml.ftl

Template

**createTargetDirectory** : BOOLEAN = false

Create the target directory on the generic server if it does not exist.

**restartRequired** : BOOLEAN = false

Restart Required

**restartRequiredForNoop** : BOOLEAN = false

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : BOOLEAN = true

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## jboss.TransactionDataSourceSpec

**Hierarchy** generic.Resource >> udm.BaseDeployable >> udm.BaseConfigurationItem

**Interfaces** udm.Deployable, udm.ConfigurationItem

## A JBoss Local Transaction Datasource (deployable)

Public Properties	
<b>connectionUrl</b> :	STRING
Connection Url	
<b>driverClass</b> :	STRING
Driver Class	
<b>jndiName</b> :	STRING
Jndi Name	
<b>maxPoolSize</b> :	STRING
Max Pool Size	
<b>minPoolSize</b> :	STRING
Min Pool Size	
<b>password</b> :	STRING
Password	
<b>targetFile</b> :	STRING
Target File	
<b>useJavaContext</b> :	STRING
Use Java Context	
<b>userName</b> :	STRING
User Name	

## **jboss.TransactionXADatasource**

**Hierarchy** [jboss.Datasource](#) >> [jboss.Resource](#) >> generic.ProcessedTemplate >> generic.AbstractDeployedArtifact  
>> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

**Interfaces** udm.Deployed, udm.ConfigurationItem

A JBoss XA Datasource

**Public Properties****connectionUrl**\* : *STRING*

Connection Url

**container**\* : *Cl<udm.Container>*

The container on which this deployed runs.

**driverClass**\* : *STRING*

Driver Class

**jndiName**\* : *STRING*

Jndi Name

**maxPoolSize**\* : *INTEGER = 20*

Max Pool Size

**minPoolSize**\* : *INTEGER = 0*

Min Pool Size

**password**\* : *STRING*

Password

**targetFile**\* : *STRING = \${deployed.deployable.name}-ds.xml*

Target File

**userName**\* : *STRING*

User Name

**deployable** : *Cl<udm.Deployable>*

The deployable that this deployed is derived from.

**useJavaContext** : *BOOLEAN = true*

Use Java Context

## Hidden Properties

**createOrder** : INTEGER = 50

Create Order

**createVerb** : STRING = Create

Create Verb

**destroyOrder** : INTEGER = 40

Destroy Order

**destroyVerb** : STRING = Destroy

Destroy Verb

**dsType** : STRING = xa-datasource

Ds Type

**inspectTemplate** : STRING = jboss/v\${deployed.container.version}/datasource/inspect.sh.ftl

Inspect Template

**modifyOrder** : INTEGER = 50

The order of the step in the step list for the modify operation.

**modifyVerb** : STRING = Modify

Modify Verb

**noopOrder** : INTEGER = 50

The order of the step in the step list for the noop operation.

**noopVerb** : STRING = Modify

Noop Verb

**targetDirectory** : STRING = \${deployed.container.home}/server/\${deployed.container.serverName}/deploy

Target Directory

**template** : STRING = jboss/v\${deployed.container.version}/datasource/template.xml.ftl

Template

**createTargetDirectory** : BOOLEAN = false

Create the target directory on the generic server if it does not exist.

**restartRequired** : BOOLEAN = false

Restart Required

**restartRequiredForNoop** : BOOLEAN = false

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : BOOLEAN = true

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## jboss.TransactionaIXADatasourceSpec

**Hierarchy** generic.Resource >> udm.BaseDeployable >> udm.BaseConfigurationItem

**Interfaces** udm.Deployable, udm.ConfigurationItem

## A JBoss XA Datasource (deployable)

Public Properties	
<b>connectionUrl</b> :	STRING
Connection Url	
<b>driverClass</b> :	STRING
Driver Class	
<b>jndiName</b> :	STRING
Jndi Name	
<b>maxPoolSize</b> :	STRING
Max Pool Size	
<b>minPoolSize</b> :	STRING
Min Pool Size	
<b>password</b> :	STRING
Password	
<b>targetFile</b> :	STRING
Target File	
<b>useJavaContext</b> :	STRING
Use Java Context	
<b>userName</b> :	STRING
User Name	

## jboss.War

**Hierarchy** jee.War >> udm.BaseDeployableArchiveArtifact >> udm.BaseDeployableFileArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem

**Interfaces** udm.Deployable, udm.SourceArtifact, udm.ArchiveArtifact, udm.Artifact, udm.DeployableArtifact, udm.ConfigurationItem, udm.FileArtifact

A JEE WAR archive

Public Properties	
<b>placeholders</b> :	SET_OF_STRING
Placeholders detected in this artifact	
Hidden Properties	
<b>textFileNamesRegex</b> *	STRING = <i>.\+\. (cfg   conf   config   ini   properties   props   txt   asp   aspx   htm   html   jsf   jsp   xht   xhtml   sql   xml   xsd   xsl   xslt)</i>
Regular expression that matches file names of text files	

## jboss.WarModule



**Hierarchy** generic.CopiedArtifact >> generic.AbstractDeployedArtifact >> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

**Interfaces** udm.Artifact, udm.Deployed, udm.ConfigurationItem, udm.DerivedArtifact

War with values configured for a deployment

### Public Properties

**container**\* : [CI<udm.Container>](#)

The container on which this deployed runs.

**deployable** : [CI<udm.Deployable>](#)

The deployable that this deployed is derived from.

**placeholders** : [MAP\\_STRING\\_STRING](#)

A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>

## Hidden Properties

**createOrder** : INTEGER = 50

Create Order

**createVerb** : STRING = *Create*

Create Verb

**destroyOrder** : INTEGER = 40

Destroy Order

**destroyVerb** : STRING = *Destroy*

Destroy Verb

**modifyOrder** : INTEGER = 50

The order of the step in the step list for the modify operation.

**modifyVerb** : STRING = *Modify*

Modify Verb

**noopOrder** : INTEGER = 50

The order of the step in the step list for the noop operation.

**noopVerb** : STRING = *Modify*

Noop Verb

**targetDirectory** : STRING = *\${deployed.container.home}/server/\${deployed.container.serverName}/deploy*

Target Directory

**targetFile** : STRING = *\${deployed.deployable.name}.war*

Target File

**createTargetDirectory** : BOOLEAN = *false*

Create the target directory on the generic server if it does not exist.

**restartRequired** : BOOLEAN = *true*

Restart Required

**restartRequiredForNoop** : BOOLEAN = *false*

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : BOOLEAN = *true*

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.