

# **Deployit JBoss Application Server Plugin Manual**

**Version 3.8.0**

## Table of Contents

Table of Contents	2
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
Deployables	6
Deployeds	6
Containers	7
Other Configuration Items	7
Configuration Item Details	7
jbossas.Artifact	7
jbossas.BaseServer	8
jbossas.Configuration	9
jbossas.ConfigurationFile	10
jbossas.ConfigurationFolder	11
jbossas.Datasource	11
jbossas.DeployedConfigurationFile	13
jbossas.DeployedConfigurationFolder	14
jbossas.DeployedLibrary	15
jbossas.Ear	16
jbossas.EarModule	17
jbossas.Library	18
jbossas.NonTransactionalDatasource	19
jbossas.NonTransactionalDatasourceSpec	20
jbossas.Queue	21
jbossas.QueueSpec	22
jbossas.Resource	23
jbossas.ServerV4	23
jbossas.ServerV5	25
jbossas.ServerV6	26
jbossas.Topic	27
jbossas.TopicSpec	28
jbossas.TransactionaDatasource	29
jbossas.TransactionaDatasourceSpec	30
jbossas.TransactionaXADatasource	31
jbossas.TransactionaXADatasourceSpec	33
jbossas.War	34
jbossas.WarModule	34

## 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)
- JBoss-specific artifacts:
  - Service Archive (SAR)
  - Resource Archive (RAR)
  - Hibernate Archive (HAR)
  - Aspect archive (AOP)
- Resources
  - Datasource
  - JMS Queue\*\*
  - JMS Topic\*\*
- Discovery

\*\* While creating JMS resources like Queues and Topics for JBoss AS 6, only the jndi name is used for the creation purpose. Other properties like RedeliveryDelay, MaxDeliveryAttempts and so on are not used even when they have been defined on the CI (in synthetic.xml) and set. These properties are defined by editing the global server configuration, which can be found at "%JBOSS\_HOME%/server/<configuration/deploy/hornetq/hornetq-jms.xml).

## Requirements

- **Deployit requirements**
  - **Deployit:** version 3.7+
  - **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: jbossas.TransactionDataSourceSpec
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: jbossas.QueueSpec
CI-jndiName: jms/testQueue
Name: testTopic
CI-Type: jbossas.TopicSpec
CI-jndiName: jms/testTopic
Name: testConfigFile.txt
CI-Name: testConfigFiles
CI-Type: jbossas.ConfigurationFile
Name: testConfigFolder
CI-Name: testConfigFolder
CI-Type: jbossas.ConfigurationFolder

## Using the deployables and deployed

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	jbossas.EarModule jbossas.WarModule
jbossas.TransactionDataSourceSpec jbossas.TransactionXADatasourceSpec jbossas.NonTransactionalDataSourceSpec	Server	jbossas.TransactionDataSource jbossas.TransactionXADatasource jbossas.NonTransactionalDataSource
jbossas.QueueSpec	Server	jbossas.Queue
jbossas.TopicSpec	Server	jbossas.Topic
jbossas.ConfigurationFile	Server	jbossas.DeployedConfigurationFile
jbossas.ConfigurationFolder	Server	jbossas.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
jbossas.EarModule jbossas.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>
jbossas.TransactionDataSource jbossas.TransactionXADatasource jbossas.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>
jbossas.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>
jbossas.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>
jbossas.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>

jbossas.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 *jbossas.EarModule*.

```
<type-modification type="jbossas.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 *jbossas.TransactionDataSource*

```
<type-modification type="jbossas.TransactionDataSource">
  <!-- 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 *jbossas.EjbJarModule*

```
<type type="jbossas.EjbJarModule" extends="generic.CopiedArtifact" deployable-type="jee.EjbJar"
  container-type="jbossas.BaseServer">
  <generate-deployable type="jbossas.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 can be discovered on a running JBoss server.

- 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', 'jbossas.ServerV5',
  ('home':'/opt/jboss/5.1.0.GA', 'host':'Infrastructure/jboss-51-host'))

taskId = deployit.createDiscoveryTask(jboss)
deployit.startTaskAndWait(taskId)
cis = deployit.retrieveDiscoveryResults(taskId)

deployit.print(cis)

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

Few things to note about the above discovery example:

- Hosts are created under the Infrastructure tree, so the host id has been kepts as 'Infrastructure/jboss-51-host'
- Host address can be the host IP address or the dns name defined for the host
- JBoss server has a containment relation with a Host (created under a Host), so the server id has been kept as 'Infrastructure/jboss-51-host/jboss-51'

## CI Reference

### Configuration Item Overview

#### Deployables

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

#### Deployeds

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

## Containers

CI	Description
<a href="#">jbossas.BaseServer</a>	Base JBoss Application Server
<a href="#">jbossas.ServerV4</a>	Jboss 4 application server
<a href="#">jbossas.ServerV5</a>	Jboss 5 application server
<a href="#">jbossas.ServerV6</a>	Jboss 6 application server

## Other Configuration Items

CI	Description
<a href="#">jbossas.Artifact</a>	Base type for a JBoss artifact
<a href="#">jbossas.BaseServer</a>	Base JBoss Application Server
<a href="#">jbossas.Configuration</a>	Base type for a JBoss configuration
<a href="#">jbossas.ConfigurationFile</a>	File containing application configurations that is deployed to the server conf directory
<a href="#">jbossas.ConfigurationFolder</a>	Folder containing files for application configuration that are deployed to the server conf directory
<a href="#">jbossas.Datasource</a>	Base type for a JBoss datasource
<a href="#">jbossas.DeployedConfigurationFile</a>	Deployed configuration file
<a href="#">jbossas.DeployedConfigurationFolder</a>	Deployed configuration folder
<a href="#">jbossas.DeployedLibrary</a>	The Library as deployed on the jboss server
<a href="#">jbossas.Ear</a>	A JEE EAR archive
<a href="#">jbossas.EarModule</a>	Ear with values configured for a deployment
<a href="#">jbossas.Library</a>	The Library as deployed on the jboss server
<a href="#">jbossas.NonTransactionalDatasource</a>	A JBoss No Transaction Datasource
<a href="#">jbossas.NonTransactionalDatasourceSpec</a>	A JBoss No Transaction Datasource (deployable)
<a href="#">jbossas.Queue</a>	A JBoss Queue
<a href="#">jbossas.QueueSpec</a>	Specification for a JBoss queue
<a href="#">jbossas.Resource</a>	Description unavailable
<a href="#">jbossas.ServerV4</a>	Jboss 4 application server
<a href="#">jbossas.ServerV5</a>	Jboss 5 application server
<a href="#">jbossas.ServerV6</a>	Jboss 6 application server
<a href="#">jbossas.Topic</a>	A JBoss topic
<a href="#">jbossas.TopicSpec</a>	Specification for a JBoss topic
<a href="#">jbossas.TransactionalDatasource</a>	A JBoss Local Transaction Datasource
<a href="#">jbossas.TransactionalDatasourceSpec</a>	A JBoss Local Transaction Datasource (deployable)
<a href="#">jbossas.TransactionalXADatasource</a>	A JBoss XA Datasource
<a href="#">jbossas.TransactionalXADatasourceSpec</a>	A JBoss XA Datasource (deployable)
<a href="#">jbossas.War</a>	A JEE WAR archive
<a href="#">jbossas.WarModule</a>	War with values configured for a deployment

## Configuration Item Details

### jbossas.Artifact

#### Virtual Type

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

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

Base type for a JBoss artifact.

Parent	
<b>* container</b> : <code>CI&lt;udm.Container&gt;</code>	The container on which this deployed runs.
Public Properties	
<b>deployable</b> : <code>CI&lt;udm.Deployable&gt;</code>	The deployable that this deployed is derived from.
<b>placeholders</b> : <code>MAP_STRING_STRING</code>	A Map containing all the placeholders mapped to their values. Special values are <code>&lt;ignore&gt;</code> or <code>&lt;empty&gt;</code>
<b>targetFile</b> : <code>STRING</code>	Name of the artifact on the generic server.
Hidden Properties	
<b>* createOrder</b> : <code>INTEGER = 50</code>	Create Order
<b>* createVerb</b> : <code>STRING = Create</code>	Create Verb
<b>* destroyOrder</b> : <code>INTEGER = 40</code>	Destroy Order
<b>* destroyVerb</b> : <code>STRING = Destroy</code>	Destroy Verb
<b>* modifyOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <code>STRING = Modify</code>	Modify Verb
<b>* noopOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <code>STRING = Modify</code>	Noop Verb
<b>* targetDirectory</b> : <code>STRING</code> = <code>\${deployed.container.home}/server/\${deployed.container.serverName}/deploy</code>	Target Directory
<b>createTargetDirectory</b> : <code>BOOLEAN = false</code>	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <code>SET_OF_STRING</code>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <code>STRING</code>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <code>SET_OF_STRING</code>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>preserveExistingFiles</b> : <code>BOOLEAN = false</code>	if true, preserve the existing files on the remote host (do not delete the copied files during the destroy operation).
<b>restartRequired</b> : <code>BOOLEAN = true</code>	Restart Required
<b>restartRequiredForNoop</b> : <code>BOOLEAN = false</code>	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <code>BOOLEAN = true</code>	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.

## jbossas.BaseServer

### Virtual Type

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

**Interfaces** udm.Taggable, udm.ConfigurationItem, generic.GenericContainer, udm.Container, overthere.HostContainer

Base JBoss Application Server



Parent	
<b>* host</b> : <code>CI&lt;overthere.Host&gt;</code>	Host upon which the container resides
Public Properties	
<b>* ajpPort</b> : <code>INTEGER = 8009</code>	AJP connector port
<b>* controlPort</b> : <code>INTEGER = 1099</code>	Connector control port
<b>* home</b> : <code>STRING</code>	JBoss application server installation directory. e.g. /opt/jboss/5.1.0.GA
<b>* httpPort</b> : <code>INTEGER = 8080</code>	HTTP/1.1 connector port
<b>* serverName</b> : <code>STRING = default</code>	Server configuration name e.g. default or minimal
<b>envVars</b> : <code>MAP_STRING_STRING</code>	Environment variables for container
<b>tags</b> : <code>SET_OF_STRING</code>	If set, only deployables with the same tag will be automatically mapped to this container.
Hidden Properties	
<b>* inspectTemplate</b> : <code>STRING = jboss/container/inspect.sh.ftl</code>	Script (a freemarker template) used to inspect the server
<b>* jmsResourceFileSuffix</b> : <code>STRING = service</code>	Jms Resource File Suffix
<b>* queryTemplate</b> : <code>STRING = jboss/container/query.sh.ftl</code>	Script (a freemarker template) used to query a mbean valuf from the server
<b>* restartOrder</b> : <code>INTEGER = 90</code>	The order of the restart container step in the step list.
<b>* startOrder</b> : <code>INTEGER = 90</code>	The order of the start container step in the step list.
<b>* startScript</b> : <code>STRING = jboss/container/start.sh</code>	Script used to start the server
<b>* startWaitTime</b> : <code>INTEGER = 30</code>	Duration (in secs) to wait after the start server step has been executed
<b>* stopOrder</b> : <code>INTEGER = 10</code>	The order of the stop container step in the step list.
<b>* stopScript</b> : <code>STRING = jboss/container/stop.sh</code>	Script used to stop the server
<b>* stopWaitTime</b> : <code>INTEGER = 10</code>	Duration (in secs) to wait after the stop server step has been executed
<b>inspectClasspathResources</b> : <code>SET_OF_STRING</code>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <code>STRING</code>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <code>SET_OF_STRING</code>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartScript</b> : <code>STRING</code>	Classpath to the script used to restart the generic container.
<b>restartWaitTime</b> : <code>INTEGER = 0</code>	The time to wait in seconds for a container restart action.

## jbossas.Configuration

### Virtual Type

**Type Hierarchy** `jbossas.Artifact` >> `generic.CopiedArtifact` >>  
`generic.AbstractDeployedArtifact` >> `generic.AbstractDeployed` >>  
`udm.BaseDeployed` >> `udm.BaseConfigurationItem`

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

Base type for a JBoss configuration.

Parent	
<b>* container</b> : <code>CI&lt;udm.Container&gt;</code>	The container on which this deployed runs.
Public Properties	
<b>deployable</b> : <code>CI&lt;udm.Deployable&gt;</code>	The deployable that this deployed is derived from.
<b>placeholders</b> : <code>MAP_STRING_STRING</code>	A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>
<b>targetFile</b> : <code>STRING</code>	Name of the artifact on the generic server.
Hidden Properties	
<b>* createOrder</b> : <code>INTEGER = 50</code>	Create Order
<b>* createVerb</b> : <code>STRING = Create</code>	Create Verb
<b>* destroyOrder</b> : <code>INTEGER = 40</code>	Destroy Order
<b>* destroyVerb</b> : <code>STRING = Destroy</code>	Destroy Verb
<b>* modifyOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <code>STRING = Modify</code>	Modify Verb
<b>* noopOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <code>STRING = Modify</code>	Noop Verb
<b>* targetDirectory</b> : <code>STRING</code> = <code>\${deployed.container.home}/server/\${deployed.container.serverName}/conf</code>	Target Directory
<b>createTargetDirectory</b> : <code>BOOLEAN = false</code>	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <code>SET_OF_STRING</code>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <code>STRING</code>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <code>SET_OF_STRING</code>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>preserveExistingFiles</b> : <code>BOOLEAN = false</code>	if true, preserve the existing files on the remote host (do not delete the copied files during the destroy operation).
<b>restartRequired</b> : <code>BOOLEAN = true</code>	Restart Required
<b>restartRequiredForNoop</b> : <code>BOOLEAN = false</code>	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <code>BOOLEAN = true</code>	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.

## jbosscas.ConfigurationFile

<b>Type Hierarchy</b>	generic.File >> udm.BaseDeployableFileArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem
<b>Interfaces</b>	udm.Taggable, udm.Deployable, udm.SourceArtifact, udm.Artifact, udm.DeployableArtifact, udm.ConfigurationItem, udm.FileArtifact

File containing application configurations that is deployed to the server conf directory

Public Properties	
<b>excludeFileNamesRegex</b> :	<b>STRING</b> Regular expression that matches file names that must be excluded from scanning
<b>placeholders</b> :	<b>SET_OF_STRING</b> Placeholders detected in this artifact
<b>scanPlaceholders</b> :	<b>BOOLEAN = true</b> Whether to scan this artifact for placeholders when it is imported
<b>tags</b> :	<b>SET_OF_STRING</b> If set, this deployable will only be mapped automatically to containers with the same tag.
<b>targetFile</b> :	<b>STRING</b> Name of the artifact on the generic server. (string)
Hidden Properties	
<b>* textFileNamesRegex</b> :	<b>STRING = .+\. (cfg   conf   config   ini   properties   props   txt   asp   aspx   htm   html   jsf   jsp   xht   xhtml   sql   xml   xsd   xsl   xslt)</b> Regular expression that matches file names of text files

### jbossas.ConfigurationFolder

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

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

Folder containing files for application configuration that are deployed to the server conf directory

Public Properties	
<b>excludeFileNamesRegex</b> :	<b>STRING</b> Regular expression that matches file names that must be excluded from scanning
<b>placeholders</b> :	<b>SET_OF_STRING</b> Placeholders detected in this artifact
<b>scanPlaceholders</b> :	<b>BOOLEAN = true</b> Whether to scan this artifact for placeholders when it is imported
<b>tags</b> :	<b>SET_OF_STRING</b> If set, this deployable will only be mapped automatically to containers with the same tag.
<b>targetFile</b> :	<b>STRING</b> Name of the artifact on the generic server. (string)
Hidden Properties	
<b>* textFileNamesRegex</b> :	<b>STRING = .+\. (cfg   conf   config   ini   properties   props   txt   asp   aspx   htm   html   jsf   jsp   xht   xhtml   sql   xml   xsd   xsl   xslt)</b> Regular expression that matches file names of text files

### jbossas.Datasource

**Virtual Type** [jbossas.Resource](#)

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

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

Base type for a JBoss datasource

Parent	
<b>* container</b> :	<b>CI&lt;udm.Container&gt;</b> The container on which this deployed runs.

Public Properties	
<b>* jndiName</b> : <b>STRING</b>	Jndi Name
<b>* maxPoolSize</b> : <b>INTEGER = 20</b>	Max Pool Size
<b>* minPoolSize</b> : <b>INTEGER = 0</b>	Min Pool Size
<b>checkValidConnectionSql</b> : <b>STRING</b>	an sql statement that is executed before it is checked out from the pool to make sure it is still valid. If the sql fails, the connection is closed and new ones created.
<b>deployable</b> : <b>CI&lt;udm.Deployable&gt;</b>	The deployable that this deployed is derived from.
<b>exceptionSorterClassName</b> : <b>STRING</b>	a class that looks at vendor specific messages to determine whether sql errors are fatal and thus the connection should be destroyed. If none specified, no errors will be treated as fatal.
<b>password</b> : <b>STRING</b>	the password used when creating the connection (not used when security is configured)
<b>useJavaContext</b> : <b>BOOLEAN = true</b>	Use Java Context
<b>userName</b> : <b>STRING</b>	the user name used when creating the connection (not used when security is configured)

Hidden Properties	
<b>* createOrder</b> : <b>INTEGER</b> = 50	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* inspectTemplate</b> : <b>STRING</b> = jboss/datasource/inspect.sh.ftl	Inspect Template
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.home}/server/\${deployed.container.serverName}/deploy	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.deployable.name}-ds.xml	Target File
<b>* template</b> : <b>STRING</b> = jboss/datasource/template.xml.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartRequired</b> : <b>BOOLEAN</b> = true	Restart Required
<b>restartRequiredForNoop</b> : <b>BOOLEAN</b> = false	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <b>BOOLEAN</b> = 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.

## jbossas.DeployedConfigurationFile

**Type Hierarchy** [jbossas.Configuration](#) >> [jbossas.Artifact](#) >> [generic.CopiedArtifact](#) >> [generic.AbstractDeployedArtifact](#) >> [generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

**Interfaces** [udm.Artifact](#), [udm.Deployed](#), [udm.ConfigurationItem](#), [udm.DerivedArtifact](#)

Deployed configuration file

Parent	
<b>* container</b> : <b>CI</b> < <a href="#">udm.Container</a> >	The container on which this deployed runs.

Public Properties	
<b>deployable</b> : <a href="#">CI&lt;udm.Deployable&gt;</a>	The deployable that this deployed is derived from.
<b>placeholders</b> : <a href="#">MAP_STRING_STRING</a>	A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>
<b>targetFile</b> : <a href="#">STRING</a>	Name of the artifact on the generic server.
Hidden Properties	
* <b>createOrder</b> : <a href="#">INTEGER</a> = 50	Create Order
* <b>createVerb</b> : <a href="#">STRING</a> = Create	Create Verb
* <b>destroyOrder</b> : <a href="#">INTEGER</a> = 40	Destroy Order
* <b>destroyVerb</b> : <a href="#">STRING</a> = Destroy	Destroy Verb
* <b>modifyOrder</b> : <a href="#">INTEGER</a> = 50	The order of the step in the step list for the modify operation.
* <b>modifyVerb</b> : <a href="#">STRING</a> = Modify	Modify Verb
* <b>noopOrder</b> : <a href="#">INTEGER</a> = 50	The order of the step in the step list for the noop operation.
* <b>noopVerb</b> : <a href="#">STRING</a> = Modify	Noop Verb
* <b>targetDirectory</b> : <a href="#">STRING</a> = <a href="#">\${deployed.container.home}/server/\${deployed.container.serverName}/conf</a>	Target Directory
<b>createTargetDirectory</b> : <a href="#">BOOLEAN</a> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <a href="#">SET_OF_STRING</a>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <a href="#">STRING</a>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <a href="#">SET_OF_STRING</a>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>preserveExistingFiles</b> : <a href="#">BOOLEAN</a> = false	if true, preserve the existing files on the remote host (do not delete the copied files during the destroy operation).
<b>restartRequired</b> : <a href="#">BOOLEAN</a> = true	Restart Required
<b>restartRequiredForNoop</b> : <a href="#">BOOLEAN</a> = false	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <a href="#">BOOLEAN</a> = 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.

## jbossas.DeployedConfigurationFolder

Type Hierarchy	<a href="#">jbossas.Configuration</a> >> <a href="#">jbossas.Artifact</a> >> <a href="#">generic.CopiedArtifact</a> >> <a href="#">generic.AbstractDeployedArtifact</a> >> <a href="#">generic.AbstractDeployed</a> >> <a href="#">udm.BaseDeployed</a> >> <a href="#">udm.BaseConfigurationItem</a>
Interfaces	<a href="#">udm.Artifact</a> , <a href="#">udm.Deployed</a> , <a href="#">udm.ConfigurationItem</a> , <a href="#">udm.DerivedArtifact</a>

Deployed configuration folder

Parent	
* <b>container</b> : <a href="#">CI&lt;udm.Container&gt;</a>	The container on which this deployed runs.

Public Properties	
<b>deployable</b> : <code>CI&lt;udm.Deployable&gt;</code>	The deployable that this deployed is derived from.
<b>placeholders</b> : <code>MAP_STRING_STRING</code>	A Map containing all the placeholders mapped to their values. Special values are <code>&lt;ignore&gt;</code> or <code>&lt;empty&gt;</code>
<b>targetFile</b> : <code>STRING</code>	Name of the artifact on the generic server.
Hidden Properties	
* <b>createOrder</b> : <code>INTEGER = 50</code>	Create Order
* <b>createVerb</b> : <code>STRING = Create</code>	Create Verb
* <b>destroyOrder</b> : <code>INTEGER = 40</code>	Destroy Order
* <b>destroyVerb</b> : <code>STRING = Destroy</code>	Destroy Verb
* <b>modifyOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the modify operation.
* <b>modifyVerb</b> : <code>STRING = Modify</code>	Modify Verb
* <b>noopOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the noop operation.
* <b>noopVerb</b> : <code>STRING = Modify</code>	Noop Verb
* <b>targetDirectory</b> : <code>STRING</code> = <code>\${deployed.container.home}/server/\${deployed.container.serverName}/conf</code>	Target Directory
<b>createTargetDirectory</b> : <code>BOOLEAN = false</code>	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <code>SET_OF_STRING</code>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <code>STRING</code>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <code>SET_OF_STRING</code>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>preserveExistingFiles</b> : <code>BOOLEAN = false</code>	if true, preserve the existing files on the remote host (do not delete the copied files during the destroy operation).
<b>restartRequired</b> : <code>BOOLEAN = true</code>	Restart Required
<b>restartRequiredForNoop</b> : <code>BOOLEAN = false</code>	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <code>BOOLEAN = true</code>	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.

## jbossas.DeployedLibrary

<b>Type Hierarchy</b>	<code>jbossas.Artifact</code> >> <code>generic.CopiedArtifact</code> >> <code>generic.AbstractDeployedArtifact</code> >> <code>generic.AbstractDeployed</code> >> <code>udm.BaseDeployed</code> >> <code>udm.BaseConfigurationItem</code>
<b>Interfaces</b>	<code>udm.Artifact</code> , <code>udm.Deployed</code> , <code>udm.ConfigurationItem</code> , <code>udm.DerivedArtifact</code>

The Library as deployed on the jboss server.

Parent	
* <b>container</b> : <code>CI&lt;udm.Container&gt;</code>	The container on which this deployed runs.

Public Properties	
<b>deployable</b> : <code>CI&lt;udm.Deployable&gt;</code>	The deployable that this deployed is derived from.
<b>placeholders</b> : <code>MAP_STRING_STRING</code>	A Map containing all the placeholders mapped to their values. Special values are <code>&lt;ignore&gt;</code> or <code>&lt;empty&gt;</code>
<b>targetFile</b> : <code>STRING</code>	Name of the artifact on the generic server.
Hidden Properties	
* <b>createOrder</b> : <code>INTEGER = 50</code>	Create Order
* <b>createVerb</b> : <code>STRING = Create</code>	Create Verb
* <b>destroyOrder</b> : <code>INTEGER = 40</code>	Destroy Order
* <b>destroyVerb</b> : <code>STRING = Destroy</code>	Destroy Verb
* <b>modifyOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the modify operation.
* <b>modifyVerb</b> : <code>STRING = Modify</code>	Modify Verb
* <b>noopOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the noop operation.
* <b>noopVerb</b> : <code>STRING = Modify</code>	Noop Verb
* <b>targetDirectory</b> : <code>STRING</code> = <code>\${deployed.container.home}/server/\${deployed.container.serverName}/lib</code>	Target Directory
<b>createTargetDirectory</b> : <code>BOOLEAN = false</code>	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <code>SET_OF_STRING</code>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <code>STRING</code>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <code>SET_OF_STRING</code>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>preserveExistingFiles</b> : <code>BOOLEAN = false</code>	if true, preserve the existing files on the remote host (do not delete the copied files during the destroy operation).
<b>restartRequired</b> : <code>BOOLEAN = true</code>	Restart Required
<b>restartRequiredForNoop</b> : <code>BOOLEAN = false</code>	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <code>BOOLEAN = true</code>	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.

## jbossas.Ear

<b>Type Hierarchy</b>	jee.Ear >> udm.BaseDeployableArchiveArtifact >> udm.BaseDeployableFileArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem
<b>Interfaces</b>	udm.Taggable, udm.Deployable, udm.SourceArtifact, udm.ArchiveArtifact, udm.Artifact, udm.DeployableArtifact, udm.ConfigurationItem, udm.FileArtifact

A JEE EAR archive



Public Properties	
<b>excludeFileNamesRegex</b> :	<b>STRING</b> Regular expression that matches file names that must be excluded from scanning
<b>placeholders</b> :	<b>SET_OF_STRING</b> Placeholders detected in this artifact
<b>scanPlaceholders</b> :	<b>BOOLEAN</b> = false Whether to scan this artifact for placeholders when it is imported
<b>tags</b> :	<b>SET_OF_STRING</b> If set, this deployable will only be mapped automatically to containers with the same tag.
Hidden Properties	
<b>* textFileNamesRegex</b> :	<b>STRING</b> = .+\. (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

## jbossas.EarModule

**Type Hierarchy** [jbossas.Artifact](#) >> 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

Parent	
<b>* container</b> :	<b>CI</b> <udm.Container> The container on which this deployed runs.
Public Properties	
<b>deployable</b> :	<b>CI</b> <udm.Deployable> The deployable that this deployed is derived from.
<b>placeholders</b> :	<b>MAP_STRING_STRING</b> A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>

Hidden Properties	
<b>* createOrder</b> : <b>INTEGER</b> = 50	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.home}/server/\${deployed.container.serverName}/deploy	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.deployable.name}.ear	Target File
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>preserveExistingFiles</b> : <b>BOOLEAN</b> = false	if true, preserve the existing files on the remote host (do not delete the copied files during the destroy operation).
<b>restartRequired</b> : <b>BOOLEAN</b> = true	Restart Required
<b>restartRequiredForNoop</b> : <b>BOOLEAN</b> = false	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <b>BOOLEAN</b> = 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.

## jbossas.Library

<b>Type Hierarchy</b>	generic.File >> udm.BaseDeployableFileArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem
<b>Interfaces</b>	udm.Taggable, udm.Deployable, udm.SourceArtifact, udm.Artifact, udm.DeployableArtifact, udm.ConfigurationItem, udm.FileArtifact

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

Public Properties	
<b>excludeFileNamesRegex</b> : <b>STRING</b>	Regular expression that matches file names that must be excluded from scanning
<b>placeholders</b> : <b>SET_OF_STRING</b>	Placeholders detected in this artifact
<b>scanPlaceholders</b> : <b>BOOLEAN</b> = true	Whether to scan this artifact for placeholders when it is imported
<b>tags</b> : <b>SET_OF_STRING</b>	If set, this deployable will only be mapped automatically to containers with the same tag.
<b>targetFile</b> : <b>STRING</b>	Name of the artifact on the generic server. (string)
Hidden Properties	
* <b>textFileNamesRegex</b> : <b>STRING</b> = .+\. (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

## jbossas.NonTransactionalDataSource

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

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

A JBoss No Transaction DataSource

Parent	
* <b>container</b> : <b>CI</b> <udm.Container>	The container on which this deployed runs.
Public Properties	
* <b>connectionUrl</b> : <b>STRING</b>	the JDBC driver connection url string
* <b>driverClass</b> : <b>STRING</b>	the JDBC driver class implementing java.sql.Driver
* <b>jndiName</b> : <b>STRING</b>	Jndi Name
* <b>maxPoolSize</b> : <b>INTEGER</b> = 20	Max Pool Size
* <b>minPoolSize</b> : <b>INTEGER</b> = 0	Min Pool Size
<b>checkValidConnectionSql</b> : <b>STRING</b>	an sql statement that is executed before it is checked out from the pool to make sure it is still valid. If the sql fails, the connection is closed and new ones created.
<b>connectionProperties</b> : <b>MAP_STRING_STRING</b>	used to configure the connections retrieved from the java.sql.Driver
<b>deployable</b> : <b>CI</b> <udm.Deployable>	The deployable that this deployed is derived from.
<b>exceptionSorterClassName</b> : <b>STRING</b>	a class that looks at vendor specific messages to determine whether sql errors are fatal and thus the connection should be destroyed. If none specified, no errors will be treated as fatal.
<b>password</b> : <b>STRING</b>	the password used when creating the connection (not used when security is configured)
<b>useJavaContext</b> : <b>BOOLEAN</b> = true	Use Java Context
<b>userName</b> : <b>STRING</b>	the user name used when creating the connection (not used when security is configured)

Hidden Properties	
<b>* createOrder</b> : <b>INTEGER</b> = 50	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* dsType</b> : <b>STRING</b> = no-tx-datasource	Ds Type
<b>* inspectTemplate</b> : <b>STRING</b> = jboss/datasource/inspect.sh.ftl	Inspect Template
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.home}/server/\${deployed.container.serverName}/deploy	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.deployable.name}-ds.xml	Target File
<b>* template</b> : <b>STRING</b> = jboss/datasource/template.xml.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartRequired</b> : <b>BOOLEAN</b> = true	Restart Required
<b>restartRequiredForNoop</b> : <b>BOOLEAN</b> = false	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <b>BOOLEAN</b> = 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.

### jbossas.NonTransactionalDatasourceSpec

**Type Hierarchy** jee.DataSourceSpec >> jee.JndiResourceSpec >> jee.ResourceSpec >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

A JBoss No Transaction Datasource (deployable)

Public Properties	
<b>checkValidConnectionSql</b> : <b>STRING</b>	an sql statement that is executed before it is checked out from the pool to make sure it is still valid. If the sql fails, the connection is closed and new ones created. (string)
<b>connectionProperties</b> : <b>MAP_STRING_STRING</b>	used to configure the connections retrieved from the java.sql.Driver (map_string_string)
<b>connectionUrl</b> : <b>STRING</b>	the JDBC driver connection url string (string)
<b>driverClass</b> : <b>STRING</b>	the JDBC driver class implementing java.sql.Driver (string)
<b>exceptionSorterClassName</b> : <b>STRING</b>	a class that looks at vendor specific messages to determine whether sql errors are fatal and thus the connection should be destroyed. If none specified, no errors will be treated as fatal. (string)
<b>jndiName</b> : <b>STRING</b>	Jndi Name (string)
<b>maxPoolSize</b> : <b>STRING</b>	Max Pool Size (integer)
<b>minPoolSize</b> : <b>STRING</b>	Min Pool Size (integer)
<b>password</b> : <b>STRING</b>	the password used when creating the connection (not used when security is configured) (string)
<b>tags</b> : <b>SET_OF_STRING</b>	If set, this deployable will only be mapped automatically to containers with the same tag.
<b>useJavaContext</b> : <b>STRING</b>	Use Java Context (boolean)
<b>userName</b> : <b>STRING</b>	the user name used when creating the connection (not used when security is configured) (string)

## jbossas.Queue

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

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

A JBoss Queue

Parent	
* <b>container</b> : <b>CI&lt;udm.Container&gt;</b>	The container on which this deployed runs.
Public Properties	
* <b>jndiName</b> : <b>STRING</b>	Jndi Name
* <b>maxDepth</b> : <b>INTEGER = 0</b>	Max Depth
<b>deployable</b> : <b>CI&lt;udm.Deployable&gt;</b>	The deployable that this deployed is derived from.

Hidden Properties	
<b>* createOrder</b> : <b>INTEGER</b> = 50	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* inspectTemplate</b> : <b>STRING</b> = jboss/queue/inspect.sh.ftl	Inspect Template
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.home}/server/\${deployed.container.serverName}/deploy	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.deployable.name}- \${deployed.container.jmsResourceFileSuffix}.xml	Target File
<b>* template</b> : <b>STRING</b> = jboss/queue/template\${deployed.container.version}.xml.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartRequired</b> : <b>BOOLEAN</b> = true	Restart Required
<b>restartRequiredForNoop</b> : <b>BOOLEAN</b> = false	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <b>BOOLEAN</b> = 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.

## jbossas.QueueSpec

**Type Hierarchy** jee.QueueSpec >> jee.JndiResourceSpec >> jee.ResourceSpec >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification for a JBoss queue

Public Properties	
<b>jndiName</b> : <b>STRING</b>	Jndi Name (string)
<b>maxDepth</b> : <b>STRING</b>	Max Depth (integer)
<b>tags</b> : <b>SET_OF_STRING</b>	If set, this deployable will only be mapped automatically to containers with the same tag.

**jbossas.Resource****Virtual Type**

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

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

Description unavailable

Parent	
<b>* container</b> : <b>CI</b> <udm.Container>	The container on which this deployed runs.
Public Properties	
<b>deployable</b> : <b>CI</b> <udm.Deployable>	The deployable that this deployed is derived from.
<b>targetFile</b> : <b>STRING</b>	Name of the artifact on the generic server.
Hidden Properties	
<b>* createOrder</b> : <b>INTEGER</b> = 50	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.home}/server/\${deployed.container.serverName}/deploy	Target Directory
<b>* template</b> : <b>STRING</b>	Classpath to the freemarker template used to generate the content of the final text base artifact.
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartRequired</b> : <b>BOOLEAN</b> = true	Restart Required
<b>restartRequiredForNoop</b> : <b>BOOLEAN</b> = false	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <b>BOOLEAN</b> = 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.

**jbossas.ServerV4**

**Type Hierarchy** `jboszas.BaseServer` >> `generic.Container` >> `udm.BaseContainer` >> `udm.BaseConfigurationItem`

**Interfaces** `udm.Taggable`, `udm.ConfigurationItem`, `udm.Container`, `generic.GenericContainer`, `overthere.HostContainer`

JBoss 4 application server

Parent	
<b>* host</b> : <code>CI&lt;overthere.Host&gt;</code>	Host upon which the container resides
Public Properties	
<b>* ajpPort</b> : <code>INTEGER</code> = 8009	AJP connector port
<b>* controlPort</b> : <code>INTEGER</code> = 1099	Connector control port
<b>* home</b> : <code>STRING</code>	JBoss application server installation directory. e.g. <code>/opt/jboss/5.1.0.GA</code>
<b>* httpPort</b> : <code>INTEGER</code> = 8080	HTTP/1.1 connector port
<b>* serverName</b> : <code>STRING</code> = default	Server configuration name e.g. default or minimal
<b>envVars</b> : <code>MAP_STRING_STRING</code>	Environment variables for container
<b>tags</b> : <code>SET_OF_STRING</code>	If set, only deployables with the same tag will be automatically mapped to this container.
Hidden Properties	
<b>* inspectTemplate</b> : <code>STRING</code> = <code>jboss/container/inspect.sh.ftl</code>	Script (a freemarker template) used to inspect the server
<b>* jmsResourceFileSuffix</b> : <code>STRING</code> = <code>service</code>	Jms Resource File Suffix
<b>* queryTemplate</b> : <code>STRING</code> = <code>jboss/container/query.sh.ftl</code>	Script (a freemarker template) used to query a mbean valuf from the server
<b>* restartOrder</b> : <code>INTEGER</code> = 90	The order of the restart container step in the step list.
<b>* startOrder</b> : <code>INTEGER</code> = 90	The order of the start container step in the step list.
<b>* startScript</b> : <code>STRING</code> = <code>jboss/container/start.sh</code>	Script used to start the server
<b>* startWaitTime</b> : <code>INTEGER</code> = 30	Duration (in secs) to wait after the start server step has been executed
<b>* stopOrder</b> : <code>INTEGER</code> = 10	The order of the stop container step in the step list.
<b>* stopScript</b> : <code>STRING</code> = <code>jboss/container/stop.sh</code>	Script used to stop the server
<b>* stopWaitTime</b> : <code>INTEGER</code> = 10	Duration (in secs) to wait after the stop server step has been executed
<b>* version</b> : <code>INTEGER</code> = 4	Version
<b>inspectClasspathResources</b> : <code>SET_OF_STRING</code>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <code>STRING</code>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <code>SET_OF_STRING</code>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartScript</b> : <code>STRING</code>	Classpath to the script used to restart the generic container.
<b>restartWaitTime</b> : <code>INTEGER</code> = 0	The time to wait in seconds for a container restart action.




## jbossas.ServerV5

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

**Interfaces** udm.Taggable, udm.ConfigurationItem, udm.Container, generic.GenericContainer, overthere.HostContainer

Jboss 5 application server

Parent
 <b>host</b> : <a href="#">CI&lt;overthere.Host&gt;</a> Host upon which the container resides

Hidden Properties	
<b>* inspectTemplate</b> : <b>STRING</b> = <b>jboss/container/inspect.sh.ftl</b>	Script (a freemarker template) used to inspect the server
<b>* jmsResourceFileSuffix</b> : <b>STRING</b> = <b>service</b>	Jms Resource File Suffix
<b>* queryTemplate</b> : <b>STRING</b> = <b>jboss/container/query.sh.ftl</b>	Script (a freemarker template) used to query a mbean valuf from the server
<b>* restartOrder</b> : <b>INTEGER</b> = <b>90</b>	The order of the restart container step in the step list.
<b>* startOrder</b> : <b>INTEGER</b> = <b>90</b>	The order of the start container step in the step list.
<b>* startScript</b> : <b>STRING</b> = <b>jboss/container/start.sh</b>	Script used to start the server
<b>* startWaitTime</b> : <b>INTEGER</b> = <b>30</b>	Duration (in secs) to wait after the start server step has been executed
<b>* stopOrder</b> : <b>INTEGER</b> = <b>10</b>	The order of the stop container step in the step list.
<b>* stopScript</b> : <b>STRING</b> = <b>jboss/container/stop.sh</b>	Script used to stop the server
<b>* stopWaitTime</b> : <b>INTEGER</b> = <b>10</b>	Duration (in secs) to wait after the stop server step has been executed
<b>* version</b> : <b>INTEGER</b> = <b>5</b>	Version
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartScript</b> : <b>STRING</b>	Classpath to the script used to restart the generic container.
<b>restartWaitTime</b> : <b>INTEGER</b> = <b>0</b>	The time to wait in seconds for a container restart action.

## jbossas.ServerV6

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

**Interfaces** [udm.Taggable](#), [udm.ConfigurationItem](#), [udm.Container](#), [generic.GenericContainer](#), [overthere.HostContainer](#)

Jboss 6 application server

Parent	
<b>* host</b> : <b>CI</b> < <a href="#">overthere.Host</a> >	Host upon which the container resides

Public Properties	
<b>* ajpPort</b> : <b>INTEGER</b> = 8009	AJP connector port
<b>* controlPort</b> : <b>INTEGER</b> = 1090	Connector control port
<b>* home</b> : <b>STRING</b>	JBoss application server installation directory. e.g. /opt/jboss/5.1.0.GA
<b>* httpPort</b> : <b>INTEGER</b> = 8080	HTTP/1.1 connector port
<b>* serverName</b> : <b>STRING</b> = default	Server configuration name e.g. default or minimal
<b>envVars</b> : <b>MAP_STRING_STRING</b>	Environment variables for container
<b>tags</b> : <b>SET_OF_STRING</b>	If set, only deployables with the same tag will be automatically mapped to this container.
Hidden Properties	
<b>* inspectTemplate</b> : <b>STRING</b> = jboss/container/inspect.sh.ftl	Script (a freemarker template) used to inspect the server
<b>* jmsResourceFileSuffix</b> : <b>STRING</b> = hometq-jms	Jms Resource File Suffix
<b>* queryTemplate</b> : <b>STRING</b> = jboss/container/query.sh.ftl	Script (a freemarker template) used to query a mbean valuf from the server
<b>* restartOrder</b> : <b>INTEGER</b> = 90	The order of the restart container step in the step list.
<b>* startOrder</b> : <b>INTEGER</b> = 90	The order of the start container step in the step list.
<b>* startScript</b> : <b>STRING</b> = jboss/container/start.sh	Script used to start the server
<b>* startWaitTime</b> : <b>INTEGER</b> = 30	Duration (in secs) to wait after the start server step has been executed
<b>* stopOrder</b> : <b>INTEGER</b> = 10	The order of the stop container step in the step list.
<b>* stopScript</b> : <b>STRING</b> = jboss/container/stop.sh	Script used to stop the server
<b>* stopWaitTime</b> : <b>INTEGER</b> = 10	Duration (in secs) to wait after the stop server step has been executed
<b>* version</b> : <b>INTEGER</b> = 6	Version
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartScript</b> : <b>STRING</b>	Classpath to the script used to restart the generic container.
<b>restartWaitTime</b> : <b>INTEGER</b> = 0	The time to wait in seconds for a container restart action.

## jbossas.Topic

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

**Interfaces** [udm.Deployed](#), [udm.ConfigurationItem](#)

A JBoss topic

Parent	
<b>* container</b> : <code>CI&lt;udm.Container&gt;</code>	The container on which this deployed runs.
Public Properties	
<b>* jndiName</b> : <code>STRING</code>	Jndi Name
<b>* maxDepth</b> : <code>INTEGER = 0</code>	Max Depth
<b>deployable</b> : <code>CI&lt;udm.Deployable&gt;</code>	The deployable that this deployed is derived from.
Hidden Properties	
<b>* createOrder</b> : <code>INTEGER = 50</code>	Create Order
<b>* createVerb</b> : <code>STRING = Create</code>	Create Verb
<b>* destroyOrder</b> : <code>INTEGER = 40</code>	Destroy Order
<b>* destroyVerb</b> : <code>STRING = Destroy</code>	Destroy Verb
<b>* inspectTemplate</b> : <code>STRING = jboss/topic/inspect.sh.ftl</code>	Inspect Template
<b>* modifyOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <code>STRING = Modify</code>	Modify Verb
<b>* noopOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <code>STRING = Modify</code>	Noop Verb
<b>* targetDirectory</b> : <code>STRING</code> = <code>\${deployed.container.home}/server/\${deployed.container.serverName}/deploy</code>	Target Directory
<b>* targetFile</b> : <code>STRING = \${deployed.deployable.name}-\${deployed.container.jmsResourceFileSuffix}.xml</code>	Target File
<b>* template</b> : <code>STRING = jboss/topic/template\${deployed.container.version}.xml.ftl</code>	Template
<b>createTargetDirectory</b> : <code>BOOLEAN = false</code>	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <code>SET_OF_STRING</code>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <code>STRING</code>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <code>SET_OF_STRING</code>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartRequired</b> : <code>BOOLEAN = true</code>	Restart Required
<b>restartRequiredForNoop</b> : <code>BOOLEAN = false</code>	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <code>BOOLEAN = true</code>	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.

## jbossas.TopicSpec

**Type Hierarchy** `jee.TopicSpec >> jee.JndiResourceSpec >> jee.ResourceSpec >> udm.BaseDeployable >> udm.BaseConfigurationItem`

**Interfaces** `udm.Taggable, udm.Deployable, udm.ConfigurationItem`

Specification for a JBoss topic

Public Properties	
<b>jndiName</b> :	<b>STRING</b> Jndi Name (string)
<b>maxDepth</b> :	<b>STRING</b> Max Depth (integer)
<b>tags</b> :	<b>SET_OF_STRING</b> If set, this deployable will only be mapped automatically to containers with the same tag.

### **jbossas.TransactionDataSource**

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

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

A JBoss Local Transaction Datasource

Parent	
* <b>container</b> :	<b>CI&lt;udm.Container&gt;</b> The container on which this deployed runs.

Hidden Properties	
<b>* createOrder</b> : <b>INTEGER</b> = 50	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* dsType</b> : <b>STRING</b> = local-tx-datasource	Ds Type
<b>* inspectTemplate</b> : <b>STRING</b> = jboss/datasource/inspect.sh.ftl	Inspect Template
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.home}/server/\${deployed.container.serverName}/deploy	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.deployable.name}-ds.xml	Target File
<b>* template</b> : <b>STRING</b> = jboss/datasource/template.xml.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartRequired</b> : <b>BOOLEAN</b> = true	Restart Required
<b>restartRequiredForNoop</b> : <b>BOOLEAN</b> = false	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <b>BOOLEAN</b> = 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.

## jbossas.TransactionDataSourceSpec

**Type Hierarchy** jee.DataSourceSpec >> jee.JndiResourceSpec >> jee.ResourceSpec >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

A JBoss Local Transaction Datasource (deployable)

Public Properties	
<b>checkValidConnectionSql</b> : <b>STRING</b>	an sql statement that is executed before it is checked out from the pool to make sure it is still valid. If the sql fails, the connection is closed and new ones created. (string)
<b>connectionProperties</b> : <b>MAP_STRING_STRING</b>	used to configure the connections retrieved from the java.sql.Driver (map_string_string)
<b>connectionUrl</b> : <b>STRING</b>	the JDBC driver connection url string (string)
<b>driverClass</b> : <b>STRING</b>	the JDBC driver class implementing java.sql.Driver (string)
<b>exceptionSorterClassName</b> : <b>STRING</b>	a class that looks at vendor specific messages to determine whether sql errors are fatal and thus the connection should be destroyed. If none specified, no errors will be treated as fatal. (string)
<b>jndiName</b> : <b>STRING</b>	Jndi Name (string)
<b>maxPoolSize</b> : <b>STRING</b>	Max Pool Size (integer)
<b>minPoolSize</b> : <b>STRING</b>	Min Pool Size (integer)
<b>password</b> : <b>STRING</b>	the password used when creating the connection (not used when security is configured) (string)
<b>tags</b> : <b>SET_OF_STRING</b>	If set, this deployable will only be mapped automatically to containers with the same tag.
<b>useJavaContext</b> : <b>STRING</b>	Use Java Context (boolean)
<b>userName</b> : <b>STRING</b>	the user name used when creating the connection (not used when security is configured) (string)

### jbossas.TransactionXADatasource

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

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

A JBoss XA Datasource

Parent	
* <b>container</b> : <b>CI&lt;udm.Container&gt;</b>	The container on which this deployed runs.

Public Properties	
<b>* jndiName</b> : <b>STRING</b>	Jndi Name
<b>* maxPoolSize</b> : <b>INTEGER</b> = 20	Max Pool Size
<b>* minPoolSize</b> : <b>INTEGER</b> = 0	Min Pool Size
<b>* xaDataSourceClass</b> : <b>STRING</b>	the class implementing the XADataSource
<b>checkValidConnectionSql</b> : <b>STRING</b>	an sql statement that is executed before it is checked out from the pool to make sure it is still valid. If the sql fails, the connection is closed and new ones created.
<b>deployable</b> : <b>CI&lt;udm.Deployable&gt;</b>	The deployable that this deployed is derived from.
<b>exceptionSorterClassName</b> : <b>STRING</b>	a class that looks at vendor specific messages to determine whether sql errors are fatal and thus the connection should be destroyed. If none specified, no errors will be treated as fatal.
<b>password</b> : <b>STRING</b>	the password used when creating the connection (not used when security is configured)
<b>trackConnectionByTx</b> : <b>BOOLEAN</b> = true	set to fix problems with Oracle (not necessarily in JBoss-5.x where it is enabled by default and the element is deprecated)
<b>useJavaContext</b> : <b>BOOLEAN</b> = true	Use Java Context
<b>userName</b> : <b>STRING</b>	the user name used when creating the connection (not used when security is configured)
<b>xaDataSourceProperties</b> : <b>MAP_STRING_STRING</b>	properties used to configure the XADataSource



Hidden Properties	
<b>* createOrder</b> : <b>INTEGER</b> = 50	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* dsType</b> : <b>STRING</b> = xa-datasource	Ds Type
<b>* inspectTemplate</b> : <b>STRING</b> = jboss/datasource/inspect.sh.ftl	Inspect Template
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.home}/server/\${deployed.container.serverName}/deploy	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.deployable.name}-ds.xml	Target File
<b>* template</b> : <b>STRING</b> = jboss/datasource/template.xml.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartRequired</b> : <b>BOOLEAN</b> = true	Restart Required
<b>restartRequiredForNoop</b> : <b>BOOLEAN</b> = false	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <b>BOOLEAN</b> = 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.

### jbossas.TransactionaIXADatasourceSpec

**Type Hierarchy**    jee.DataSourceSpec >> jee.JndiResourceSpec >> jee.ResourceSpec >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

A JBoss XA Datasource (deployable)

Public Properties	
<b>checkValidConnectionSql</b> :	<b>STRING</b> an sql statement that is executed before it is checked out from the pool to make sure it is still valid. If the sql fails, the connection is closed and new ones created. (string)
<b>exceptionSorterClassName</b> :	<b>STRING</b> a class that looks at vendor specific messages to determine whether sql errors are fatal and thus the connection should be destroyed. If none specified, no errors will be treated as fatal. (string)
<b>jndiName</b> :	<b>STRING</b> Jndi Name (string)
<b>maxPoolSize</b> :	<b>STRING</b> Max Pool Size (integer)
<b>minPoolSize</b> :	<b>STRING</b> Min Pool Size (integer)
<b>password</b> :	<b>STRING</b> the password used when creating the connection (not used when security is configured) (string)
<b>tags</b> :	<b>SET_OF_STRING</b> If set, this deployable will only be mapped automatically to containers with the same tag.
<b>trackConnectionByTx</b> :	<b>STRING</b> set to fix problems with Oracle (not necessarily in JBoss-5.x where it is enabled by default and the element is deprecated) (boolean)
<b>useJavaContext</b> :	<b>STRING</b> Use Java Context (boolean)
<b>userName</b> :	<b>STRING</b> the user name used when creating the connection (not used when security is configured) (string)
<b>xaDataSourceClass</b> :	<b>STRING</b> the class implementing the XADataSource (string)
<b>xaDataSourceProperties</b> :	<b>MAP_STRING_STRING</b> properties used to configure the XADataSource (map_string_string)

## jbossas.War

<b>Type Hierarchy</b>	jee.War >> udm.BaseDeployableArchiveArtifact >> udm.BaseDeployableFileArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem
<b>Interfaces</b>	udm.Taggable, udm.Deployable, udm.SourceArtifact, udm.ArchiveArtifact, udm.Artifact, udm.DeployableArtifact, udm.ConfigurationItem, udm.FileArtifact

A JEE WAR archive

Public Properties	
<b>excludeFileNamesRegex</b> :	<b>STRING</b> Regular expression that matches file names that must be excluded from scanning
<b>placeholders</b> :	<b>SET_OF_STRING</b> Placeholders detected in this artifact
<b>scanPlaceholders</b> :	<b>BOOLEAN = false</b> Whether to scan this artifact for placeholders when it is imported
<b>tags</b> :	<b>SET_OF_STRING</b> If set, this deployable will only be mapped automatically to containers with the same tag.
Hidden Properties	
<b>* textFileNamesRegex</b> :	<b>STRING = .+\. (cfg   conf   config   ini   properties   props   txt   asp   aspx   htm   html   jsf   jsp   xht   xhtml   sql   xml   xsd   xsl   xslt)</b> Regular expression that matches file names of text files

## jbossas.WarModule

<b>Type Hierarchy</b>	<b>jbossas.Artifact</b> >> generic.CopiedArtifact >> generic.AbstractDeployedArtifact >> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem
<b>Interfaces</b>	udm.Artifact, udm.Deployed, udm.ConfigurationItem, udm.DerivedArtifact

War with values configured for a deployment

Parent	
<b>* container</b> : <code>CI&lt;udm.Container&gt;</code>	The container on which this deployed runs.
Public Properties	
<b>deployable</b> : <code>CI&lt;udm.Deployable&gt;</code>	The deployable that this deployed is derived from.
<b>placeholders</b> : <code>MAP_STRING_STRING</code>	A Map containing all the placeholders mapped to their values. Special values are <code>&lt;ignore&gt;</code> or <code>&lt;empty&gt;</code>
Hidden Properties	
<b>* createOrder</b> : <code>INTEGER = 50</code>	Create Order
<b>* createVerb</b> : <code>STRING = Create</code>	Create Verb
<b>* destroyOrder</b> : <code>INTEGER = 40</code>	Destroy Order
<b>* destroyVerb</b> : <code>STRING = Destroy</code>	Destroy Verb
<b>* modifyOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <code>STRING = Modify</code>	Modify Verb
<b>* noopOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <code>STRING = Modify</code>	Noop Verb
<b>* targetDirectory</b> : <code>STRING</code> = <code>\${deployed.container.home}/server/\${deployed.container.serverName}/deploy</code>	Target Directory
<b>* targetFile</b> : <code>STRING = \${deployed.deployable.name}.war</code>	Target File
<b>createTargetDirectory</b> : <code>BOOLEAN = false</code>	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <code>SET_OF_STRING</code>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <code>STRING</code>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <code>SET_OF_STRING</code>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>preserveExistingFiles</b> : <code>BOOLEAN = false</code>	if true, preserve the existing files on the remote host (do not delete the copied files during the destroy operation).
<b>restartRequired</b> : <code>BOOLEAN = true</code>	Restart Required
<b>restartRequiredForNoop</b> : <code>BOOLEAN = false</code>	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <code>BOOLEAN = true</code>	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.