

Deployit JBoss Application Server Plugin Manual

Version 3.8.1

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	10
jbossas.ConfigurationFile	11
jbossas.ConfigurationFolder	12
jbossas.Datasource	12
jbossas.DeployedConfigurationFile	14
jbossas.DeployedConfigurationFolder	15
jbossas.DeployedLibrary	16
jbossas.Ear	17
jbossas.EarModule	18
jbossas.Library	19
jbossas.NonTransactionalDatasource	20
jbossas.NonTransactionalDatasourceSpec	21
jbossas.Queue	22
jbossas.QueueSpec	23
jbossas.Resource	24
jbossas.ServerV4	24
jbossas.ServerV5	26
jbossas.ServerV6	28
jbossas.Topic	30
jbossas.TopicSpec	31
jbossas.TransactionalDatasource	32
jbossas.TransactionalDatasourceSpec	33
jbossas.TransactionalXADatasource	34
jbossas.TransactionalXADatasourceSpec	36
jbossas.War	37
jbossas.WarModule	37

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/hometq/hometq-jms.xml.

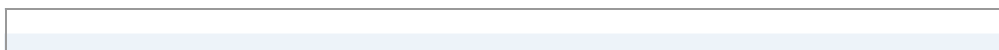
Requirements

- **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.
 - JBoss installed as a service when running on Windows. See the [JBoss Native](#) guide for one possible method of installing JBoss as a service.

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"> stop Server deploy application start Server 	<ul style="list-style-type: none"> stop Server undeploy old application version start Server 	<ul style="list-style-type: none"> stop Server undeploy old application version deploy new application version start Server
jbossas.TransactionDataSource jbossas.TransactionXADatasource jbossas.NonTransactionalDataSource	<ul style="list-style-type: none"> create datasource 	<ul style="list-style-type: none"> destroy datasource 	<ul style="list-style-type: none"> destroy datasource create modified datasource
jbossas.Queue	<ul style="list-style-type: none"> stop Server create Queue start Server 	<ul style="list-style-type: none"> stop Server destroy Queue start Server 	<ul style="list-style-type: none"> stop Server destroy Queue create modified Queue start Server
jbossas.Topic	<ul style="list-style-type: none"> stop Server create Topic start Server 	<ul style="list-style-type: none"> stop Server destroy Topic start Server 	<ul style="list-style-type: none"> stop Server destroy Topic create modified Topic start Server
jbossas.DeployedConfigurationFile	<ul style="list-style-type: none"> stop Server deploying config files 	<ul style="list-style-type: none"> stop Server undeploy config files 	<ul style="list-style-type: none"> stop Server undeploy config files

	<ul style="list-style-type: none"> • start Server 	<ul style="list-style-type: none"> • start Server 	<ul style="list-style-type: none"> • deploy new config files • start Server
jbossas.DeployedConfigurationFolder	<ul style="list-style-type: none"> • stop Server • deploying configuration folder • start Server 	<ul style="list-style-type: none"> • stop Server • undeploy configuration folder • start Server 	<ul style="list-style-type: none"> • stop Server • undeploy configuration folder • deploy new configuration folder • start Server

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.TransactionaDataSource*

```
<type-modification type="jbossas.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*. Similar '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 kept 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
jbossas.ConfigurationFile	File containing application configurations that is deployed to the server conf directory
jbossas.ConfigurationFolder	Folder containing files for application configuration that are deployed to the server conf directory
jbossas.Ear	A JEE EAR archive
jbossas.Library	The Library as deployed on the jboss server
jbossas.NonTransactionalDatasourceSpec	A JBoss No Transaction Datasource (deployable)
jbossas.QueueSpec	Specification for a JBoss queue
jbossas.TopicSpec	Specification for a JBoss topic
jbossas.TransactionalDatasourceSpec	A JBoss Local Transaction Datasource (deployable)
jbossas.TransactionalXADatasourceSpec	A JBoss XA Datasource (deployable)
jbossas.War	A JEE WAR archive

Deployeds

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

Containers

CI	Description
jbossas.BaseServer	Base JBoss Application Server
jbossas.ServerV4	Jboss 4 application server
jbossas.ServerV5	Jboss 5 application server
jbossas.ServerV6	Jboss 6 application server

Other Configuration Items

CI	Description
jbossas.Artifact	Base type for a JBoss artifact
jbossas.BaseServer	Base JBoss Application Server
jbossas.Configuration	Base type for a JBoss configuration
jbossas.ConfigurationFile	File containing application configurations that is deployed to the server conf directory
jbossas.ConfigurationFolder	Folder containing files for application configuration that are deployed to the server conf directory
jbossas.Datasource	Base type for a JBoss datasource
jbossas.DeployedConfigurationFile	Deployed configuration file
jbossas.DeployedConfigurationFolder	Deployed configuration folder
jbossas.DeployedLibrary	The Library as deployed on the jboss server
jbossas.Ear	A JEE EAR archive
jbossas.EarModule	Ear with values configured for a deployment
jbossas.Library	The Library as deployed on the jboss server
jbossas.NonTransactionalDatasource	A JBoss No Transaction Datasource
jbossas.NonTransactionalDatasourceSpec	A JBoss No Transaction Datasource (deployable)
jbossas.Queue	A JBoss Queue
jbossas.QueueSpec	Specification for a JBoss queue
jbossas.Resource	Description unavailable
jbossas.ServerV4	Jboss 4 application server
jbossas.ServerV5	Jboss 5 application server
jbossas.ServerV6	Jboss 6 application server
jbossas.Topic	A JBoss topic
jbossas.TopicSpec	Specification for a JBoss topic
jbossas.TransactionalDatasource	A JBoss Local Transaction Datasource
jbossas.TransactionalDatasourceSpec	A JBoss Local Transaction Datasource (deployable)
jbossas.TransactionalXADatasource	A JBoss XA Datasource
jbossas.TransactionalXADatasourceSpec	A JBoss XA Datasource (deployable)
jbossas.War	A JEE WAR archive
jbossas.WarModule	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	
* container : <code>Cl<udm.Container></code>	The container on which this deployed runs.
Public Properties	
deployable : <code>Cl<udm.Deployable></code>	The deployable that this deployed is derived from.
placeholders : <code>MAP_STRING_STRING</code>	A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>
targetFile : <code>STRING</code>	Name of the artifact on the generic server.
Hidden Properties	
* createOrder : <code>INTEGER = 50</code>	Create Order
* createVerb : <code>STRING = Create</code>	Create Verb
* destroyOrder : <code>INTEGER = 40</code>	Destroy Order
* destroyVerb : <code>STRING = Destroy</code>	Destroy Verb
* modifyOrder : <code>INTEGER = 50</code>	The order of the step in the step list for the modify operation.
* modifyVerb : <code>STRING = Modify</code>	Modify Verb
* noopOrder : <code>INTEGER = 50</code>	The order of the step in the step list for the noop operation.
* noopVerb : <code>STRING = Modify</code>	Noop Verb
* targetDirectory : <code>STRING</code> = <code>\${deployed.container.home}/server/\${deployed.container.serverName}/deploy</code>	Target Directory
createTargetDirectory : <code>BOOLEAN = false</code>	Create the target directory on the generic server if it does not exist.
inspectClasspathResources : <code>SET_OF_STRING</code>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : <code>STRING</code>	Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : <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.
preserveExistingFiles : <code>BOOLEAN = false</code>	if true, preserve the existing files on the remote host (do not delete the copied files during the destroy operation).
restartRequired : <code>BOOLEAN = true</code>	Restart Required
restartRequiredForNoop : <code>BOOLEAN = false</code>	The generic container requires a restart for the NOOP action performed by this deployed.
targetDirectoryShared : <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	
* host : <code>Cl<over>here.Host</over></code>	Host upon which the container resides
Public Properties	
* ajpPort : <code>INTEGER = 8009</code>	AJP connector port
* controlPort : <code>INTEGER = 1099</code>	Connector control port
* home : <code>STRING</code>	JBoss application server installation directory. e.g. /opt/jboss/5.1.0.GA
* httpPort : <code>INTEGER = 8080</code>	HTTP/1.1 connector port
* serverName : <code>STRING = default</code>	Server configuration name e.g. default or minimal
adminPassword : <code>STRING</code>	Password of the user who connect to jboss using twiddle
adminUsername : <code>STRING</code>	Username used to connect to jboss using twiddle
envVars : <code>MAP_STRING_STRING</code>	Environment variables for container
tags : <code>SET_OF_STRING</code>	If set, only deployables with the same tag will be automatically mapped to this container.
windowsServiceName : <code>STRING</code>	The service name under which JBoss is running. By default, Deployit expects JBoss to be installed as a service on Windows.

Hidden Properties	
* bindAddress : STRING = 0.0.0.0	Address where to bind the server on starting.
* inspectTemplate : STRING = jboss/container/inspect	Script (a freemarker template) used to inspect the server
* inspectTemplateClasspathResources : SET_OF_STRING = [twiddle-invoker.ftl, jboss/container/inspect.bat, jboss/container/inspect.sh, jboss/container/query.sh, jboss/container/query.bat]	Inspect Template Classpath Resources
* jmsResourceFileSuffix : STRING = service	Jms Resource File Suffix
* queryTemplate : STRING = jboss/container/query	Script (a freemarker template) used to query a mbean value from the server
* 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/container/start	Script used to start the server
* startWaitTime : INTEGER = 30	Duration (in secs) to wait after the start server step has been executed
* stopOrder : INTEGER = 10	The order of the stop container step in the step list.
* stopScript : STRING = jboss/container/stop	Script used to stop the server
* stopWaitTime : INTEGER = 10	Duration (in secs) to wait after the stop server step has been executed
inspectClasspathResources : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
restartScript : STRING	Classpath to the script used to restart the generic container.
restartWaitTime : INTEGER = 0	The time to wait in seconds for a container restart action.
startOptions : STRING =	Optional command line arguments to pass to the JBoss startup script.

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	
* container : Cl <udm.Container>	The container on which this deployed runs.
Public Properties	
deployable : Cl <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.
inspectClasspathResources : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : SET_OF_STRING	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.
preserveExistingFiles : BOOLEAN = false	if true, preserve the existing files on the remote host (do not delete the copied files during the destroy operation).
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.

jbosscas.ConfigurationFile

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

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

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

jbossas.Datasource

Virtual Type	
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	
* container : Cl <udm.Container>	The container on which this deployed runs.

Public Properties	
* jndiName : STRING	Jndi Name
* maxPoolSize : INTEGER = 20	Max Pool Size
* minPoolSize : INTEGER = 0	Min Pool Size
checkValidConnectionSql : STRING	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.
deployable : Cl<udm.Deployable>	The deployable that this deployed is derived from.
exceptionSorterClassName : STRING	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.
password : STRING	the password used when creating the connection (not used when security is configured)
useJavaContext : BOOLEAN = true	Use Java Context
userName : STRING	the user name used when creating the connection (not used when security is configured)

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/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
* targetFile : STRING = \${deployed.deployable.name}-ds.xml	Target File
* template : STRING = jboss/datasource/template.xml.ftl	Template
createTargetDirectory : BOOLEAN = false	Create the target directory on the generic server if it does not exist.
inspectClasspathResources : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : SET_OF_STRING	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.
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.

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	
* container : CL <udm.Container>	The container on which this deployed runs.

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

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 folder

Parent	
* container : <code>Cl<udm.Container></code>	The container on which this deployed runs.

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

Type Hierarchy `jbossas.Artifact` >> `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.

Parent	
* container : <code>Cl<udm.Container></code>	The container on which this deployed runs.

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

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

A JEE EAR archive

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

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	
* container : Cl <udm.Container>	The container on which this deployed runs.
Public Properties	
deployable : Cl <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.
inspectClasspathResources : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : SET_OF_STRING	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.
preserveExistingFiles : BOOLEAN = false	if true, preserve the existing files on the remote host (do not delete the copied files during the destroy operation).
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.

jbossas.Library

Type Hierarchy	generic.File >> udm.BaseDeployableFileArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem
Interfaces	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	
excludeFileNamesRegex : STRING	Regular expression that matches file names that must be excluded from scanning
placeholders : SET_OF_STRING	Placeholders detected in this artifact
scanPlaceholders : BOOLEAN = true	Whether to scan this artifact for placeholders when it is imported
tags : SET_OF_STRING	If set, this deployable will only be mapped automatically to containers with the same tag.
targetFile : STRING	Name of the artifact on the generic server. (string)
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

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	
* container : Cl <udm.Container>	The container on which this deployed runs.
Public Properties	
* connectionUrl : STRING	the JDBC driver connection url string
* driverClass : STRING	the JDBC driver class implementing java.sql.Driver
* jndiName : STRING	Jndi Name
* maxPoolSize : INTEGER = 20	Max Pool Size
* minPoolSize : INTEGER = 0	Min Pool Size
checkValidConnectionSql : STRING	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.
connectionProperties : MAP_STRING_STRING	used to configure the connections retrieved from the java.sql.Driver
deployable : Cl <udm.Deployable>	The deployable that this deployed is derived from.
exceptionSorterClassName : STRING	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.
password : STRING	the password used when creating the connection (not used when security is configured)
useJavaContext : BOOLEAN = true	Use Java Context
userName : STRING	the user name used when creating the connection (not used when security is configured)

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/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
* targetFile : STRING = \${deployed.deployable.name}-ds.xml	Target File
* template : STRING = jboss/datasource/template.xml.ftl	Template
createTargetDirectory : BOOLEAN = false	Create the target directory on the generic server if it does not exist.
inspectClasspathResources : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : SET_OF_STRING	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.
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.

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	
checkValidConnectionSql : STRING	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)
connectionProperties : MAP_STRING_STRING	used to configure the connections retrieved from the java.sql.Driver (map_string_string)
connectionUrl : STRING	the JDBC driver connection url string (string)
driverClass : STRING	the JDBC driver class implementing java.sql.Driver (string)
exceptionSorterClassName : STRING	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)
jndiName : STRING	Jndi Name (string)
maxPoolSize : STRING	Max Pool Size (integer)
minPoolSize : STRING	Min Pool Size (integer)
password : STRING	the password used when creating the connection (not used when security is configured) (string)
tags : SET_OF_STRING	If set, this deployable will only be mapped automatically to containers with the same tag.
useJavaContext : STRING	Use Java Context (boolean)
userName : STRING	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	
* container : Cl<udm.Container>	The container on which this deployed runs.
Public Properties	
* jndiName : STRING	Jndi Name
* maxDepth : INTEGER = 0	Max Depth
deployable : Cl<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/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
* targetFile : STRING = \${deployed.deployable.name}- \${deployed.container.jmsResourceFileSuffix}.xml	Target File
* template : STRING = jboss/queue/template\${deployed.container.version}.xml.ftl	Template
createTargetDirectory : BOOLEAN = false	Create the target directory on the generic server if it does not exist.
inspectClasspathResources : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : SET_OF_STRING	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.
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.

jbossas.QueueSpec

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

Interfaces udm.Tagable, udm.Deployable, udm.ConfigurationItem

Specification for a JBoss queue

Public Properties	
jndiName : STRING	Jndi Name (string)
maxDepth : STRING	Max Depth (integer)
tags : SET_OF_STRING	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	
* container : <code>CI<udm.Container></code>	The container on which this deployed runs.
Public Properties	
deployable : <code>CI<udm.Deployable></code>	The deployable that this deployed is derived from.
targetFile : <code>STRING</code>	Name of the artifact on the generic server.
Hidden Properties	
* createOrder : <code>INTEGER = 50</code>	Create Order
* createVerb : <code>STRING = Create</code>	Create Verb
* destroyOrder : <code>INTEGER = 40</code>	Destroy Order
* destroyVerb : <code>STRING = Destroy</code>	Destroy Verb
* modifyOrder : <code>INTEGER = 50</code>	The order of the step in the step list for the modify operation.
* modifyVerb : <code>STRING = Modify</code>	Modify Verb
* noopOrder : <code>INTEGER = 50</code>	The order of the step in the step list for the noop operation.
* noopVerb : <code>STRING = Modify</code>	Noop Verb
* targetDirectory : <code>STRING</code> = <code>\${deployed.container.home}/server/\${deployed.container.serverName}/deploy</code>	Target Directory
* template : <code>STRING</code>	Classpath to the freemarker template used to generate the content of the final text base artifact.
createTargetDirectory : <code>BOOLEAN = false</code>	Create the target directory on the generic server if it does not exist.
inspectClasspathResources : <code>SET_OF_STRING</code>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : <code>STRING</code>	Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : <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.
restartRequired : <code>BOOLEAN = true</code>	Restart Required
restartRequiredForNoop : <code>BOOLEAN = false</code>	The generic container requires a restart for the NOOP action performed by this deployed.
targetDirectoryShared : <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.ServerV4

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

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

JBoss 4 application server

Parent	
* host : Ck<overthere.Host>	Host upon which the container resides
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
* httpPort : INTEGER = 8080	HTTP/1.1 connector port
* serverName : STRING = default	Server configuration name e.g. default or minimal
 adminPassword : STRING	Password of the user who connect to jboss using twiddle
 adminUsername : STRING	Username used to connect to jboss using twiddle
envVars : MAP_STRING_STRING	Environment variables for container
tags : SET_OF_STRING	If set, only deployables with the same tag will be automatically mapped to this container.
windowsServiceName : STRING	The service name under which JBoss is running. By default, Deployit expects JBoss to be installed as a service on Windows.

Hidden Properties	
* bindAddress : STRING = 0.0.0.0	Address where to bind the server on starting.
* inspectTemplate : STRING = jboss/container/inspect	Script (a freemarker template) used to inspect the server
* inspectTemplateClasspathResources : SET_OF_STRING = [twiddle-invoker.ftl, jboss/container/inspect.bat, jboss/container/inspect.sh, jboss/container/query.sh, jboss/container/query.bat]	Inspect Template Classpath Resources
* jmsResourceFileSuffix : STRING = service	Jms Resource File Suffix
* queryTemplate : STRING = jboss/container/query	Script (a freemarker template) used to query a mbean value from the server
* 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/container/start	Script used to start the server
* startWaitTime : INTEGER = 30	Duration (in secs) to wait after the start server step has been executed
* stopOrder : INTEGER = 10	The order of the stop container step in the step list.
* stopScript : STRING = jboss/container/stop	Script used to stop the server
* stopWaitTime : INTEGER = 10	Duration (in secs) to wait after the stop server step has been executed
* version : INTEGER = 4	Version
inspectClasspathResources : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
restartScript : STRING	Classpath to the script used to restart the generic container.
restartWaitTime : INTEGER = 0	The time to wait in seconds for a container restart action.
startOptions : STRING =	Optional command line arguments to pass to the JBoss startup script.





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	
* host : Cl<overthere.Host>	Host upon which the container resides

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
* httpPort : INTEGER = 8080	HTTP/1.1 connector port
* serverName : STRING = default	Server configuration name e.g. default or minimal
 adminPassword : STRING	Password of the user who connect to jboss using twiddle
 adminUsername : STRING	Username used to connect to jboss using twiddle
envVars : MAP_STRING_STRING	Environment variables for container
tags : SET_OF_STRING	If set, only deployables with the same tag will be automatically mapped to this container.
windowsServiceName : STRING	The service name under which JBoss is running. By default, Deployit expects JBoss to be installed as a service on Windows.

Hidden Properties	
* bindAddress : STRING = 0.0.0.0	Address where to bind the server on starting.
* inspectTemplate : STRING = jboss/container/inspect	Script (a freemarker template) used to inspect the server
* inspectTemplateClasspathResources : SET_OF_STRING = [twiddle-invoker.ftl, jboss/container/inspect.bat, jboss/container/inspect.sh, jboss/container/query.sh, jboss/container/query.bat]	Inspect Template Classpath Resources
* jmsResourceFileSuffix : STRING = service	Jms Resource File Suffix
* queryTemplate : STRING = jboss/container/query	Script (a freemarker template) used to query a mbean value from the server
* 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/container/start	Script used to start the server
* startWaitTime : INTEGER = 30	Duration (in secs) to wait after the start server step has been executed
* stopOrder : INTEGER = 10	The order of the stop container step in the step list.
* stopScript : STRING = jboss/container/stop	Script used to stop the server
* stopWaitTime : INTEGER = 10	Duration (in secs) to wait after the stop server step has been executed
* version : INTEGER = 5	Version
inspectClasspathResources : SET_OF_STRING	
Additional classpath resources that should be uploaded to the working directory before executing the inspect script.	
inspectScript : STRING	
Classpath to the script used to inspect the generic container.	
restartScript : STRING	
Classpath to the script used to restart the generic container.	
restartWaitTime : INTEGER = 0	
The time to wait in seconds for a container restart action.	
startOptions : STRING =	
Optional command line arguments to pass to the JBoss startup script.	

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	
* host : Cl<overthere.Host>	Host upon which the container resides

Public Properties	
* ajpPort : INTEGER = 8009	AJP connector port
* controlPort : INTEGER = 1090	Connector control port
* home : STRING	JBoss application server installation directory. e.g. /opt/jboss/5.1.0.GA
* httpPort : INTEGER = 8080	HTTP/1.1 connector port
* serverName : STRING = default	Server configuration name e.g. default or minimal
adminPassword : STRING	Password of the user who connect to jboss using twiddle
adminUsername : STRING	Username used to connect to jboss using twiddle
envVars : MAP_STRING_STRING	Environment variables for container
tags : SET_OF_STRING	If set, only deployables with the same tag will be automatically mapped to this container.
windowsServiceName : STRING	The service name under which JBoss is running. By default, Deployit expects JBoss to be installed as a service on Windows.

Hidden Properties	
* bindAddress : STRING = 0.0.0.0	Address where to bind the server on starting.
* inspectTemplate : STRING = jboss/container/inspect	Script (a freemarker template) used to inspect the server
* inspectTemplateClasspathResources : SET_OF_STRING = [twiddle-invoker.ftl, jboss/container/inspect.bat, jboss/container/inspect.sh, jboss/container/query.sh, jboss/container/query.bat]	Inspect Template Classpath Resources
* jmsResourceFileSuffix : STRING = hornetq-jms	Jms Resource File Suffix
* queryTemplate : STRING = jboss/container/query	Script (a freemarker template) used to query a mbean value from the server
* 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/container/start	Script used to start the server
* startWaitTime : INTEGER = 30	Duration (in secs) to wait after the start server step has been executed
* stopOrder : INTEGER = 10	The order of the stop container step in the step list.
* stopScript : STRING = jboss/container/stop	Script used to stop the server
* stopWaitTime : INTEGER = 10	Duration (in secs) to wait after the stop server step has been executed
* version : INTEGER = 6	Version
inspectClasspathResources : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
restartScript : STRING	Classpath to the script used to restart the generic container.
restartWaitTime : INTEGER = 0	The time to wait in seconds for a container restart action.
startOptions : STRING =	Optional command line arguments to pass to the JBoss startup script.

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	
* container : Cl <udm.Container>	The container on which this deployed runs.
Public Properties	
* jndiName : STRING	Jndi Name
* maxDepth : INTEGER = 0	Max Depth
deployable : Cl <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/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
* targetFile : STRING = \${deployed.deployable.name}- \${deployed.container.jmsResourceFileSuffix}.xml	Target File
* template : STRING = jboss/topic/template\${deployed.container.version}.xml.ftl	Template
createTargetDirectory : BOOLEAN = false	Create the target directory on the generic server if it does not exist.
inspectClasspathResources : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : SET_OF_STRING	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.
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.

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	
jndiName : STRING	Jndi Name (string)
maxDepth : STRING	Max Depth (integer)
tags : SET_OF_STRING	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
 container : Ck<udm.Container> The container on which this deployed runs.

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/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
* targetFile : STRING = \${deployed.deployable.name}-ds.xml	Target File
* template : STRING = jboss/datasource/template.xml.ftl	Template
createTargetDirectory : BOOLEAN = false	Create the target directory on the generic server if it does not exist.
inspectClasspathResources : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : SET_OF_STRING	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.
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.

jbosstxdatasource.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	
checkValidConnectionSql : STRING	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)
connectionProperties : MAP_STRING_STRING	used to configure the connections retrieved from the java.sql.Driver (map_string_string)
connectionUrl : STRING	the JDBC driver connection url string (string)
driverClass : STRING	the JDBC driver class implementing java.sql.Driver (string)
exceptionSorterClassName : STRING	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)
jndiName : STRING	Jndi Name (string)
maxPoolSize : STRING	Max Pool Size (integer)
minPoolSize : STRING	Min Pool Size (integer)
password : STRING	the password used when creating the connection (not used when security is configured) (string)
tags : SET_OF_STRING	If set, this deployable will only be mapped automatically to containers with the same tag.
useJavaContext : STRING	Use Java Context (boolean)
userName : STRING	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	
* container : Cl<udm.Container>	The container on which this deployed runs.

Public Properties	
* jndiName : STRING	Jndi Name
* maxPoolSize : INTEGER = 20	Max Pool Size
* minPoolSize : INTEGER = 0	Min Pool Size
* xaDataSourceClass : STRING	the class implementing the XADataSource
checkValidConnectionSql : STRING	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.
deployable : Cl<udm.Deployable>	The deployable that this deployed is derived from.
exceptionSorterClassName : STRING	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.
password : STRING	the password used when creating the connection (not used when security is configured)
trackConnectionByTx : BOOLEAN = true	set to fix problems with Oracle (not necessarily in JBoss-5.x where it is enabled by default and the element is deprecated)
useJavaContext : BOOLEAN = true	Use Java Context
userName : STRING	the user name used when creating the connection (not used when security is configured)
xaDataSourceProperties : MAP_STRING_STRING	properties used to configure the XADataSource

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/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
* targetFile : STRING = \${deployed.deployable.name}-ds.xml	Target File
* template : STRING = jboss/datasource/template.xml.ftl	Template
createTargetDirectory : BOOLEAN = false	Create the target directory on the generic server if it does not exist.
inspectClasspathResources : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : SET_OF_STRING	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.
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.

jbosscas.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	
checkValidConnectionSql : STRING	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)
exceptionSorterClassName : STRING	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)
jndiName : STRING	Jndi Name (string)
maxPoolSize : STRING	Max Pool Size (integer)
minPoolSize : STRING	Min Pool Size (integer)
password : STRING	the password used when creating the connection (not used when security is configured) (string)
tags : SET_OF_STRING	If set, this deployable will only be mapped automatically to containers with the same tag.
trackConnectionByTx : STRING	set to fix problems with Oracle (not necessarily in JBoss-5.x where it is enabled by default and the element is deprecated) (boolean)
useJavaContext : STRING	Use Java Context (boolean)
userName : STRING	the user name used when creating the connection (not used when security is configured) (string)
xaDataSourceClass : STRING	the class implementing the XADataSource (string)
xaDataSourceProperties : MAP_STRING_STRING	properties used to configure the XADataSource (map_string_string)

jbossas.War

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

A JEE WAR archive

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

jbossas.WarModule

Type Hierarchy	jbossas.Artifact >> 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

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