

# **Deployit JBoss Application Server 7+ Plugin Manual**

**Version 3.7.0-BETA-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 deployments	4
Deployable vs. Container table	4
Deployed Actions Table	4
Deploying applications	4
Standalone Mode	4
Domain Mode	4
Discovery	4
Extension points	5
Extending the plugin to support JDBC Driver deployment	5
Extending the plugin with custom control task	6
CI Reference	7
Configuration Item Overview	7
Deployable Configuration Items	7
Deployed Configuration Items	7
Topology Configuration Items	7
Virtual Deployed Configuration Items	7
Virtual Topology Configuration Items	7
Configuration Item Details	7
jbossdm.BaseDataSource	7
jbossdm.CliBasedContainer	9
jbossdm.CliManagedDeployed	9
jbossdm.CliManagedDeployedArtifact	10
jbossdm.DataSource	11
jbossdm.DataSourceSpec	13
jbossdm.Domain	14
jbossdm.Ear	15
jbossdm.EarModule	15
jbossdm.JeeDataSource	16
jbossdm.JeeXaDataSource	18
jbossdm.Profile	20
jbossdm.Queue	20
jbossdm.QueueSpec	21
jbossdm.ServerGroup	21
jbossdm.StandaloneServer	21
jbossdm.Topic	22
jbossdm.TopicSpec	23
jbossdm.War	23
jbossdm.WarModule	23
jbossdm.XaDataSource	24
jbossdm.XaDataSourceSpec	26

## Preface

This document describes the functionality provided by the JBoss Domain (JBoss AS 7.1+, JBoss EAP 6) plugin.

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

## Overview

The JBoss Domain plugin is a Deployit plugin that adds the capability to manage deployments and resources on JBoss application server 7.1+ or JBoss EAP 6. The plugin has the capability of managing application artifacts, datasource and other JMS resources via the JBoss Cli, and can easily be extended to support more deployment options or management of new artifacts/resources on JBoss AS.

## Features

- Domain and Standalone mode support
- Deployment of application artifacts
  - Enterprise application (EAR)
  - Web application (WAR)
- Deployment of resources
  - Datasource including XA Datasource
  - JMS Queue
  - JMS Topic
- Discovery of Profiles and ServerGroups in Domain

## Requirements

- **Deployit requirements**
  - **Deployit:** version 3.7+
  - **Jython:** jython-standalone-2.5.1.jar installed in `<DEPLOYIT_SERVER_HOME>/lib`
- **Infrastructural requirements**
  - **JBoss AS versions:** 7.1+
  - **JBoss EAP versions:** 6.x
  - **User credentials** for accessing the Host and JBoss Cli.

## 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 contains 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: jbosscdm.DatasourceSpec
CI-jndiName: jdbc/sampleDatasource
CI-connectionUrl: jdbc:mysql://localhost/test
CI-driverName: mysql
CI-username: {{DATABASE_USERNAME}}
CI-password: {{DATABASE_PASSWORD}}

Name: testQueue
CI-Type: jbosscdm.QueueSpec
CI-jndiName: jms/testQueue

Name: testTopic
```

```
CI-Type: jbossdm.TopicSpec
CI-jndiName: jms/testTopic
```

## Using the deployables and deployed

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

### Deployable vs. Container table

Deployable	Container	Generated deployed
jee.Ear jbossdm.Ear	jbossdm.ApplicationContainer	jbossdm.EarModule
jee.War jbossdm.War	jbossdm.ApplicationContainer	jbossdm.WarModule
jee.DataSourceSpec	jbossdm.ResourceContainer	jbossdm.JeeXaDataSource jbossdm.JeeDataSource
jbossdm.XaDataSourceSpec	jbossdm.ResourceContainer	jbossdm.XaDataSource
jbossdm.DataSourceSpec	jbossdm.ResourceContainer	jbossdm.Datasource
jee.QueueSpec jbossdm.QueueSpec	jbossdm.ResourceContainer	jbossdm.Queue
jee.TopicSpec jbossdm.TopicSpec	jbossdm.ResourceContainer	jbossdm.Topic

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

### Deployed Actions Table

Deployed	Actions performed for operations		
	Create	Destroy	Modify
jbossdm.EarModule jbossdm.WarModule	<ul style="list-style-type: none"> <li>upload artifact</li> <li>deploy application</li> </ul>	<ul style="list-style-type: none"> <li>undeploy old application version</li> </ul>	<ul style="list-style-type: none"> <li>undeploy old application version</li> <li>deploy new application version</li> </ul>
jbossdm.DataSource jbossdm.XaDataSource jbossdm.JeeDataSource jbossdm.JeeXaDataSource	<ul style="list-style-type: none"> <li>create datasource</li> <li>set connection/datasource properties</li> <li>enable 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 datasource</li> <li>set connection/datasource properties</li> <li>enable datasource</li> </ul>
jbossdm.Queue	<ul style="list-style-type: none"> <li>create Queue</li> </ul>	<ul style="list-style-type: none"> <li>destroy Queue</li> </ul>	<ul style="list-style-type: none"> <li>destroy Queue</li> <li>create modified Queue</li> </ul>
jbossdm.Topic	<ul style="list-style-type: none"> <li>create Topic</li> </ul>	<ul style="list-style-type: none"> <li>destroy Topic</li> </ul>	<ul style="list-style-type: none"> <li>destroy Topic</li> <li>create modified Topic</li> </ul>

## Deploying applications

Note that the plugin uses the JBoss Cli to (un)install artifacts and resources. As such, the plugin assumes that the JBoss Domain or Standalone server has already been started. The plugin does not support the starting of the domain or standalone server prior to a deployment.

### Standalone Mode

Artifacts (war, ear) and resources (datasources, queues, topics, etc) can be targeted to a [StandaloneServer](#).

### Domain Mode

Artifacts (war, ear) can be targeted to either a [Domain](#) or [ServerGroup](#). When targeted to a domain, the artifacts are (un)installed on all server groups defined for the domain. For specific targetting of artifacts to certain server groups, you can define the server groups in your environment.

Resources (datasources, queues, topics, etc) can be targeted to either a [Domain](#) or [Profile](#). When targeted to a domain, the resources are (un)installed in the "default" profile. For specific targetting of resources to certain profiles, you can define the profiles in your environment.

## Discovery

The plugin supports the discovery of Profiles and Server Groups in a Domain.

Here is an example CLI script which discovers a sample Domain:

```
host = repository.create(factory.configurationItem('Infrastructure/jboss-host',
```

```
'overthere.SshHost',
    {'connectionType':'SFTP','address': 'jboss-7','username':
'root','password':'centos','os':'UNIX'}})
jboss = factory.configurationItem('Infrastructure/jboss-host/jboss-domain',
'jbossdm.Domain',
    {'home':'/opt/jboss/7', 'host':'Infrastructure/jboss-host',
'username':'jbossAdmin', 'password':'jboss'})
cis = deployit.discover(jboss)
print cis

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

Few things to note about the above discovery example:

- JBoss Domain has a containment relation with a Host (created under a Host), so the server id has been kept as 'Infrastructure/jboss-host/jboss-domain'

## Extension points

The plugin is designed to be extended through Deployit's Plugin API type system and jython. The plugin wraps the JBoss Cli with a jython runtime environment, thus allowing extenders to interact with JBoss and Deployit from the script. Note that the jython script is executed on the Deployit Server itself and has full access to the following Deployit objects :

- **deployed**: The current deployed object on which the operation has been triggered.
- **step**: The step object that the script is being executed from. Exposes an overthere remote connection for file manipulation and a method to execute JBoss Cli commands.
- **container**: The container object to which the deployed is targeted to.
- **delta**: The delta specification that lead to the script being executed.
- **deployedApplication**: The entire deployed application.

The plugin associates **Create**, **Modify**, **Destroy**, **Noop** and **Inspect** operations received from Deployit with jython scripts that need to be executed for the specific operation to be performed.

There also exists an advanced method to extend the plugin, but the implementation of this form of extension needs to be written in the Java programming language and consists of writing so-called `DeployedContributors`, `PlanPreProcessors` and `Contributors`.

Please refer to the *Customization Manual* for a detailed explanation of the type system and advanced methods of customization of plugins. Also refer to the Overthere documentation for working with remote files.

## Extending the plugin to support JDBC Driver deployment

In this example we will deploy a jdbc driver jar to a [Domain](#) or [StandaloneServer](#) as a module and register the driver with JBoss datasources subsystem.

### Define the deployed and deployable to represent a JDBC Driver

The following synthetic.xml snippet shows the definition of the JDBC Driver deployed. The deployed will be targeted to a [Domain](#) or a [StandaloneServer](#). Please refer to the CI reference section of this document to understand the interfaces and class hierarchy of these types.

```
<type type="jbossdm.JdbcDriverModule"
extends="jbossdm.CliManagedDeployedArtifact"
    deployable-type="jbossdm.JdbcDriver" container-
type="jbossdm.CliManagingContainer">
  <generate-deployable type="jbossdm.JdbcDriver"
extends="udm.BaseDeployableArchiveArtifact">

  <property name="driverName"/>
  <property name="driverModuleName"/>
  <property name="driverXaDataSourceClassName"/>

  <!-- hidden properties to specify the jython scripts to execute for an
operation -->
  <property name="createScript" default="jboss/dm/ds/create-jdbc-driver.py"
hidden="true"/>
</type>
```

### Defined the create-jdbc-driver.py

```
from com.xebialabs.overthere.util import OverthereUtils
```

```
#create module directory to copy jar and module.xml to
driverModuleName = deployed.getProperty("driverModuleName")
moduleRelPath = driverModuleName.replaceAll("\\.", "/")
moduleAbsolutePath = "%s/modules/%s" % (container.getProperty("home"),
moduleRelPath)
moduleDir = step.getRemoteConnection().getFile(moduleAbsolutePath);
moduleDir.mkdirs();
#upload jar
moduleJar = moduleDir.getFile(deployed.file.getName())
deployed.file.copyTo(moduleJar)

moduleXmlContent = """
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="%s">
  <resources>
    <resource-root path="%s"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
""" % (deployed.getProperty("driverModuleName"), deployed.file.getName())

#create module.xml
moduleXml = moduleDir.getFile("module.xml")
OverthereUtils.write(moduleXmlContent.getBytes(), moduleXml)

#register driver with the datasource subsystem
driverName = deployed.getProperty("driverName")
xaClassName = deployed.getProperty("driverXaDatasourceClassName")
cmd = '/subsystem=datasources/jdbc-driver=%s:add(driver-name="%s",driver-
module-name="%s",driver-xa-datasource-class-name="%s")' % (driverName,
driverName, driverName, xaClassName)
cmd = prependProfilePath(cmd) #prefix with profile if deploying to domain
executeCmd(cmd) #used to execute a JBoss Cli command.
```

## Extending the plugin with custom control task

The plugin has the capability to add control tasks to [CliManagedDeployed](#) or [CliManagedContainer](#). The control task can be specified as a jython script that will be executed on the Deployit Server or as an OS shell script that will be run on the target host. The OS shell script is first processed with FreeMarker before being executed.

### Creating a jython based control task to list jdbc drivers in a StandaloneServer

Synthetic.xml snippet

```
<type-modification type="jbossdm.StandaloneServer">
  <property name="listJdbcDriversPythonTaskScript" hidden="true"
default="jboss/dm/container/list-jdbc-drivers.py"/>
  <!-- Note "PythonTaskScript" is appended to the method name to determine the
script to run. -->
  <method name="listJdbcDrivers"/>
</type>
```

list-jdbc-drivers.py snippet

```
drivers = executeCmd("/subsystem=datasources:installed-drivers-list")
logOutput(drivers) #outputs to the step log
```

### Start the StandaloneServer

Synthetic.xml snippet

```
<type-modification type="jbossdm.StandaloneServer">
  <property name="startShellTaskScript" hidden="true"
default="jboss/dm/container/start-standalone"/>
  <!-- Note "ShellTaskScript" is appended to the method name to determine the
script to run. -->
  <method name="start"/>
</type>
```

start-standalone.sh snippet

```
nohup ${container.home}/bin/standalone.sh >>nohup.out 2>&1 &
sleep 2
echo background process to start standalone server executed.
```

## CI Reference

### Configuration Item Overview

#### Deployable Configuration Items

CI	Description
<a href="#">jbossdm.DataSourceSpec</a>	DataSource
<a href="#">jbossdm.Ear</a>	A JEE EAR archive
<a href="#">jbossdm.QueueSpec</a>	A Queue
<a href="#">jbossdm.TopicSpec</a>	A Topic
<a href="#">jbossdm.War</a>	A JEE WAR archive
<a href="#">jbossdm.XaDataSourceSpec</a>	XA DataSource

#### Deployed Configuration Items

CI	Description
<a href="#">jbossdm.DataSource</a>	DataSource
<a href="#">jbossdm.EarModule</a>	Ear with values configured for a deployment
<a href="#">jbossdm.JeeDataSource</a>	Datasource
<a href="#">jbossdm.JeeXaDataSource</a>	XA DataSource
<a href="#">jbossdm.Queue</a>	A Jboss Queue
<a href="#">jbossdm.Topic</a>	A JBoss topic
<a href="#">jbossdm.WarModule</a>	War with values configured for a deployment
<a href="#">jbossdm.XaDataSource</a>	XA DataSource

#### Topology Configuration Items

CI	Description
<a href="#">jbossdm.Domain</a>	JBoss Cli Managed Container
<a href="#">jbossdm.Profile</a>	JBoss Profile
<a href="#">jbossdm.ServerGroup</a>	JBoss Server Group
<a href="#">jbossdm.StandaloneServer</a>	JBoss Standalone Server

#### Virtual Deployed Configuration Items

CI	Description
<a href="#">jbossdm.BaseDataSource</a>	Base definition of a DataSource
<a href="#">jbossdm.CliManagedDeployed</a>	Base for all deployed that utilize the JBoss Cli for configuration
<a href="#">jbossdm.CliManagedDeployedArtifact</a>	Base for all deployed artifacts that utilize the JBoss Cli for configuration

#### Virtual Topology Configuration Items

CI	Description
<a href="#">jbossdm.CliBasedContainer</a>	JBoss Cli Managed Container


### Configuration Item Details

#### jbossdm.BaseDataSource

**Hierarchy** [jbossdm.CliManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Base definition of a DataSource

Public Properties	
 <b>container</b>	: <a href="#">CI</a> < <a href="#">udm.Container</a> >
The container on which this deployed runs.	
<b>driverName</b>	: <a href="#">STRING</a>
Defines the JDBC driver the datasource should use. It is a symbolic name matching the the name of installed driver. In case the driver is deployed as jar, the name is the name of deployment unit.	
<b>jndiName</b>	: <a href="#">STRING</a>
Specifies the JNDI name for the datasource	

**backgroundValidation** : **BOOLEAN** = false

An element to specify that connections should be validated on a background thread versus being validated prior to use. Changing this value can be done only on disabled datasource, requires a server restart otherwise.

**checkValidSql** : **STRING**

Specify an SQL statement to check validity of a pool connection. This may be called when managed connection is obtained from the pool

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

The deployable that this deployed is derived from.

**exceptionSorter** : **STRING**

An org.jboss.jca.adapters.jdbc.ExceptionSorter that provides an isExceptionFatal(SQLException) method to validate if an exception should broadcast an error

**maxPoolSize** : **INTEGER** = 0

The max-pool-size element specifies the maximum number of connections for a pool. No more connections will be created in each sub-pool

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

The min-pool-size element specifies the minimum number of connections for a pool

**password** : **STRING**

Specifies the password used when creating a new connection

**prefillEnabled** : **BOOLEAN** = false

Should the pool be prefilled. Changing this value can be done only on disabled datasource, requires a server restart otherwise.

**securityDomain** : **STRING**

Specifies the security domain which defines the javax.security.auth.Subject that are used to distinguish connections in the pool

**sharePreparedStatements** : **BOOLEAN**

Whether to share prepared statements, i.e. whether asking for same statement twice without closing uses the same underlying prepared statement

**staleConnectionChecker** : **STRING**

An org.jboss.jca.adapters.jdbc.StaleConnectionChecker that provides an isStaleConnection(SQLException) method which if it returns true will wrap the exception in an org.jboss.jca.adapters.jdbc.StaleConnectionException

**statementCacheSize** : **INTEGER** = -1

The number of prepared statements per connection in an LRU cache

**strictMinimum** : **BOOLEAN** = false

Specifies if the min-pool-size should be considered strictly

**username** : **STRING**

Specify the user name used when creating a new connection

**validConnectionChecker** : **STRING**

An org.jboss.jca.adapters.jdbc.ValidConnectionChecker that provides an isValidConnection(Connection) method to validate a connection. If an exception is returned that means the connection is invalid. This overrides the check-valid-connection-sql element

**validateOnMatch** : **BOOLEAN** = false

The validate-on-match element specifies if connection validation should be done when a connection factory attempts to match a managed connection. This is typically exclusive to the use of background validation

**validationMillis** : **INTEGER** = -1

The background-validation-millis element specifies the amount of time, in milliseconds, that background validation will run. Changing this value can be done only on disabled datasource, requires a server restart otherwise

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

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

**createScript** : **STRING**

Classpath to the script that is uploaded and executed on the generic container for the create operation.

**createVerb** : **STRING** = Create

Create Verb

**destroyOrder** : **INTEGER** = 40

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

**destroyVerb** : **STRING** = Destroy

Destroy Verb

**libraries** : **LIST\_OF\_STRING** = [jboss/dm/ds/datasource-lib.py]











Libraries
<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> = <b>Modify</b>
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> = <b>Modify</b>
Noop Verb
<b>destroyScript</b> : <b>STRING</b>
Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
<b>inspectScript</b> : <b>STRING</b>
Classpath to the script used to inspect the generic container.
<b>modifyScript</b> : <b>STRING</b>
Classpath to the script that is uploaded and executed on the generic container for the modify operation.
<b>noopScript</b> : <b>STRING</b>
Classpath to the script that is uploaded and executed on the generic container for the noop operation.

## jbosssdm.CliBasedContainer

**Hierarchy** udm.BaseContainer >> udm.BaseConfigurationItem

**Interfaces** udm.Taggable, [jbosssdm.CliManagedContainer](#), [jbosssdm.ApplicationContainer](#), [jbosssdm.ResourceContainer](#), udm.ConfigurationItem, [jbosssdm.CliManagingContainer](#), udm.Container, [overthere.HostContainer](#)

JBoss Cli Managed Container

Public Properties	
 <b>cliScriptPrefix</b> : <b>STRING</b> = <b>jboss-cli</b>	
JBoss CLI script prefix. Depending on the host, either an '.sh' or '.bat' will be appended to get the cli script name.	
 <b>home</b> : <b>STRING</b>	
JBoss home directory	
 <b>host</b> : <b>CI</b> < <a href="#">overthere.Host</a> >	
Host	
 <b>adminHostAddress</b> : <b>STRING</b> = <b>localhost</b>	
Host which is used to login to JBoss Native Administration, default is localhost	
 <b>enableDaemon</b> : <b>BOOLEAN</b> = <b>true</b>	
Connection to CLI is setup using a daemon. Set to false if host connection does not support streaming.	
 <b>password</b> : <b>STRING</b>	
Password which is used to login to JBoss Native Administration.	
 <b>port</b> : <b>INTEGER</b> = 9999	
TCP port which is used to login to JBoss Native Administration, default is 9999	
<b>tags</b> : <b>SET_OF_STRING</b>	
The tags to map deployables to containers.	
 <b>username</b> : <b>STRING</b>	
Username which is used to login to JBoss Native Administration.	
Hidden Properties	
<b>libraries</b> : <b>LIST_OF_STRING</b> = [ <a href="#">jboss/dm/library/runtime.py</a> ]	
List of python library scripts that should be automatically loaded when using a JBoss CLI script	


## jbosssdm.CliManagedDeployed

**Hierarchy** `udm.BaseDeployed >> udm.BaseConfigurationItem`

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

Base for all deployed that utilize the JBoss Cli for configuration

#### Public Properties

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

The container on which this deployed runs.

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

The deployable that this deployed is derived from.

#### Hidden Properties

**createOrder** : `INTEGER = 50`

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

**createScript** : `STRING`

Classpath to the script that is uploaded and executed on the generic container for the create operation.

**createVerb** : `STRING = Create`

Create Verb

**destroyOrder** : `INTEGER = 40`

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

**destroyVerb** : `STRING = Destroy`

Destroy Verb

**modifyOrder** : `INTEGER = 50`

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

**modifyVerb** : `STRING = Modify`

Modify Verb

**noopOrder** : `INTEGER = 50`

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

**noopVerb** : `STRING = Modify`

Noop Verb

**destroyScript** : `STRING`

Classpath to the script that is uploaded and executed on the generic container for the destroy operation.

**inspectScript** : `STRING`

Classpath to the script used to inspect the generic container.

**libraries** : `LIST_OF_STRING`

List of python library scripts that should be automatically loaded when using a JBoss CLI script.

**modifyScript** : `STRING`

Classpath to the script that is uploaded and executed on the generic container for the modify operation.

**noopScript** : `STRING`

Classpath to the script that is uploaded and executed on the generic container for the noop operation.


### jbosssdm.CliManagedDeployedArtifact

**Hierarchy** `jbosssdm.CliManagedDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem`

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

Base for all deployed artifacts that utilize the JBoss Cli for configuration

#### Public Properties

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

The container on which this deployed runs.

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

The deployable that this deployed is derived from.

**placeholders** : `MAP_STRING_STRING`

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

#### Hidden Properties

**createOrder** : `INTEGER = 50`

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

**createScript** : **STRING**

Classpath to the script that is uploaded and executed on the generic container for the create operation.

**createVerb** : **STRING = Create**

Create Verb

**destroyOrder** : **INTEGER = 40**

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

**destroyVerb** : **STRING = Destroy**

Destroy Verb

**modifyOrder** : **INTEGER = 50**

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

**modifyVerb** : **STRING = Modify**

Modify Verb

**noopOrder** : **INTEGER = 50**

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

**noopVerb** : **STRING = Modify**

Noop Verb

**destroyScript** : **STRING**

Classpath to the script that is uploaded and executed on the generic container for the destroy operation.

**inspectScript** : **STRING**

Classpath to the script used to inspect the generic container.

**libraries** : **LIST\_OF\_STRING**

List of python library scripts that should be automatically loaded when using a JBoss CLI script.

**modifyScript** : **STRING**

Classpath to the script that is uploaded and executed on the generic container for the modify operation.

**noopScript** : **STRING**

Classpath to the script that is uploaded and executed on the generic container for the noop operation.

## jbosssdm.DataSource

**Hierarchy** [jbosssdm.JeeDataSource](#) >> [jbosssdm.BaseDataSource](#) >>  
[jbosssdm.CliManagedDeployed](#) >> [udm.BaseDeployed](#) >>  
[udm.BaseConfigurationItem](#)

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

DataSource. This deployed is used when a [jbosssdm.DataSourceSpec](#) is specified in a package.

### Public Properties

**connectionUrl** : **STRING**

The JDBC driver connection URL



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

The container on which this deployed runs.

**driverName** : **STRING**

Defines the JDBC driver the datasource should use. It is a symbolic name matching the the name of installed driver. In case the driver is deployed as jar, the name is the name of deployment unit.

**jndiName** : **STRING**

Specifies the JNDI name for the datasource

**backgroundValidation** : **BOOLEAN = false**

An element to specify that connections should be validated on a background thread versus being validated prior to use. Changing this value can be done only on disabled datasource, requires a server restart otherwise.

**checkValidSql** : **STRING**

Specify an SQL statement to check validity of a pool connection. This may be called when managed connection is obtained from the pool

**connectionProperties** : **MAP\_STRING\_STRING**

JDBC connection properties

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

The deployable that this deployed is derived from.
<b>exceptionSorter</b> : <b>STRING</b>
An org.jboss.jca.adapters.jdbc.ExceptionSorter that provides an isExceptionFatal(SQLException) method to validate if an exception should broadcast an error
<b>maxPoolSize</b> : <b>INTEGER = 0</b>
The max-pool-size element specifies the maximum number of connections for a pool. No more connections will be created in each sub-pool
<b>minPoolSize</b> : <b>INTEGER = 0</b>
The min-pool-size element specifies the minimum number of connections for a pool
<b>newConnectionSql</b> : <b>STRING</b>
Specifies an SQL statement to execute whenever a connection is added to the connection pool
<b>password</b> : <b>STRING</b>
Specifies the password used when creating a new connection
<b>prefillEnabled</b> : <b>BOOLEAN = false</b>
Should the pool be prefilled. Changing this value can be done only on disabled datasource, requires a server restart otherwise.
<b>securityDomain</b> : <b>STRING</b>
Specifies the security domain which defines the javax.security.auth.Subject that are used to distinguish connections in the pool
<b>sharePreparedStatements</b> : <b>BOOLEAN</b>
Whether to share prepared statements, i.e. whether asking for same statement twice without closing uses the same underlying prepared statement
<b>staleConnectionChecker</b> : <b>STRING</b>
An org.jboss.jca.adapters.jdbc.StaleConnectionChecker that provides an isStaleConnection(SQLException) method which if it returns true will wrap the exception in an org.jboss.jca.adapters.jdbc.StaleConnectionException
<b>statementCacheSize</b> : <b>INTEGER = -1</b>
The number of prepared statements per connection in an LRU cache
<b>strictMinimum</b> : <b>BOOLEAN = false</b>
Specifies if the min-pool-size should be considered strictly
<b>transactionIsolation</b> : <b>STRING</b>
Set the java.sql.Connection transaction isolation level. Valid values are: TRANSACTION_READ_UNCOMMITTED, TRANSACTION_READ_COMMITTED, TRANSACTION_REPEATABLE_READ, TRANSACTION_SERIALIZABLE and TRANSACTION_NONE
<b>useCcm</b> : <b>BOOLEAN = false</b>
Enable the use of a cached connection manager
<b>useJta</b> : <b>BOOLEAN = false</b>
Enable JTA integration
<b>username</b> : <b>STRING</b>
Specify the user name used when creating a new connection
<b>validConnectionChecker</b> : <b>STRING</b>
An org.jboss.jca.adapters.jdbc.ValidConnectionChecker that provides an isValidConnection(Connection) method to validate a connection. If an exception is returned that means the connection is invalid. This overrides the check-valid-connection-sql element
<b>validateOnMatch</b> : <b>BOOLEAN = false</b>
The validate-on-match element specifies if connection validation should be done when a connection factory attempts to match a managed connection. This is typically exclusive to the use of background validation
<b>validationMillis</b> : <b>INTEGER = -1</b>
The background-validation-millis element specifies the amount of time, in milliseconds, that background validation will run. Changing this value can be done only on disabled datasource, requires a server restart otherwise

#### Hidden Properties

<b>createOrder</b> : <b>INTEGER = 50</b>
The order of the step in the step list for the create operation.
<b>createScript</b> : <b>STRING = jboss/dm/ds/create-datasource.py</b>
Create Script
<b>createVerb</b> : <b>STRING = Create</b>
Create Verb
<b>destroyOrder</b> : <b>INTEGER = 40</b>
The order of the step in the step list for the destroy operation.
<b>destroyScript</b> : <b>STRING = jboss/dm/ds/destroy-datasource.py</b>

Destroy Script
<b>destroyVerb</b> : <b>STRING</b> = <b>Destroy</b>
Destroy Verb
<b>inspectScript</b> : <b>STRING</b> = <b>jboss/dm/ds/inspect-datasource.py</b>
Inspect Script
<b>libraries</b> : <b>LIST_OF_STRING</b> = <b>[jboss/dm/ds/datasource-lib.py]</b>
Libraries
<b>modifyOrder</b> : <b>INTEGER</b> = <b>50</b>
The order of the step in the step list for the modify operation.
<b>modifyVerb</b> : <b>STRING</b> = <b>Modify</b>
Modify Verb
<b>noopOrder</b> : <b>INTEGER</b> = <b>50</b>
The order of the step in the step list for the noop operation.
<b>noopVerb</b> : <b>STRING</b> = <b>Modify</b>
Noop Verb
<b>modifyScript</b> : <b>STRING</b>
Classpath to the script that is uploaded and executed on the generic container for the modify operation.
<b>noopScript</b> : <b>STRING</b>
Classpath to the script that is uploaded and executed on the generic container for the noop operation.

## jbosssdm.DataSourceSpec

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

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

DataSource. This deployed is used when a `jbosssdm.DataSourceSpec` is specified in a package. (deployable)

Public Properties
<b>backgroundValidation</b> : <b>STRING</b>
An element to specify that connections should be validated on a background thread versus being validated prior to use. Changing this value can be done only on disabled datasource, requires a server restart otherwise. (boolean)
<b>checkValidSql</b> : <b>STRING</b>
Specify an SQL statement to check validity of a pool connection. This may be called when managed connection is obtained from the pool (string)
<b>connectionProperties</b> : <b>MAP_STRING_STRING</b>
JDBC connection properties (map_string_string)
<b>connectionUrl</b> : <b>STRING</b>
The JDBC driver connection URL (string)
<b>driverName</b> : <b>STRING</b>
Defines the JDBC driver the datasource should use. It is a symbolic name matching the the name of installed driver. In case the driver is deployed as jar, the name is the name of deployment unit. (string)
<b>exceptionSorter</b> : <b>STRING</b>
An <code>org.jboss.jca.adapters.jdbc.ExceptionSorter</code> that provides an <code>isExceptionFatal(SQLException)</code> method to validate if an exception should broadcast an error (string)
<b>jndiName</b> : <b>STRING</b>
Specifies the JNDI name for the datasource (string)
<b>maxPoolSize</b> : <b>STRING</b>
The max-pool-size element specifies the maximum number of connections for a pool. No more connections will be created in each sub-pool (integer)
<b>minPoolSize</b> : <b>STRING</b>
The min-pool-size element specifies the minimum number of connections for a pool (integer)
<b>newConnectionSql</b> : <b>STRING</b>
Specifies an SQL statement to execute whenever a connection is added to the connection pool (string)
<b>password</b> : <b>STRING</b>
Specifies the password used when creating a new connection (string)
<b>prefillEnabled</b> : <b>STRING</b>






Should the pool be prefilled. Changing this value can be done only on disabled datasource, requires a server restart otherwise. (boolean)
<b>securityDomain</b> : <b>STRING</b>
Specifies the security domain which defines the javax.security.auth.Subject that are used to distinguish connections in the pool (string)
<b>sharePreparedStatements</b> : <b>STRING</b>
Whether to share prepared statements, i.e. whether asking for same statement twice without closing uses the same underlying prepared statement (boolean)
<b>staleConnectionChecker</b> : <b>STRING</b>
An org.jboss.jca.adapters.jdbc.StaleConnectionChecker that provides an isStaleConnection(SQLException) method which if it returns true will wrap the exception in an org.jboss.jca.adapters.jdbc.StaleConnectionException (string)
<b>statementCacheSize</b> : <b>STRING</b>
The number of prepared statements per connection in an LRU cache (integer)
<b>strictMinimum</b> : <b>STRING</b>
Specifies if the min-pool-size should be considered strictly (boolean)
<b>tags</b> : <b>SET_OF_STRING</b>
The tags to map deployables to containers.
<b>transactionIsolation</b> : <b>STRING</b>
Set the java.sql.Connection transaction isolation level. Valid values are: TRANSACTION_READ_UNCOMMITTED, TRANSACTION_READ_COMMITTED, TRANSACTION_REPEATABLE_READ, TRANSACTION_SERIALIZABLE and TRANSACTION_NONE (string)
<b>useCcm</b> : <b>STRING</b>
Enable the use of a cached connection manager (boolean)
<b>useJta</b> : <b>STRING</b>
Enable JTA integration (boolean)
<b>username</b> : <b>STRING</b>
Specify the user name used when creating a new connection (string)
<b>validConnectionChecker</b> : <b>STRING</b>
An org.jboss.jca.adapters.jdbc.ValidConnectionChecker that provides an isValidConnection(Connection) method to validate a connection. If an exception is returned that means the connection is invalid. This overrides the check-valid-connection-sql element (string)
<b>validateOnMatch</b> : <b>STRING</b>
The validate-on-match element specifies if connection validation should be done when a connection factory attempts to match a managed connection. This is typically exclusive to the use of background validation (boolean)
<b>validationMillis</b> : <b>STRING</b>
The background-validation-millis element specifies the amount of time, in milliseconds, that background validation will run. Changing this value can be done only on disabled datasource, requires a server restart otherwise (integer)

## jbosscdm.Domain


**Hierarchy** [jbosscdm.CliBasedContainer](#) >> [udm.BaseContainer](#) >> [udm.BaseConfigurationItem](#)

**Interfaces** [udm.Taggable](#), [jbosscdm.CliManagedContainer](#), [jbosscdm.ApplicationContainer](#), [jbosscdm.ResourceContainer](#), [udm.ConfigurationItem](#), [udm.Container](#), [jbosscdm.CliManagingContainer](#), [overthere.HostContainer](#)

JBoss Cli Managed Container

Public Properties	
 <b>cliScriptPrefix</b>	: <b>STRING</b> = <b>jboss-cli</b>
JBoss CLI script prefix. Depending on the host, either an '.sh' or '.bat' will be appended to get the cli script name.	
 <b>home</b>	: <b>STRING</b>
JBoss home directory	
 <b>host</b>	: <b>CI</b> < <a href="#">overthere.Host</a> >
Host	
 <b>adminHostAddress</b>	: <b>STRING</b> = <b>localhost</b>
Host which is used to login to JBoss Native Administration, default is localhost	
 <b>enableDaemon</b>	: <b>BOOLEAN</b> = <b>true</b>

Connection to CLI is setup using a daemon. Set to false if host connection does not support streaming.

 **password** : **STRING**

Password which is used to login to JBoss Native Administration.

 **port** : **INTEGER** = 9999

TCP port which is used to login to JBoss Native Administration, default is 9999

 **profiles** : **LIST\_OF\_CI**<jbossdm.Profile>

Profiles defined in domain

 **serverGroups** : **LIST\_OF\_CI**<jbossdm.ServerGroup>

Server groups defined in domain

**tags** : **SET\_OF\_STRING**

The tags to map deployables to containers.

 **username** : **STRING**

Username which is used to login to JBoss Native Administration.

#### Hidden Properties

**libraries** : **LIST\_OF\_STRING** = [jboss/dm/library/runtime.py]

List of python library scripts that should be automatically loaded when using a JBoss CLI script

### jbossdm.Ear

**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** = true

Scan Placeholders

**tags** : **SET\_OF\_STRING**

The tags to map deployables to containers.

#### 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


### jbossdm.EarModule

**Hierarchy** jbossdm.CliManagedDeployedArtifact >> jbossdm.CliManagedDeployed >>  
udm.BaseDeployed >> udm.BaseConfigurationItem

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

Ear with values configured for a deployment

#### Public Properties

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

The container on which this deployed runs.

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

The deployable that this deployed is derived from.

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

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

#### Hidden Properties


<b>createOrder</b> : <b>INTEGER</b> = 50
The order of the step in the step list for the create operation.
<b>createScript</b> : <b>STRING</b> = <code>jboss/dm/application/install-ear.py</code>
Create Script
<b>createVerb</b> : <b>STRING</b> = <code>Create</code>
Create Verb
<b>destroyOrder</b> : <b>INTEGER</b> = 40
The order of the step in the step list for the destroy operation.
<b>destroyScript</b> : <b>STRING</b> = <code>jboss/dm/application/uninstall-ear.py</code>
Destroy Script
<b>destroyVerb</b> : <b>STRING</b> = <code>Destroy</code>
Destroy Verb
<b>inspectScript</b> : <b>STRING</b> = <code>jboss/dm/application/inspect-ear.py</code>
Inspect Script
<b>libraries</b> : <b>LIST_OF_STRING</b> = <code>[jboss/dm/application/application-lib.py]</code>
Libraries
<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> = <code>Modify</code>
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> = <code>Modify</code>
Noop Verb
<b>modifyScript</b> : <b>STRING</b>
Classpath to the script that is uploaded and executed on the generic container for the modify operation.
<b>noopScript</b> : <b>STRING</b>
Classpath to the script that is uploaded and executed on the generic container for the noop operation.

## jbosssdm.JeeDataSource

**Hierarchy** `jbosssdm.BaseDataSource` >> `jbosssdm.CliManagedDeployed` >>  
`udm.BaseDeployed` >> `udm.BaseConfigurationItem`

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

Datasource. This deployed is used when a `jee.DataSourceSpec` is specified in a package.

Public Properties
<b>connectionUrl</b> : <b>STRING</b>
The JDBC driver connection URL
 <b>container</b> : <b>CI</b> < <code>udm.Container</code> >
The container on which this deployed runs.
<b>driverName</b> : <b>STRING</b>
Defines the JDBC driver the datasource should use. It is a symbolic name matching the the name of installed driver. In case the driver is deployed as jar, the name is the name of deployment unit.
<b>jndiName</b> : <b>STRING</b>
Specifies the JNDI name for the datasource
<b>backgroundValidation</b> : <b>BOOLEAN</b> = <code>false</code>
An element to specify that connections should be validated on a background thread versus being validated prior to use. Changing this value can be done only on disabled datasource, requires a server restart otherwise.
<b>checkValidSql</b> : <b>STRING</b>
Specify an SQL statement to check validity of a pool connection. This may be called when managed connection is obtained from the pool
<b>connectionProperties</b> : <b>MAP_STRING_STRING</b>
JDBC connection properties
<b>deployable</b> : <b>CI</b> < <code>udm.Deployable</code> >
The deployable that this deployed is derived from.
<b>exceptionSorter</b> : <b>STRING</b>
An <code>org.jboss.jca.adapters.jdbc.ExceptionSorter</code> that provides an



isExceptionFatal(SQLException) method to validate if an exception should broadcast an error
<b>maxPoolSize</b> : <b>INTEGER</b> = 0
The max-pool-size element specifies the maximum number of connections for a pool. No more connections will be created in each sub-pool
<b>minPoolSize</b> : <b>INTEGER</b> = 0
The min-pool-size element specifies the minimum number of connections for a pool
<b>newConnectionSql</b> : <b>STRING</b>
Specifies an SQL statement to execute whenever a connection is added to the connection pool
<b>password</b> : <b>STRING</b>
Specifies the password used when creating a new connection
<b>prefillEnabled</b> : <b>BOOLEAN</b> = false
Should the pool be prefilled. Changing this value can be done only on disabled datasource, requires a server restart otherwise.
<b>securityDomain</b> : <b>STRING</b>
Specifies the security domain which defines the javax.security.auth.Subject that are used to distinguish connections in the pool
<b>sharePreparedStatements</b> : <b>BOOLEAN</b>
Whether to share prepared statements, i.e. whether asking for same statement twice without closing uses the same underlying prepared statement
<b>staleConnectionChecker</b> : <b>STRING</b>
An org.jboss.jca.adapters.jdbc.StaleConnectionChecker that provides an isStaleConnection(SQLException) method which if it returns true will wrap the exception in an org.jboss.jca.adapters.jdbc.StaleConnectionException
<b>statementCacheSize</b> : <b>INTEGER</b> = -1
The number of prepared statements per connection in an LRU cache
<b>strictMinimum</b> : <b>BOOLEAN</b> = false
Specifies if the min-pool-size should be considered strictly
<b>transactionIsolation</b> : <b>STRING</b>
Set the java.sql.Connection transaction isolation level. Valid values are: TRANSACTION_READ_UNCOMMITTED, TRANSACTION_READ_COMMITTED, TRANSACTION_REPEATABLE_READ, TRANSACTION_SERIALIZABLE and TRANSACTION_NONE
<b>useCcm</b> : <b>BOOLEAN</b> = false
Enable the use of a cached connection manager
<b>useJta</b> : <b>BOOLEAN</b> = false
Enable JTA integration
<b>username</b> : <b>STRING</b>
Specify the user name used when creating a new connection
<b>validConnectionChecker</b> : <b>STRING</b>
An org.jboss.jca.adapters.jdbc.ValidConnectionChecker that provides an isValidConnection(Connection) method to validate a connection. If an exception is returned that means the connection is invalid. This overrides the check-valid-connection-sql element
<b>validateOnMatch</b> : <b>BOOLEAN</b> = false
The validate-on-match element specifies if connection validation should be done when a connection factory attempts to match a managed connection. This is typically exclusive to the use of background validation
<b>validationMillis</b> : <b>INTEGER</b> = -1
The background-validation-millis element specifies the amount of time, in milliseconds, that background validation will run. Changing this value can be done only on disabled datasource, requires a server restart otherwise

#### Hidden Properties

<b>createOrder</b> : <b>INTEGER</b> = 50
The order of the step in the step list for the create operation.
<b>createScript</b> : <b>STRING</b> = jboss/dm/ds/create-datasource.py
Create Script
<b>createVerb</b> : <b>STRING</b> = Create
Create Verb
<b>destroyOrder</b> : <b>INTEGER</b> = 40
The order of the step in the step list for the destroy operation.
<b>destroyScript</b> : <b>STRING</b> = jboss/dm/ds/destroy-datasource.py
Destroy Script
<b>destroyVerb</b> : <b>STRING</b> = Destroy


Destroy Verb
<b>inspectScript</b> : <b>STRING</b> = <code>jboss/dm/ds/inspect-datasource.py</code>
Inspect Script
<b>libraries</b> : <b>LIST_OF_STRING</b> = <code>[jboss/dm/ds/datasource-lib.py]</code>
Libraries
<b>modifyOrder</b> : <b>INTEGER</b> = <code>50</code>
The order of the step in the step list for the modify operation.
<b>modifyVerb</b> : <b>STRING</b> = <code>Modify</code>
Modify Verb
<b>noopOrder</b> : <b>INTEGER</b> = <code>50</code>
The order of the step in the step list for the noop operation.
<b>noopVerb</b> : <b>STRING</b> = <code>Modify</code>
Noop Verb
<b>modifyScript</b> : <b>STRING</b>
Classpath to the script that is uploaded and executed on the generic container for the modify operation.
<b>noopScript</b> : <b>STRING</b>
Classpath to the script that is uploaded and executed on the generic container for the noop operation.

## jbosssdm.JeeXaDataSource

**Hierarchy** `jbosssdm.BaseDataSource` >> `jbosssdm.CliManagedDeployed` >> `udm.BaseDeployed` >> `udm.BaseConfigurationItem`

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

XA DataSource. This deployed is used when a `jee.DataSourceSpec` is specified in a package.

Public Properties	
 <b>container</b> : <b>CI</b> < <code>udm.Container</code> >	The container on which this deployed runs.
<b>driverName</b> : <b>STRING</b>	
	Defines the JDBC driver the datasource should use. It is a symbolic name matching the the name of installed driver. In case the driver is deployed as jar, the name is the name of deployment unit.
<b>jndiName</b> : <b>STRING</b>	
	Specifies the JNDI name for the datasource
<b>xaProperties</b> : <b>MAP_STRING_STRING</b>	
	Properties to assign to the XADataSource implementation class. At least one XA property is required (i.e. url)
<b>backgroundValidation</b> : <b>BOOLEAN</b> = <code>false</code>	
	An element to specify that connections should be validated on a background thread versus being validated prior to use. Changing this value can be done only on disabled datasource, requires a server restart otherwise.
<b>checkValidSql</b> : <b>STRING</b>	
	Specify an SQL statement to check validity of a pool connection. This may be called when managed connection is obtained from the pool
<b>deployable</b> : <b>CI</b> < <code>udm.Deployable</code> >	
	The deployable that this deployed is derived from.
<b>exceptionSorter</b> : <b>STRING</b>	
	An <code>org.jboss.jca.adapters.jdbc.ExceptionSorter</code> that provides an <code>isExceptionFatal(SQLException)</code> method to validate if an exception should broadcast an error
<b>interleave</b> : <b>BOOLEAN</b> = <code>false</code>	
	An element to enable interleaving for XA connections
<b>maxPoolSize</b> : <b>INTEGER</b> = <code>0</code>	
	The max-pool-size element specifies the maximum number of connections for a pool. No more connections will be created in each sub-pool
<b>minPoolSize</b> : <b>INTEGER</b> = <code>0</code>	
	The min-pool-size element specifies the minimum number of connections for a pool
<b>newConnectionSql</b> : <b>STRING</b>	
	Specifies an SQL statement to execute whenever a connection is added to the connection pool
<b>padXid</b> : <b>BOOLEAN</b> = <code>false</code>	

Should the Xid be padded
<b>password</b> : <b>STRING</b>
Specifies the password used when creating a new connection
<b>prefillEnabled</b> : <b>BOOLEAN</b> = <b>false</b>
Should the pool be prefilled. Changing this value can be done only on disabled datasource, requires a server restart otherwise.
<b>sameRmOverride</b> : <b>BOOLEAN</b> = <b>false</b>
The is-same-rm-override element allows one to unconditionally set whether the <code>javax.transaction.xa.XAResource.isSameRM(XAResource)</code> returns true or false
<b>securityDomain</b> : <b>STRING</b>
Specifies the security domain which defines the <code>javax.security.auth.Subject</code> that are used to distinguish connections in the pool
<b>sharePreparedStatements</b> : <b>BOOLEAN</b>
Whether to share prepared statements, i.e. whether asking for same statement twice without closing uses the same underlying prepared statement
<b>staleConnectionChecker</b> : <b>STRING</b>
An <code>org.jboss.jca.adapters.jdbc.StaleConnectionChecker</code> that provides an <code>isStaleConnection(SQLException)</code> method which if it returns true will wrap the exception in an <code>org.jboss.jca.adapters.jdbc.StaleConnectionException</code>
<b>statementCacheSize</b> : <b>INTEGER</b> = <b>-1</b>
The number of prepared statements per connection in an LRU cache
<b>strictMinimum</b> : <b>BOOLEAN</b> = <b>false</b>
Specifies if the min-pool-size should be considered strictly
<b>transactionIsolation</b> : <b>STRING</b>
Set the <code>java.sql.Connection</code> transaction isolation level. Valid values are: <code>TRANSACTION_READ_UNCOMMITTED</code> , <code>TRANSACTION_READ_COMMITTED</code> , <code>TRANSACTION_REPEATABLE_READ</code> , <code>TRANSACTION_SERIALIZABLE</code> and <code>TRANSACTION_NONE</code>
<b>username</b> : <b>STRING</b>
Specify the user name used when creating a new connection
<b>validConnectionChecker</b> : <b>STRING</b>
An <code>org.jboss.jca.adapters.jdbc.ValidConnectionChecker</code> that provides an <code>isValidConnection(Connection)</code> method to validate a connection. If an exception is returned that means the connection is invalid. This overrides the <code>check-valid-connection-sql</code> element
<b>validateOnMatch</b> : <b>BOOLEAN</b> = <b>false</b>
The <code>validate-on-match</code> element specifies if connection validation should be done when a connection factory attempts to match a managed connection. This is typically exclusive to the use of background validation
<b>validationMillis</b> : <b>INTEGER</b> = <b>-1</b>
The <code>background-validation-millis</code> element specifies the amount of time, in milliseconds, that background validation will run. Changing this value can be done only on disabled datasource, requires a server restart otherwise
<b>wrapXa</b> : <b>BOOLEAN</b> = <b>false</b>
Should the <code>XAResource</code> instances be wrapped in a <code>org.jboss.tm.XAResourceWrapper</code> instance


Hidden Properties
<b>createOrder</b> : <b>INTEGER</b> = <b>50</b>
The order of the step in the step list for the create operation.
<b>createScript</b> : <b>STRING</b> = <code>jboss/dm/ds/create-xa-datasource.py</code>
Create Script
<b>createVerb</b> : <b>STRING</b> = <b>Create</b>
Create Verb
<b>destroyOrder</b> : <b>INTEGER</b> = <b>40</b>
The order of the step in the step list for the destroy operation.
<b>destroyScript</b> : <b>STRING</b> = <code>jboss/dm/ds/destroy-xa-datasource.py</code>
Destroy Script
<b>destroyVerb</b> : <b>STRING</b> = <b>Destroy</b>
Destroy Verb
<b>inspectScript</b> : <b>STRING</b> = <code>jboss/dm/ds/inspect-xa-datasource.py</code>
Inspect Script
<b>libraries</b> : <b>LIST_OF_STRING</b> = <code>[jboss/dm/ds/datasource-lib.py]</code>
Libraries
<b>modifyOrder</b> : <b>INTEGER</b> = <b>50</b>
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>modifyScript</b>	: <b>STRING</b>
Classpath to the script that is uploaded and executed on the generic container for the modify operation.	
<b>noopScript</b>	: <b>STRING</b>
Classpath to the script that is uploaded and executed on the generic container for the noop operation.	

## jbosssdm.Profile

<b>Hierarchy</b>	udm.BaseContainer >> udm.BaseConfigurationItem
<b>Interfaces</b>	udm.Tagable, <a href="#">jbosssdm.CliManagedContainer</a> , <a href="#">jbosssdm.ResourceContainer</a> , udm.ConfigurationItem, udm.Container, overthere.HostContainer


JBoss Profile

Public Properties	
 <b>domain</b>	: <b>CI</b> < <a href="#">jbosssdm.Domain</a> >
Domain to which the server group belongs.	
<b>tags</b>	: <b>SET_OF_STRING</b>
The tags to map deployables to containers.	

## jbosssdm.Queue

<b>Hierarchy</b>	<a href="#">jbosssdm.CliManagedDeployed</a> >> udm.BaseDeployed >> udm.BaseConfigurationItem
<b>Interfaces</b>	udm.Deployed, udm.ConfigurationItem

A Jboss Queue

Public Properties	
 <b>container</b>	: <b>CI</b> < <a href="#">udm.Container</a> >
The container on which this deployed runs.	
<b>jndiName</b>	: <b>STRING</b>
(Comma separated list) The jndi names the queue will be bound to.	
<b>deployable</b>	: <b>CI</b> < <a href="#">udm.Deployable</a> >
The deployable that this deployed is derived from.	
<b>durable</b>	: <b>BOOLEAN</b> = true
Whether the queue is durable or not	
<b>selector</b>	: <b>STRING</b>
The queue selector	

Hidden Properties	
<b>createOrder</b>	: <b>INTEGER</b> = 50
The order of the step in the step list for the create operation.	
<b>createScript</b>	: <b>STRING</b> = <a href="#">jboss/dm/jms/create-queue.py</a>
Create Script	
<b>createVerb</b>	: <b>STRING</b> = Create
Create Verb	
<b>destroyOrder</b>	: <b>INTEGER</b> = 40
The order of the step in the step list for the destroy operation.	
<b>destroyScript</b>	: <b>STRING</b> = <a href="#">jboss/dm/jms/destroy-queue.py</a>
Destroy Script	
<b>destroyVerb</b>	: <b>STRING</b> = Destroy
Destroy Verb	
<b>inspectScript</b>	: <b>STRING</b> = <a href="#">jboss/dm/jms/inspect-queue.py</a>
Inspect Script	
<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> = <b>Modify</b>
Modify Verb
<b>noopOrder</b> : <b>INTEGER</b> = <b>50</b>
The order of the step in the step list for the noop operation.
<b>noopVerb</b> : <b>STRING</b> = <b>Modify</b>
Noop Verb
<b>libraries</b> : <b>LIST_OF_STRING</b>
List of python library scripts that should be automatically loaded when using a JBoss CLI script.
<b>modifyScript</b> : <b>STRING</b>
Classpath to the script that is uploaded and executed on the generic container for the modify operation.
<b>noopScript</b> : <b>STRING</b>
Classpath to the script that is uploaded and executed on the generic container for the noop operation.

## jbosssdm.QueueSpec

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

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

A Queue


Public Properties
<b> durable</b> : <b>STRING</b>
Whether the queue is durable or not (boolean)
<b> jndiName</b> : <b>STRING</b>
(Comma separated list) The jndi names the queue will be bound to. (string)
<b> selector</b> : <b>STRING</b>
The queue selector (string)
<b> tags</b> : <b>SET_OF_STRING</b>
The tags to map deployables to containers.

## jbosssdm.ServerGroup

**Hierarchy** `udm.BaseContainer >> udm.BaseConfigurationItem`

**Interfaces** `udm.Taggable, jbosssdm.CliManagedContainer, jbosssdm.ApplicationContainer, udm.ConfigurationItem, udm.Container, overthere.HostContainer`

JBoss Server Group



Public Properties
 <b> domain</b> : <b>CI</b> < <code>jbosssdm.Domain</code> >
Domain to which the server group belongs.
<b> tags</b> : <b>SET_OF_STRING</b>
The tags to map deployables to containers.







## jbosssdm.StandaloneServer

**Hierarchy** `jbosssdm.CliBasedContainer >> udm.BaseContainer >> udm.BaseConfigurationItem`

**Interfaces** `udm.Taggable, jbosssdm.CliManagedContainer, jbosssdm.ApplicationContainer, jbosssdm.ResourceContainer, udm.ConfigurationItem, udm.Container, jbosssdm.CliManagingContainer, overthere.HostContainer`

JBoss Standalone Server

Public Properties
 <b> cliScriptPrefix</b> : <b>STRING</b> = <code>jboss-cli</code>
JBoss CLI script prefix. Depending on the host, either an '.sh' or '.bat' will be appended to get the cli script name.
 <b> home</b> : <b>STRING</b>
JBoss home directory

	<b>host</b> : <code>CI&lt;overthere.Host&gt;</code>
Host	
	<b>adminHostAddress</b> : <code>STRING = localhost</code>
Host which is used to login to JBoss Native Administration, default is localhost	
	<b>enableDaemon</b> : <code>BOOLEAN = true</code>
Connection to CLI is setup using a daemon. Set to false if host connection does not support streaming.	
	<b>password</b> : <code>STRING</code>
Password which is used to login to JBoss Native Administration.	
	<b>port</b> : <code>INTEGER = 9999</code>
TCP port which is used to login to JBoss Native Administration, default is 9999	
<b>tags</b> : <code>SET_OF_STRING</code>	
The tags to map deployables to containers.	
	<b>username</b> : <code>STRING</code>
Username which is used to login to JBoss Native Administration.	

#### Hidden Properties

<b>libraries</b>	: <code>LIST_OF_STRING = [jboss/dm/library/runtime.py]</code>
List of python library scripts that should be automatically loaded when using a JBoss CLI script	


### jbossdm.Topic

**Hierarchy** `jbossdm.CliManagedDeployed` >> `udm.BaseDeployed` >> `udm.BaseConfigurationItem`

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

A JBoss topic

#### Public Properties

	<b>container</b> : <code>CI&lt;udm.Container&gt;</code>
The container on which this deployed runs.	
<b>jndiName</b>	: <code>STRING</code>
(Comma separated list) The jndi names the topic will be bound to.	
<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>
The order of the step in the step list for the create operation.	
<b>createScript</b>	: <code>STRING = jboss/dm/jms/create-topic.py</code>
Create Script	
<b>createVerb</b>	: <code>STRING = Create</code>
Create Verb	
<b>destroyOrder</b>	: <code>INTEGER = 40</code>
The order of the step in the step list for the destroy operation.	
<b>destroyScript</b>	: <code>STRING = jboss/dm/jms/destroy-topic.py</code>
Destroy Script	
<b>destroyVerb</b>	: <code>STRING = Destroy</code>
Destroy Verb	
<b>inspectScript</b>	: <code>STRING = jboss/dm/jms/inspect-topic.py</code>
Inspect Script	
<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.	

**noopVerb** : **STRING** = Modify

Noop Verb

**libraries** : **LIST\_OF\_STRING**

List of python library scripts that should be automatically loaded when using a JBoss CLI script.

**modifyScript** : **STRING**

Classpath to the script that is uploaded and executed on the generic container for the modify operation.

**noopScript** : **STRING**

Classpath to the script that is uploaded and executed on the generic container for the noop operation.

## jbosssdm.TopicSpec

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

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

A Topic

### Public Properties

**jndiName** : **STRING**

(Comma separated list) The jndi names the topic will be bound to. (string)

**tags** : **SET\_OF\_STRING**

The tags to map deployables to containers.

## jbosssdm.War

**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

**contextRoot** : **STRING**

Context root for the web application (string)

**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

Scan Placeholders

**tags** : **SET\_OF\_STRING**

The tags to map deployables to containers.

### 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

## jbosssdm.WarModule

**Hierarchy** `jbosssdm.CliManagedDeployedArtifact >> jbosssdm.CliManagedDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem`

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

War with values configured for a deployment

### Public Properties



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

The container on which this deployed runs.

**contextRoot** : **STRING**

Context root for the web application

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

The deployable that this deployed is derived from.

**placeholders** : `MAP_STRING_STRING`

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

#### Hidden Properties

**createOrder** : `INTEGER = 50`

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

**createScript** : `STRING = jboss/dm/application/install-war.py`

Create Script

**createVerb** : `STRING = Create`

Create Verb

**destroyOrder** : `INTEGER = 40`

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

**destroyScript** : `STRING = jboss/dm/application/uninstall-war.py`

Destroy Script

**destroyVerb** : `STRING = Destroy`

Destroy Verb

**inspectScript** : `STRING = jboss/dm/application/inspect-war.py`

Inspect Script

**libraries** : `LIST_OF_STRING = [jboss/dm/application/application-lib.py]`

Libraries

**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

**modifyScript** : `STRING`

Classpath to the script that is uploaded and executed on the generic container for the modify operation.

**noopScript** : `STRING`

Classpath to the script that is uploaded and executed on the generic container for the noop operation.

## jbosssdm.XaDataSource

**Hierarchy** `jbosssdm.JeeXaDataSource >> jbosssdm.BaseDataSource >> jbosssdm.CliManagedDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem`

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

XA DataSource. This deployed is used when a `jbosssdm.XaDataSourceSpec` is specified in a package.

#### Public Properties



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

The container on which this deployed runs.

**driverName** : `STRING`

Defines the JDBC driver the datasource should use. It is a symbolic name matching the the name of installed driver. In case the driver is deployed as jar, the name is the name of deployment unit.

**jndiName** : `STRING`

Specifies the JNDI name for the datasource

**xaProperties** : `MAP_STRING_STRING`

Properties to assign to the XADataSource implementation class. At least one XA property is required (i.e. url)

**backgroundValidation** : `BOOLEAN = false`

An element to specify that connections should be validated on a background thread versus being validated prior to use. Changing this value can be done only on disabled datasource, requires a server restart otherwise.

**checkValidSql** : `STRING`



Specify an SQL statement to check validity of a pool connection. This may be called when managed connection is obtained from the pool <b>deployable</b> : <code>CI&lt;udm.Deployable&gt;</code>
The deployable that this deployed is derived from. <b>exceptionSorter</b> : <code>STRING</code>
An <code>org.jboss.jca.adapters.jdbc.ExceptionSorter</code> that provides an <code>isExceptionFatal(SQLException)</code> method to validate if an exception should broadcast an error <b>interleave</b> : <code>BOOLEAN = false</code>
An element to enable interleaving for XA connections <b>maxPoolSize</b> : <code>INTEGER = 0</code>
The max-pool-size element specifies the maximum number of connections for a pool. No more connections will be created in each sub-pool <b>minPoolSize</b> : <code>INTEGER = 0</code>
The min-pool-size element specifies the minimum number of connections for a pool <b>newConnectionSql</b> : <code>STRING</code>
Specifies an SQL statement to execute whenever a connection is added to the connection pool <b>padXid</b> : <code>BOOLEAN = false</code>
Should the Xid be padded <b>password</b> : <code>STRING</code>
Specifies the password used when creating a new connection <b>prefillEnabled</b> : <code>BOOLEAN = false</code>
Should the pool be prefilled. Changing this value can be done only on disabled datasource, requires a server restart otherwise. <b>sameRmOverride</b> : <code>BOOLEAN = false</code>
The is-same-rm-override element allows one to unconditionally set whether the <code>javax.transaction.xa.XAResource.isSameRM(XAResource)</code> returns true or false <b>securityDomain</b> : <code>STRING</code>
Specifies the security domain which defines the <code>javax.security.auth.Subject</code> that are used to distinguish connections in the pool <b>sharePreparedStatements</b> : <code>BOOLEAN</code>
Whether to share prepared statements, i.e. whether asking for same statement twice without closing uses the same underlying prepared statement <b>staleConnectionChecker</b> : <code>STRING</code>
An <code>org.jboss.jca.adapters.jdbc.StaleConnectionChecker</code> that provides an <code>isStaleConnection(SQLException)</code> method which if it returns true will wrap the exception in an <code>org.jboss.jca.adapters.jdbc.StaleConnectionException</code> <b>statementCacheSize</b> : <code>INTEGER = -1</code>
The number of prepared statements per connection in an LRU cache <b>strictMinimum</b> : <code>BOOLEAN = false</code>
Specifies if the min-pool-size should be considered strictly <b>transactionIsolation</b> : <code>STRING</code>
Set the <code>java.sql.Connection</code> transaction isolation level. Valid values are: <code>TRANSACTION_READ_UNCOMMITTED</code> , <code>TRANSACTION_READ_COMMITTED</code> , <code>TRANSACTION_REPEATABLE_READ</code> , <code>TRANSACTION_SERIALIZABLE</code> and <code>TRANSACTION_NONE</code> <b>username</b> : <code>STRING</code>
Specify the user name used when creating a new connection <b>validConnectionChecker</b> : <code>STRING</code>
An <code>org.jboss.jca.adapters.jdbc.ValidConnectionChecker</code> that provides an <code>isValidConnection(Connection)</code> method to validate a connection. If an exception is returned that means the connection is invalid. This overrides the <code>check-valid-connection-sql</code> element <b>validateOnMatch</b> : <code>BOOLEAN = false</code>
The validate-on-match element specifies if connection validation should be done when a connection factory attempts to match a managed connection. This is typically exclusive to the use of background validation <b>validationMillis</b> : <code>INTEGER = -1</code>
The background-validation-millis element specifies the amount of time, in milliseconds, that background validation will run. Changing this value can be done only on disabled datasource, requires a server restart otherwise <b>wrapXa</b> : <code>BOOLEAN = false</code>
Should the <code>XAResource</code> instances be wrapped in a <code>org.jboss.tm.XAResourceWrapper</code> instance

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

The order of the step in the step list for the create operation.
<b>createScript</b> : <code>STRING</code> = <code>jboss/dm/ds/create-xa-datasource.py</code>
Create Script
<b>createVerb</b> : <code>STRING</code> = <code>Create</code>
Create Verb
<b>destroyOrder</b> : <code>INTEGER</code> = <code>40</code>
The order of the step in the step list for the destroy operation.
<b>destroyScript</b> : <code>STRING</code> = <code>jboss/dm/ds/destroy-xa-datasource.py</code>
Destroy Script
<b>destroyVerb</b> : <code>STRING</code> = <code>Destroy</code>
Destroy Verb
<b>inspectScript</b> : <code>STRING</code> = <code>jboss/dm/ds/inspect-xa-datasource.py</code>
Inspect Script
<b>libraries</b> : <code>LIST_OF_STRING</code> = <code>[jboss/dm/ds/datasource-lib.py]</code>
Libraries
<b>modifyOrder</b> : <code>INTEGER</code> = <code>50</code>
The order of the step in the step list for the modify operation.
<b>modifyVerb</b> : <code>STRING</code> = <code>Modify</code>
Modify Verb
<b>noopOrder</b> : <code>INTEGER</code> = <code>50</code>
The order of the step in the step list for the noop operation.
<b>noopVerb</b> : <code>STRING</code> = <code>Modify</code>
Noop Verb
<b>modifyScript</b> : <code>STRING</code>
Classpath to the script that is uploaded and executed on the generic container for the modify operation.
<b>noopScript</b> : <code>STRING</code>
Classpath to the script that is uploaded and executed on the generic container for the noop operation.

## jbosssdm.XaDataSourceSpec

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

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

XA DataSource. This deployed is used when a `jbosssdm.XaDataSourceSpec` is specified in a package. (deployable)

Public Properties
<b>backgroundValidation</b> : <code>STRING</code>
An element to specify that connections should be validated on a background thread versus being validated prior to use. Changing this value can be done only on disabled datasource, requires a server restart otherwise. (boolean)
<b>checkValidSql</b> : <code>STRING</code>
Specify an SQL statement to check validity of a pool connection. This may be called when managed connection is obtained from the pool (string)
<b>driverName</b> : <code>STRING</code>
Defines the JDBC driver the datasource should use. It is a symbolic name matching the the name of installed driver. In case the driver is deployed as jar, the name is the name of deployment unit. (string)
<b>exceptionSorter</b> : <code>STRING</code>
An <code>org.jboss.jca.adapters.jdbc.ExceptionSorter</code> that provides an <code>isExceptionFatal(SQLException)</code> method to validate if an exception should broadcast an error (string)
<b>interleave</b> : <code>STRING</code>
An element to enable interleaving for XA connections (boolean)
<b>jndiName</b> : <code>STRING</code>
Specifies the JNDI name for the datasource (string)
<b>maxPoolSize</b> : <code>STRING</code>
The max-pool-size element specifies the maximum number of connections for a pool. No more connections will be created in each sub-pool (integer)
<b>minPoolSize</b> : <code>STRING</code>
The min-pool-size element specifies the minimum number of connections for a pool (integer)

**newConnectionSql** : **STRING**

Specifies an SQL statement to execute whenever a connection is added to the connection pool (string)

**padXid** : **STRING**

Should the Xid be padded (boolean)

**password** : **STRING**

Specifies the password used when creating a new connection (string)

**prefillEnabled** : **STRING**

Should the pool be prefilled. Changing this value can be done only on disabled datasource, requires a server restart otherwise. (boolean)

**sameRmOverride** : **STRING**

The is-same-rm-override element allows one to unconditionally set whether the javax.transaction.xa.XAResource.isSameRM(XAResource) returns true or false (boolean)

**securityDomain** : **STRING**

Specifies the security domain which defines the javax.security.auth.Subject that are used to distinguish connections in the pool (string)

**sharePreparedStatements** : **STRING**

Whether to share prepared statements, i.e. whether asking for same statement twice without closing uses the same underlying prepared statement (boolean)

**staleConnectionChecker** : **STRING**

An org.jboss.jca.adapters.jdbc.StaleConnectionChecker that provides an isStaleConnection(SQLException) method which if it returns true will wrap the exception in an org.jboss.jca.adapters.jdbc.StaleConnectionException (string)

**statementCacheSize** : **STRING**

The number of prepared statements per connection in an LRU cache (integer)

**strictMinimum** : **STRING**

Specifies if the min-pool-size should be considered strictly (boolean)

**tags** : **SET\_OF\_STRING**

The tags to map deployables to containers.

**transactionIsolation** : **STRING**

Set the java.sql.Connection transaction isolation level. Valid values are: TRANSACTION\_READ\_UNCOMMITTED, TRANSACTION\_READ\_COMMITTED, TRANSACTION\_REPEATABLE\_READ, TRANSACTION\_SERIALIZABLE and TRANSACTION\_NONE (string)

**username** : **STRING**

Specify the user name used when creating a new connection (string)

**validConnectionChecker** : **STRING**

An org.jboss.jca.adapters.jdbc.ValidConnectionChecker that provides an isValidConnection(Connection) method to validate a connection. If an exception is returned that means the connection is invalid. This overrides the check-valid-connection-sql element (string)

**validateOnMatch** : **STRING**

The validate-on-match element specifies if connection validation should be done when a connection factory attempts to match a managed connection. This is typically exclusive to the use of background validation (boolean)

**validationMillis** : **STRING**

The background-validation-millis element specifies the amount of time, in milliseconds, that background validation will run. Changing this value can be done only on disabled datasource, requires a server restart otherwise (integer)

**wrapXa** : **STRING**

Should the XAResource instances be wrapped in a org.jboss.tm.XAResourceWrapper instance (boolean)

**xaProperties** : **MAP\_STRING\_STRING**

Properties to assign to the XADataSource implementation class. At least one XA property is required (i.e. url) (map\_string\_string)