

Deployit IBM WebSphere Application Server Plugin Manual

Version 3.7.0

Table of Content

Preface	4
Overview	4
Features	4
Requirements	4
Usage in Deployment Packages	4
Using the deployables and deployed	5
Deployable vs. containers table	5
Deployed actions table	5
Discovering WAS middleware	6
Create a Configuration Item starting point	7
Start discovery passing a Configuration Item	7
Store the CIs in the repository	7
Complete discovered middleware CIs	8
Adding CIs to Environments	8
Extending the WAS plugin	8
Extending using XML and Python scripts	9
Adding a property	10
Fixing a property	10
Inspection	10
Execution order	11
CI Reference	11
Configuration Item Overview	11
Deployable Configuration Items	11
Deployed Configuration Items	12
Topology Configuration Items	13
Virtual Deployable Configuration Items	14
Virtual Deployed Configuration Items	14
Virtual Topology Configuration Items	14
Configuration Item Details	14
was.BaseCell	14
was.Cell	15
was.Cluster	15
was.DB2Datasource	16
was.DB2Type2Datasource	18
was.DB2Type2DatasourceSpec	21
was.DB2Type4Datasource	22
was.DB2Type4DatasourceSpec	25
was.Datasource	26
was.Deployable	29
was.DeploymentManager	29
was.Ear	30
was.EarModule	31
was.EjbJar	34
was.EjbModule	34
was.ExtensibleDeployed	37
was.ExtensibleDeployedArtifact	39
was.ExtensibleDeployedResource	41
was.JmsResource	43
was.ManagedServer	45
was.ManagedWebServer	45
was.Module	45
was.Node	48
was.NodeAgent	48
was.OracleDatasource	48
was.OracleDatasourceSpec	51
was.Resource	52
was.Server	54

was.SharedLibrary	54
was.SharedLibrarySpec	56
was.SibDestination	56
was.SibQueue	58
was.SibQueueConnectionFactory	60
was.SibQueueConnectionFactorySpec	62
was.SibQueueDestination	62
was.SibQueueDestinationSpec	65
was.SibQueueSpec	65
was.SibTopic	66
was.SibTopicConnectionFactory	68
was.SibTopicConnectionFactorySpec	70
was.SibTopicSpaceDestination	70
was.SibTopicSpaceDestinationSpec	73
was.SibTopicSpec	73
was.UnmanagedServer	74
was.V5DefaultQueue	75
was.V5DefaultQueueConnectionFactory	77
was.V5DefaultQueueConnectionFactorySpec	79
was.V5DefaultQueueSpec	79
was.V5DefaultTopic	80
was.V5DefaultTopicConnectionFactory	82
was.V5DefaultTopicConnectionFactorySpec	84
was.V5DefaultTopicSpec	84
was.VirtualHost	85
was.VirtualHostSpec	87
was.War	87
was.WarModule	88
was.WasAppContainer	91
was.WasContainer	91
was.WmqQueue	91
was.WmqQueueConnectionFactory	93
was.WmqQueueConnectionFactorySpec	95
was.WmqQueueSpec	95
was.WmqTopic	96
was.WmqTopicConnectionFactory	98
was.WmqTopicConnectionFactorySpec	100
was.WmqTopicSpec	100

Preface

This document describes the functionality provided by the IBM WebSphere Application Server (WAS) plugin.

Refer to **Deployit Reference Manual** for background information on Deployit and deployment concepts.

Overview

The WAS plugin is a Deployit plugin that adds capability for managing deployments and resources on an existing WebSphere application server. It offers out of the box support for deploying/undeploying application artifacts, datasources, JMS resources and other Java EE resources and WAS configurations. See the **Features** section below for details. It can easily be extended to support more deployment options or management of new artifacts/resources on WAS.

Features

- Deploys and undeploys Java EE application artifacts:
 - Enterprise applications (EAR)
 - Web applications (WAR)
 - Enterprise JavaBeans (EJB JAR)
 - Creates and removes Java EE resources:
 - Oracle datasources
 - DB2 datasources (type 2 and type 4)
 - V5 default JMS resources: queues, topics and connection factories.
 - WebSphere MQ JMS resources: queues, topics and connection factories.
 - SIB JMS resources: queues, topics, connection factories and destinations.
- Creates and removes WAS configuration elements:
 - Shared libraries
 - Virtual hosts
- Discovers WAS topologies: cells, nodes, clusters, server, web servers.

Requirements

- **Deployit requirements**
 - **Deployit:** version 3.6+
 - **IBM WebSphere Application Server Standard and Network Deployment:** 6.1, 7.0 and 8.0 (Unix and Windows)
 - **Other Deployit Plugins:** None

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 WebSphere specific deployment package. It contain declarations for an [Ear](#), an [Oracle datasource spec](#) and a couple of JMS resources.

```
Manifest-Version: 1.0
Deployit-Package-Format-Version: 1.3
CI-Application: PetClinic-ear
CI-Version: 1.0-was

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

Name: sampleOracleDataSource	
CI-Type: was.OracleDataSourceSpec	
CI-jndiName: jdbc/sampleOracleDataSource	
CI-jdbcProvider: Oracle JDBC Driver	
CI-datasourceHelperClassname: com.ibm.websphere.rsadapter.Oracle10gDataStoreHelper	
CI-username: {{DATABASE_USERNAME}}	
CI-password: {{DATABASE_PASSWORD}}	
Name: sampleSibQueueDestination	
CI-Type: was.SibQueueDestinationSpec	
CI-busName: sampleSIBus	
Name: sampleSibQueue	
CI-Type: was.SibQueueSpec	
CI-queueName: sampleSibQueueDestination	
CI-busName: sampleSIBus	
CI-description: sample sib queue	

Using the deployables and deployed

The following table shows the possible containers a deployable can be targeted to and the deployed that will be created as a result of that.

Deployable vs. containers table

Deployable	Containers	Deployed
jee.Ear or was.Ear jee.War or was.War jee.EjbJar or was.EjbJar	was.UnmanagedServer was.ManagedServer was.Cluster	was.EarModule was.WarModule was.EjbJarModule
was.SharedLibrarySpec	was.UnmanagedServer was.ManagedServer was.Cluster was.NodeAgent was.DeploymentManager	was.SharedLibrary
was.VirtualHostSpec	was.UnmanagedServer was.DeploymentManager	was.VirtualHost
was.V5DefaultQueueSpec was.V5DefaultQueueConnectionFactorySpec was.V5DefaultTopicSpec was.V5DefaultTopicConnectionFactorySpec	was.UnmanagedServer was.ManagedServer was.Cluster was.NodeAgent was.DeploymentManager	was.V5DefaultQueue was.V5DefaultTopic was.V5DefaultQueueConnectionFactory was.V5DefaultTopicConnectionFactory
was.WmqQueueSpec was.WmqQueueConnectionFactorySpec was.WmqTopicSpec was.WmqTopicConnectionFactorySpec	was.UnmanagedServer was.ManagedServer was.Cluster was.NodeAgent was.DeploymentManager	was.WmqQueue was.WmqQueueConnectionFactory was.WmqTopic was.WmqTopicConnectionFactory
was.SibQueueSpec was.SibQueueConnectionFactorySpec was.SibQueueDestinationSpec was.SibTopicSpec was.SibTopicConnectionFactorySpec was.SibTopicSpaceDestinationSpec	was.UnmanagedServer was.ManagedServer was.Cluster was.NodeAgent was.DeploymentManager	was.SibQueue was.SibQueueConnectionFactory was.SibQueueDestination was.SibTopic was.SibTopicConnectionFactory was.SibTopicSpaceDestination

Deployed actions table

The following table shows the actions taken when creating, modifying or destroying a deployed.

Deployed	Create	Modify	Destroy
was.EarModule was.WarModule was.EjbJarModule	<ul style="list-style-type: none"> • deploy application • WAS ND only: synchronize applicable nodes • start application 	<ul style="list-style-type: none"> • stop application • undeploy application • WAS ND only: synchronize applicable nodes • deploy application • WAS ND only: synchronize applicable nodes • start application 	<ul style="list-style-type: none"> • stop application • undeploy application • WAS ND only: synchronize applicable nodes
was.SharedLibrary	<ul style="list-style-type: none"> • create shared library • WAS ND only: synchronize applicable nodes 	<ul style="list-style-type: none"> • destroy shared library • WAS ND only: synchronize applicable nodes • create shared library • WAS ND only: synchronize applicable nodes 	<ul style="list-style-type: none"> • destroy shared library • WAS ND only: synchronize applicable nodes
was.VirtualHost	<ul style="list-style-type: none"> • create virtual host • WAS ND only: synchronize applicable nodes 	<ul style="list-style-type: none"> • destroy virtual host • WAS ND only: synchronize applicable nodes • create virtual host • WAS ND only: synchronize applicable nodes 	<ul style="list-style-type: none"> • destroy virtual host • WAS ND only: synchronize applicable nodes
was.V5DefaultQueue was.V5DefaultTopic was.V5DefaultQueueConnectionFactory was.V5DefaultTopicConnectionFactory	<ul style="list-style-type: none"> • create V5 JMS resource • WAS ND only: synchronize applicable nodes 	<ul style="list-style-type: none"> • destroy V5 JMS resource • WAS ND only: synchronize applicable nodes • create V5 JMS resource • WAS ND only: synchronize applicable nodes 	<ul style="list-style-type: none"> • destroy V5 JMS resource • WAS ND only: synchronize applicable nodes
was.WmqQueue was.WmqQueueConnectionFactory was.WmqTopic was.WmqTopicConnectionFactory	<ul style="list-style-type: none"> • create WebSphere MQ JMS resource • WAS ND only: synchronize applicable nodes 	<ul style="list-style-type: none"> • destroy WebSphere MQ JMS resource • WAS ND only: synchronize applicable nodes • create WebSphere MQ JMS resource • WAS ND only: synchronize applicable nodes 	<ul style="list-style-type: none"> • destroy WebSphere MQ JMS resource • WAS ND only: synchronize applicable nodes
was.SibQueue was.SibQueueConnectionFactory was.SibQueueDestination was.SibTopic was.SibTopicConnectionFactory was.SibTopicSpaceDestination	<ul style="list-style-type: none"> • create WebSphere SIB JMS resource • WAS ND only: synchronize applicable nodes 	<ul style="list-style-type: none"> • destroy WebSphere MQ JMS resource • WAS ND only: synchronize applicable nodes • create WebSphere SIB JMS resource • WAS ND only: synchronize applicable nodes 	<ul style="list-style-type: none"> • destroy WebSphere SIB JMS resource • WAS ND only: synchronize applicable nodes

Discovering WAS middleware

Deployit can scan your environment for as far as possible and create Configuration Items in its repository based on the configurations it encounters during the scan. This process is known as *discovery*.

The CIs discovered during discovery will help you in setting up your infrastructure in an easy way. However, they need not be complete: some CIs contain properties that can not be automatically discovered, like passwords. These specific kind of properties will still need to be entered manually.

Because discovery is part of the Deployit plugin suite, the exact discovery functionality available varies depending on the middleware platforms present in your environment.

The following steps comprehend discovery:

1. Create a CI representing the starting point for discovery (often a *Host* CI).
2. Start discovery passing this CI.
3. Store the discovered CIs in the repository.
4. Complete the discovered CIs manually by providing missing needed properties.
5. Add the discovered CIs to an environment.

Note however that the last step of discovery is optional. The discovered CIs will be stored under the *Infrastructure* root node in the repository and may be added to an environment at some later time.

Create a Configuration Item starting point

The first step taken in discovery is to create a starting point to kick off the process from. This starting point consists of a Configuration Item specifying at least the host that discovery should start at. Depending on the middleware you are trying to discover, additional parameters may be needed.

Following is an example of how to start discovery for a WebSphere Application Server (WAS). First a CI is created for the host itself and next a Configuration Item is created for the deployment manager running on that host. The deployment manager CI will be the starting point for discovery.

```
# Create a CI with the required discovery parameters filled in
wasHost = factory.configurationItem('Infrastructure/rs94asob.k94.corp.com', 'overthere.SshHost', {
  'address': 'was-61',
  'username': 'root',
  'password': 'rootpwd',
  'os': 'UNIX',
  'accessMethod': 'SSH_SFTP'
})

repository.create(wasHost)

# Now create a WAS deployment manager
dmManager = factory.configurationItem(wasHost.id + '/wasDM', 'was.DeploymentManager', {
  'host': 'Infrastructure/rs94asob.k94.corp.com',
  'version': 'WAS_61',
  'wasHome': '/opt/ws/6.1/profiles/dmgr',
  'username': 'wsadmin',
  'password': 'wsadmin'
})
```

Start discovery passing a Configuration Item

After the CI starting point has been created, it can be used to perform discovery. The Deployit CLI discovery functionality is synchronous, which means that the CLI will wait until the discovery process finishes.

The process of discovery works exactly like a regular task in that it executes a number of steps behind the scenes. Whenever one of these steps fails, the entire discovery fails and aborts. It is not possible to continue an interrupted discovery process.

The command to start discovery is:

```
discoveredCIs = deployit.discover(dmManager)
```

Note there are no single- or double quotes around `dmManager`, because it's an object and not a string. The result of this command will be an object containing a list of discovered CIs.

Store the CIs in the repository

Deployit returns a list of discovered middleware CIs. Note that these are not yet persisted. To store them in the repository, use the following code:

```
repository.create(discoveredCIs)
```

Complete discovered middleware CIs

The easiest way to find out which of the discovered CIs require additional information is by printing them. Any CIs that contain passwords (displayed as '*****') will need to be completed. To print the stored CIs, the following code can be used:

```
for ci in discoveredCIs.objects: deployit.print(repository.read(ci.id));
```

Note: the created CIs can also be edited in the GUI using the Repository Browser.

Adding CIs to Environments

Middleware that is used as a deployment target must be grouped together in an environment. Environments are CIs and like all CIs, they can be created from the CLI. The following command can be used for this:

```
env = factory.configurationItem('Environments/DiscoveredEnv', 'udm.Environment')
```

Add the discovered CIs to the environment:

```
env.values['members'] = [ci.id for ci in discoveredCIs.objects]
```

Note that not all of the discovered CIs should necessarily be stored in an environment. For example, in the case of WAS, some nested CIs may be discovered of which only the top-level one must be stored.

Store the new environment:

```
repository.create(env)
```

The newly created environment can now be used as a deployment target.

Note: the user needs specific permission to store CIs in the database. See the *Deployit System Administration Manual*.

Extending the WAS plugin

The WAS plugin is designed to be extended through Deployit's Plugin API type system and through the use of custom, user defined WAS Python scripts.

The WAS plugin associates **Create**, **Modify**, **Destroy** and **Inspect** operations received from Deployit with WAS Python scripts that need to be executed for the specific operation to be performed. The operation specific script is given a Python object representation of the `Deployed` that triggered the operation. The script is then executed using **wsadmin**.

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

Please refer to the *Customization Manual* for a detailed explanation of the type system and advanced methods of customization of plugins.

Extending using XML and Python scripts

The easiest way of extending the WAS plugin is by using XML and Python scripts. In this way no Java code needs to be written. Extending the behaviour of the plugin is done by simply defining the necessary deployables and deployments for the specific environment. When Deployit starts up, it reads a file called `synthetic.xml` from the server class-path, i.e. the `ext` directory of the server.

This file contains the type definitions of the deployables and the deployments, i.e. the information about the types and their properties. In addition to that, it defines which Python scripts should be executed for a particular operation, as mentioned above. The scripts have all the information from the `Deployed` at their disposal to perform their work.

As an example, this is the type definition of a virtual host as it appears in `synthetic.xml`:

```
<type type="was.VirtualHost" extends="was.Resource" deployable-type="was.VirtualHostSpec" container-type="was.Cell">
  <generate-deployable type="was.VirtualHostSpec" extends="was.Deployable" />
  <property name="createScript" default="was/virtualhost/create-virtual-host.py" hidden="true" />
  <property name="destroyScript" default="was/virtualhost/destroy-virtual-host.py" hidden="true" />
  <property name="inspectScript" default="was/virtualhost/inspect-virtual-host.py" hidden="true" />
  <property name="aliases" kind="set_of_string" description="Virtual host aliases - enter alias as: hostname:port" />
</type>
```

Looking at this type definition more closely, it indicates that a virtual host, the type of which is `was.VirtualHost` will be created on its target infrastructure, called container, with a `container-type` of `was.Cell`; On WAS a virtual host may be created on a cell and that is exactly what this definition means.

The attribute `extends` tells the plugin what resource is extended by this definition. In this case it's a simple basic resource so it extends the type `was.Resource`. Since multiple virtual hosts may be created this way, each with its own set of properties, a specification of what will be deployed is created, a so-called `deployable-type`, and is itself of type `was.VirtualHostSpec`.

Within the type definition, there is the possibility of specifying properties of exactly how the virtual host is to be created. The first property is called `createScript` and specifies the script to be executed by **wsadmin** for the creation of the virtual host. An extension of this plugin could specify a different creation script. The plugin comes with a default creation script (`createScript`):

create-virtual-host.py

```
import re

pattern = re.compile('^[^:]+:\d{,5}')
virtualHostParent = AdminConfig.getid('/Cell:%s/' % (deployed.container.cellName))
attributes = [['name', deployed.name]]
attributes.append(['aliases', [[['hostname', alias.split(':')[0]], ['port', alias.split(':')[1]]]
                             for alias in deployed.aliases if pattern.match(alias) != None]])

re.purge()

print "Creating virtual host %s on target scope %s with attribute(s) %s" % (deployed.name, virtualHostParent, attributes)
AdminConfig.create('VirtualHost', virtualHostParent, attributes)
```

This script also shows that `aliases` are also created for the specified virtual host. Aliases may be specified using the property `aliases` and it takes a set of strings. An example of this property is:

```
<property name='aliases' kind='set_of_string' value='www.my-domain.com:80,www.proxy-domain.com:8443' />
```

In addition to a creation script, a destruction script (`destroyScript`) must also be specified:

destroy-virtual-host.py

```
virtualHostContainmentPath = '/Cell:%s/VirtualHost:%s' % (deployed.container.cellName, deployed.name)
virtualHostId = validateNotEmpty(AdminConfig.getid(virtualHostContainmentPath),
"Cannot find virtual host with id: %s" % (virtualHostContainmentPath))

print "Destroying virtual host %s" % (deployed.name)
AdminConfig.remove(virtualHostId)
```

Finally, a modification script (`modifyScript`) may also be specified. If that is not present, the destruction script is invoked to remove the resource with the old settings and then the creation script is invoked to create the resource with the new settings.

Adding a property

The architecture of the WAS plugin enables the transfer of properties from the deployed - `was.VirtualHost` in this example - to the accompanying Python scripts defined with the properties `createScript` and `destroyScript`. In order for this to be possible, the properties are bound to an object called `deployed` and can be accessed as `deployed.<property-name>`. This can be seen in the creation script, where the property `aliases` is available as `deployed.aliases` and the name of the virtual host as `deployed.name`. Using this convention as many properties as needed by the scripts can be bound to the type definition. Care should be taken that the scripts use the same type as specified in the definition, so if a property defines a `kind=integer`, the script should also treat the value of this property as being of type `integer`.

An example of adding another property is:

```
<property name='index-range' kind='integer' value='999' description='maximum index of an array of 1000 items' />
```

Fixing a property

In the case where it is necessary that a specific property should always contain a fixed predefined value, the attribute `hidden=true` should be used on the definition of the property. This way an end-user, who is performing the deployment using the Deployit user interface, won't be able to see this property and therefore not have the possibility of modifying it. Usually this is done for the Python scripts which once written are not allowed to be changed by an end-user.

Using the example of the previous section and wanting to fix the maximum range, the example becomes:

```
<property name='index-range' kind='integer' value='999' hidden='true' description='maximum index of an array of 1000 items' />
```

Inspection

One thing left to be mentioned is the property called `inspectScript`. When the Domain manager of WAS is known, it is possible for Deployit to discover on its own most of the WAS topology automatically. This specific script is called upon by Deployit when it's trying, in our example, to discover a virtual host on a WAS cell. Following is the default script implementation shipped with the WAS plugin:

inspect-virtual-host.py

```
print "Inspecting virtual host %s" % (deployed.name)
```

```

virtualHostContainmentPath = '/Cell:%s/VirtualHost:%s' % (deployed.container.cellName, deployed.name)
virtualHostId = validateNotEmpty(AdminConfig.getid(virtualHostContainmentPath),
"Cannot find virtual host %s" % (virtualHostContainmentPath))

inspected('name', AdminConfig.showAttribute(virtualHostId, 'name'))
aliases = AdminConfig.getid(virtualHostContainmentPath + '/HostAlias:').split()
inspected('aliases', [AdminConfig.showAttribute(alias, 'hostname') + ':' +
AdminConfig.showAttribute(alias, 'port') for alias in aliases if alias != ""])

```

In the above script, discovered information is handed back to the plugin by use of the `inspected` method. This method takes two parameters, the first one being the name of the discovered item or property and the second one the actual value. The value should be representable as either a string, or a list of strings. The declaration of this method can be found in the Plugin API in the file `base.py`.

More information about the process of discovery and how to use it can be found in the *Command Line Interface (CLI) manual*.

Execution order

A deployment process consists of a series of steps that are executed sequentially. Plugins offer the ability to influence the order of execution of the steps contributed to the deployment process in relation to other contributed steps and operations, not necessarily contributed by the same plugin(s), that are part of the deployment process.

The order of execution allows for the chaining of scripts or operations to create a logical sequence of events. In order to specify the order, properties with the name of `createOrder` and `destroyOrder` may be used with an attribute called `default` to specify the order ordinal. For example, the following `synthetic.xml` snippet says that creation of the virtual host, `default=60`, will happen before any step with a higher order, for example `default=70`, but after any step with a lower order, i.e. lower than `default=60`.

```

<type type="was.VirtualHost" extends="was.Resource" deployable-type="was.VirtualHostSpec" container-type="was.Cell">
  <generate-deployable type="was.VirtualHostSpec" extends="was.Deployable" />

  <property name="createScript" default="was/virtualhost/create-virtual-host.py" hidden="true" />
  <property name="createVerb" default="Deploy" hidden="true" />
  <property name="createOrder" kind="integer" default="60" hidden="true" />

  <property name="destroyScript" default="was/virtualhost/destroy-virtual-host.py" hidden="true" />
  <property name="destroyVerb" default="Undeploy" hidden="true" />
  <property name="destroyOrder" kind="integer" default="30" hidden="true" />

  <property name="inspectScript" default="was/virtualhost/inspect-virtual-host.py" hidden="true" />
  <property name="aliases" kind="set_of_string" description="Virtual host aliases - enter alias as: hostname:port" />
</type>

```

Notice that the `destroyOrder` has a low order, because when executing a deployment the virtual host should be destroyed before it can be created again.

CI Reference

Configuration Item Overview

Deployable Configuration Items

CI	Description
was.DB2Type2DatasourceSpec	Specification for a DB2 type 2 data source
was.DB2Type4DatasourceSpec	Specification for a DB2 type 4 data source
was.Ear	Java EE EAR archive
was.EjbJar	Java EE EJB archive
was.OracleDatasourceSpec	Specification for an Oracle datasource
was.SharedLibrarySpec	Specification of a shared library
was.SibQueueConnectionFactorySpec	Specification of a SIB queue connection factory
was.SibQueueDestinationSpec	Specification of a SIB queue destination
was.SibQueueSpec	Specification of a SIB queue
was.SibTopicConnectionFactorySpec	Specification of a SIB topic connection factory
was.SibTopicSpaceDestinationSpec	Specification of a SIB topic space destination
was.SibTopicSpec	Specification of a SIB topic
was.V5DefaultQueueConnectionFactorySpec	Specification of a V5 default queue connection factory
was.V5DefaultQueueSpec	Specification of a V5 default queue
was.V5DefaultTopicConnectionFactorySpec	Specification of a V5 default topic connection factory
was.V5DefaultTopicSpec	Specification of a V5 default topic
was.VirtualHostSpec	Specification of a virtual host
was.War	Java EE WAR archive
was.WmqQueueConnectionFactorySpec	Specification of a WMQ queue connection factory
was.WmqQueueSpec	Specification of a WMQ queue
was.WmqTopicConnectionFactorySpec	Specification of a WMQ topic connection factory
was.WmqTopicSpec	Specification of a WMQ topic

Deployed Configuration Items

CI	Description
was.DB2Type2Datasource	DB2 type 2 data source object supplies your application with connections for accessing the database
was.DB2Type4Datasource	DB2 type 4 data source object supplies your application with connections for accessing the database
was.EarModule	EAR with values configured for a deployment
was.EjbModule	EJB with values configured for a deployment
was.OracleDatasource	Oracle data source object supplies your application with connections for accessing the database
was.SharedLibrary	Container-wide shared library that can be used by deployed applications
was.SibQueue	JMS queue is used as a destination for point-to-point messaging
was.SibQueueConnectionFactory	JMS queue connection factory is used to create connections to the associated JMS provider of JMS queues, for point-to-point messaging
was.SibQueueDestination	Queue for point-to-point messaging
was.SibTopic	JMS topic is used as a destination for publish/subscribe messaging
was.SibTopicConnectionFactory	JMS topic connection factory is used to create connections to the associated JMS provider of JMS topics, for publish/subscribe messaging
was.SibTopicSpaceDestination	Topic space is a location for publish/subscribe messaging
was.V5DefaultQueue	Queue destinations provided for point-to-point messaging by the internal WebSphere JMS provider
was.V5DefaultQueueConnectionFactory	Specifies a topic connection factory, which is used to create connections to the associated JMS provider of JMS queue destinations for point-to-point messaging
was.V5DefaultTopic	Specifies the topic destinations for publish and subscribe messaging by the internal WebSphere JMS provider
was.V5DefaultTopicConnectionFactory	Specifies a topic connection factory, which is used to create connections to the associated JMS provider of JMS topic destinations for publish and subscribe messaging
was.VirtualHost	Virtual host with a unique set of Web access ports
was.WarModule	WAR with values configured for a deployment
was.WmqQueue	Queue destinations provided for point-to-point messaging by the WebSphere MQ JMS provider
was.WmqQueueConnectionFactory	Queue connection factory is used to create connections to the associated JMS provider of JMS queue destinations, for point-to-point messaging
was.WmqTopic	Topic destinations provided for publish and subscribe messaging by the WebSphere MQ JMS provider
was.WmqTopicConnectionFactory	Topic connection factory is used to create connections to the associated JMS provider of JMS topic destinations, for publish and subscribe messaging

Topology Configuration Items

CI	Description
was.Cluster	A WebSphere cluster managed by a deployment manager (WAS ND)
was.DeploymentManager	A WebSphere Application Server deployment manager (WAS ND)
was.ManagedServer	A WebSphere server managed by a node that is part of a deployment manager (WAS ND)
was.ManagedWebServer	An HTTP server managed by a node that is part of a deployment manager (WAS ND)
was.NodeAgent	A WebSphere node agent
was.UnmanagedServer	An unmanaged WebSphere Application Server (WAS Base/SA)

Virtual Deployable Configuration Items

CI	Description
was.Deployable	Base class for all deployable configuration items

Virtual Deployed Configuration Items

CI	Description
was.DB2Datasource	Base class for all deployed DB2 data source configuration items
was.Datasource	Base class for all deployed data source configuration items
was.ExtensibleDeployed	Base class for all extensible deployed configuration items
was.ExtensibleDeployedArtifact	Base class for all extensible deployed artifact configuration items
was.ExtensibleDeployedResource	Base class for all extensible deployed resource configuration items
was.JmsResource	Base class for all deployed JMS resource configuration items
was.Module	Base class for all deployed JEE module configuration items
was.Resource	Base class for all deployed JEE resource configuration items
was.SibDestination	Base class for all deployed SIB destination configuration items

Virtual Topology Configuration Items

CI	Description
was.BaseCell	Base class for a Webshpere Cell
was.Cell	
was.Node	
was.Server	
was.WasAppContainer	
was.WasContainer	

Configuration Item Details

[was.BaseCell](#)

Hierarchy `udm.BaseContainer >> udm.BaseConfigurationItem`

Interfaces `udm.Taggable`, `python.PythonManagingContainer`, `python.PythonManagedContainer`, [was.WasContainer](#), `udm.ConfigurationItem`, [was.Cell](#), `udm.Container`

Base class for a Webshpere Cell

Public Properties



host : `CI<overthere.Host>`

Host on which the unmanaged WAS server runs

version : `ENUM [WAS_61, WAS_70, WAS_80]`

Version of WebSphere Application Server.



wasHome : `STRING`

Root path of the WebSphere installation path. e.g. `/opt/ws/6.1/appserver/profiles/AppSrv01`



password : `STRING`

Password which is used to login to the WebSphere Administration.



port : `INTEGER`

TCP port which is used to login to the WebSphere Administration, default is 8880

tags : `SET_OF_STRING`

The tags to map deployables to containers.



username : `STRING`

Username which is used to login to the WebSphere Administration.

Hidden Properties

runWithDaemon : `BOOLEAN = true`

Set to true to execute commands with the Python daemon

was.Cell

null

was.Cluster

Hierarchy `udm.BaseContainer >> udm.BaseConfigurationItem`

Interfaces `udm.Tagable`, [was.WasAppContainer](#), `python.PythonManagedContainer`, [was.WasContainer](#), `udm.ConfigurationItem`, `udm.Container`

A WebSphere cluster managed by a deployment manager (WAS ND)

Public Properties



cell : `CI<was.DeploymentManager>`

Deployment manager that manages this this cluster

servers : `SET_OF_CI<was.ManagedServer>`

Servers that are part of this cluster

tags : `SET_OF_STRING`

The tags to map deployables to containers.

was.DB2Datasource

Hierarchy [was.Datasource](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Base class for all deployed DB2 data source configuration items.

Public Properties



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

The container on which this deployed runs.

databaseName : [STRING](#)

This is an actual database name, and its not the locally catalogued database name. The Universal JDBC Driver does not rely on information catalogued in the DB2 database directory.

datasourceHelperClassname : [STRING](#)

Specifies the data source helper class

jdbcProvider : [STRING](#)

Name of the JDBC Provider for this data source

jndiName : [STRING](#)

JNDI name of the data source

password : [STRING](#)

Password to use when connecting to the data source

username : [STRING](#)

Username to use when connecting to the data source

ConnectionPool_connectionTimeout : [INTEGER](#)

Connection timeout value for a JDBC data source.

ConnectionPool_maxConnections : [INTEGER](#)

Maximum number of connections for a JDBC data source.

ConnectionPool_minConnections : [INTEGER](#)

Minimum number of connections for a JDBC data source.

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

The deployable that this deployed is derived from.

description : [STRING](#)

Description of this data source

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jndiName, jdbcProvider, datasourceHelperClassname, username, password, ConnectionPool*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/datasource/create-datasource.py*

Python wsadmin script invoked to create a JDBC data source.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/datasource/destroy-datasource.py*

Python wsadmin script invoked to destroy a JDBC data source.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/datasource/inspect-datasource.py*

Python wsadmin script invoked to inspect a JDBC data source.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : *INTEGER = 65*

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : **INTEGER** = *45*

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = *65*

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.DB2Type2Datasource

Hierarchy [was.DB2Datasource](#) >> [was.Datasource](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

DB2 type 2 data source object supplies your application with connections for accessing the database.

Public Properties



container : `CI<udm.Container>`

The container on which this deployed runs.

databaseName : `STRING`

This is an actual database name, and its not the locally catalogued database name. The Universal JDBC Driver does not rely on information catalogued in the DB2 database directory.

datasourceHelperClassname : `STRING`

Specifies the data source helper class

jdbcProvider : `STRING`

Name of the JDBC Provider for this data source

jndiName : `STRING`

JNDI name of the data source

password : `STRING`

Password to use when connecting to the data source

username : `STRING`

Username to use when connecting to the data source

ConnectionPool_connectionTimeout : `INTEGER`

Connection timeout value for a JDBC data source.

ConnectionPool_maxConnections : `INTEGER`

Maximum number of connections for a JDBC data source.

ConnectionPool_minConnections : `INTEGER`

Minimum number of connections for a JDBC data source.

deployable : `CI<udm.Deployable>`

The deployable that this deployed is derived from.

description : `STRING`

Description of this data source

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jndiName, jdbcProvider, datasourceHelperClassname, username, password, ConnectionPool*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/datasource/create-datasource.py*

Python wsadmin script invoked to create a JDBC data source.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/datasource/destroy-datasource.py*

Python wsadmin script invoked to destroy a JDBC data source.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

driverType : *STRING = 2*

JDBC connectivity-type of a data source. If you want to use type 4 driver, set the value to 4. If you want to use type 2 driver, set the value to 2. On WebSphere Application Server for Z/OS, driverType 2 uses RRS and supports 2-phase commit processing.

inspectScript : *STRING = was/datasource/inspect-datasource.py*

Python wsadmin script invoked to inspect a JDBC data source.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : **INTEGER** = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : **INTEGER** = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.DB2Type2DatasourceSpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification for a DB2 type 2 data source. On WebSphere Application Server for Z/OS, driverType 2 uses RRS and supports 2-phase commit processing.

Public Properties**ConnectionPool_connectionTimeout** : **STRING**

Connection timeout value for a JDBC data source. (integer)

ConnectionPool_maxConnections : **STRING**

Maximum number of connections for a JDBC data source. (integer)

ConnectionPool_minConnections : **STRING**

Minimum number of connections for a JDBC data source. (integer)

databaseName : **STRING**

This is an actual database name, and its not the locally catalogued database name. The Universal JDBC Driver does not rely on information catalogued in the DB2 database directory. (string)

datasourceHelperClassname : **STRING**

Specifies the data source helper class (string)

description : **STRING**

Description of this data source (string)

jdbcProvider : **STRING**

Name of the JDBC Provider for this data source (string)

jndiName : **STRING**

JNDI name of the data source (string)

password : **STRING**

Password to use when connecting to the data source (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

username : **STRING**

Username to use when connecting to the data source (string)

was.DB2Type4Datasource

Hierarchy [was.DB2Datasource](#) >> [was.Datasource](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

DB2 type 4 data source object supplies your application with connections for accessing the database.

Public Properties

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

The container on which this deployed runs.

databaseName : STRING

This is an actual database name, and its not the locally catalogued database name. The Universal JDBC Driver does not rely on information catalogued in the DB2 database directory.

datasourceHelperClassname : STRING

Specifies the data source helper class

jdbcProvider : STRING

Name of the JDBC Provider for this data source

jndiName : STRING

JNDI name of the data source

password : STRING

Password to use when connecting to the data source

portNumber : INTEGER = 50000

The TCP/IP port number where the DRDA server resides. If property driverType is set to 4, this property is required.

serverName : STRING

The TCP/IP address or host name for the DRDA server. If property driverType is set to 4, this property is required.

username : STRING

Username to use when connecting to the data source

ConnectionPool_connectionTimeout : INTEGER

Connection timeout value for a JDBC data source.

ConnectionPool_maxConnections : INTEGER

Maximum number of connections for a JDBC data source.

ConnectionPool_minConnections : INTEGER

Minimum number of connections for a JDBC data source.

deployable : CI<udm.Deployable>

The deployable that this deployed is derived from.

description : STRING

Description of this data source

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jndiName, jdbcProvider, datasourceHelperClassname, username, password, ConnectionPool*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/datasource/create-datasource.py*

Python wsadmin script invoked to create a JDBC data source.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/datasource/destroy-datasource.py*

Python wsadmin script invoked to destroy a JDBC data source.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

driverType : *STRING = 4*

JDBC connectivity-type of a data source. If you want to use type 4 driver, set the value to 4. If you want to use type 2 driver, set the value to 2. On WebSphere Application Server for Z/OS, driverType 2 uses RRS and supports 2-phase commit processing.

inspectScript : *STRING = was/datasource/inspect-datasource.py*

Python wsadmin script invoked to inspect a JDBC data source.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : **INTEGER** = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : **INTEGER** = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.DB2Type4DatasourceSpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification for a DB2 type 4 data source.

Public Properties**ConnectionPool_connectionTimeout** : **STRING**

Connection timeout value for a JDBC data source. (integer)

ConnectionPool_maxConnections : **STRING**

Maximum number of connections for a JDBC data source. (integer)

ConnectionPool_minConnections : **STRING**

Minimum number of connections for a JDBC data source. (integer)

databaseName : **STRING**

This is an actual database name, and its not the locally catalogued database name. The Universal JDBC Driver does not rely on information catalogued in the DB2 database directory. (string)

datasourceHelperClassname : **STRING**

Specifies the data source helper class (string)

description : **STRING**

Description of this data source (string)

jdbcProvider : **STRING**

Name of the JDBC Provider for this data source (string)

jndiName : **STRING**

JNDI name of the data source (string)

password : **STRING**

Password to use when connecting to the data source (string)

portNumber : **STRING**

The TCP/IP port number where the DRDA server resides. If property driverType is set to 4, this property is required. (integer)

serverName : **STRING**

The TCP/IP address or host name for the DRDA server. If property driverType is set to 4, this property is required. (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

username : **STRING**

Username to use when connecting to the data source (string)

was.Datasource

Hierarchy [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >>
[python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Base class for all deployed data source configuration items.

Public Properties

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

The container on which this deployed runs.

datasourceHelperClassName : `STRING`

Specifies the data source helper class

jdbcProvider : `STRING`

Name of the JDBC Provider for this data source

jndiName : `STRING`

JNDI name of the data source

password : `STRING`

Password to use when connecting to the data source

username : `STRING`

Username to use when connecting to the data source

ConnectionPool_connectionTimeout : `INTEGER`

Connection timeout value for a JDBC data source.

ConnectionPool_maxConnections : `INTEGER`

Maximum number of connections for a JDBC data source.

ConnectionPool_minConnections : `INTEGER`

Minimum number of connections for a JDBC data source.

deployable : `CI<udm.Deployable>`

The deployable that this deployed is derived from.

description : `STRING`

Description of this data source

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jndiName, jdbcProvider, datasourceHelperClassname, username, password, ConnectionPool*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/datasource/create-datasource.py*

Python wsadmin script invoked to create a JDBC data source.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/datasource/destroy-datasource.py*

Python wsadmin script invoked to destroy a JDBC data source.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/datasource/inspect-datasource.py*

Python wsadmin script invoked to inspect a JDBC data source.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : *INTEGER = 65*

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : INTEGER = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : INTEGER = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : BOOLEAN = false

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : STRING

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : STRING

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : STRING

Python script invoked to stop a Java EE artifact or Java EE resource

was.Deployable

Hierarchy udm.BaseDeployable >> udm.BaseConfigurationItem

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

Base class for all deployable configuration items.

Public Properties

tags : SET_OF_STRING

The tags to map deployables to containers.

was.DeploymentManager

Hierarchy was.BaseCell >> udm.BaseContainer >> udm.BaseConfigurationItem

Interfaces udm.Tagable, python.PythonManagingContainer, python.PythonManagedContainer, was.WasContainer, udm.ConfigurationItem, was.Cell, udm.Container

A WebSphere Application Server deployment manager (WAS ND)

Public Properties



host : `CI<overthere.Host>`

Host on which the unmanaged WAS server runs



nodeAgents : `SET_OF_CI<was.NodeAgent>`

WebSphere nodes belonging to cell

version : `ENUM [WAS_61, WAS_70, WAS_80]`

Version of WebSphere Application Server.



wasHome : `STRING`

Root path of the WebSphere installation path. e.g. `/opt/ws/6.1/appserver/profiles/AppSrv01`



password : `STRING`

Password which is used to login to the WebSphere Administration.



port : `INTEGER`

TCP port which is used to login to the WebSphere Administration, default is 8880

tags : `SET_OF_STRING`

The tags to map deployables to containers.



username : `STRING`

Username which is used to login to the WebSphere Administration.

Hidden Properties

runWithDaemon : `BOOLEAN = true`

Set to true to execute commands with the Python daemon

was.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`

Java EE 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

preCompileJsps : **STRING**

Specify whether to precompile JavaServer Pages (JSP) files as a part of installation. The default is not to precompile JSP files. (boolean)

roleMappings : **MAP_STRING_STRING**

Security role to user/group mapping (map_string_string)

scanPlaceholders : **BOOLEAN** = *true*

Scan Placeholders

startingWeight : **STRING**

Specifies the order in which modules are started when the server starts. The module with the lowest starting weight is started first. (integer)

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

was.EarModule

Hierarchy [was.Module](#) >> [was.ExtensibleDeployedArtifact](#) >> [was.ExtensibleDeployed](#) >>

[python.PythonManagedDeployed](#) >> [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 or

preCompileJsps : `BOOLEAN = false`

Specify whether to precompile JavaServer Pages (JSP) files as a part of installation. The default is not to precompile JSP files.

roleMappings : `MAP_STRING_STRING`

Security role to user/group mapping

sharedLibraries : `SET_OF_CI<was.SharedLibrary>`

Shared libraries used by this application

startingWeight : `INTEGER = -1`

Specifies the order in which modules are started when the server starts. The module with the lowest starting weight is started first.

virtualHost : `CI<was.VirtualHost>`

Virtual host this application should be mapped to

webServers : `SET_OF_CI<was.ManagedWebServer>`

Web servers this application should be mapped to

Hidden Properties

createOrder : INTEGER = 70

The order in which a create step will be executed.

createScript : STRING = *was/application/deploy-application.py*

Python wsadmin script invoked to create a Java EE artifact.

createVerb : STRING = *Deploy*

The word is used to prefix a step description for the create operation.

destroyOrder : INTEGER = 30

The order in which a destroy step will be executed.

destroyScript : STRING = *was/application/undeploy-application.py*

Python wsadmin script invoked to destroy a Java EE artifact.

destroyVerb : STRING = *Undeploy*

The word is used to prefix a step description for the destroy operation.

inspectVerb : STRING = *Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : INTEGER = 70

The order in which a modify step will be executed.

modifyVerb : STRING = *Upgrade*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : STRING = *id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : INTEGER = 90

The order in which a start step will be executed.

startScript : STRING = *was/application/start-application.py*

Python wsadmin script invoked to start running a Java EE artifact

startVerb : STRING = *Start*

The word is used to prefix a step description for the start operation.

stopOrder : INTEGER = 10

The order in which a stop step will be executed.

stopScript : STRING = *was/application/stop-application.py*

Python wsadmin script invoked to stop running a Java EE artifact.

stopVerb : STRING = *Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : INTEGER = 75

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : INTEGER = 35

Sync After Destroy Order

syncAfterModifyOrder : **INTEGER** = *75*

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

inspectScript : **STRING**

Python script invoked to inspect a Java EE artifact or Java EE resource

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

was.EjbJar

Hierarchy jee.EjbJar >> 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

Java EE EJB 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

roleMappings : **MAP_STRING_STRING**

Security role to user/group mapping (map_string_string)

scanPlaceholders : **BOOLEAN** = *true*

Scan Placeholders

startingWeight : **STRING**

Specifies the order in which modules are started when the server starts. The module with the lowest starting weight is started first. (integer)

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

was.EjbModule

Hierarchy was.Module >> was.ExtensibleDeployedArtifact >> was.ExtensibleDeployed >>

python.PythonManagedDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

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

EJB 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 or

roleMappings : MAP_STRING_STRING

Security role to user/group mapping

sharedLibraries : SET_OF_CI<was.SharedLibrary>

Shared libraries used by this application

startingWeight : INTEGER = -1

Specifies the order in which modules are started when the server starts. The module with the lowest starting weight is started first.

Hidden Properties

createOrder : INTEGER = 70

The order in which a create step will be executed.

createScript : STRING = *was/application/deploy-application.py*

Python wsadmin script invoked to create a Java EE artifact.

createVerb : STRING = *Deploy*

The word is used to prefix a step description for the create operation.

destroyOrder : INTEGER = 30

The order in which a destroy step will be executed.

destroyScript : STRING = *was/application/undeploy-application.py*

Python wsadmin script invoked to destroy a Java EE artifact.

destroyVerb : STRING = *Undeploy*

The word is used to prefix a step description for the destroy operation.

inspectVerb : STRING = *Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : INTEGER = 70

The order in which a modify step will be executed.

modifyVerb : STRING = *Upgrade*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : STRING = *id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : INTEGER = 90

The order in which a start step will be executed.

startScript : STRING = *was/application/start-application.py*

Python wsadmin script invoked to start running a Java EE artifact

startVerb : STRING = *Start*

The word is used to prefix a step description for the start operation.

stopOrder : INTEGER = 10

The order in which a stop step will be executed.

stopScript : STRING = *was/application/stop-application.py*

Python wsadmin script invoked to stop running a Java EE artifact.

stopVerb : STRING = *Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : INTEGER = 75

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : INTEGER = 35

Sync After Destroy Order

syncAfterModifyOrder : **INTEGER** = **75**

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

inspectScript : **STRING**

Python script invoked to inspect a Java EE artifact or Java EE resource

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

was.ExtensibleDeployed

Hierarchy python.PythonManagedDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

Interfaces udm.Deployed, udm.ConfigurationItem

Base class for all extensible deployed configuration items.

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 = 60

The order in which a create step will be executed.

createVerb : STRING = *Create*

The word is used to prefix a step description for the create operation.

destroyOrder : INTEGER = 40

The order in which a destroy step will be executed.

destroyVerb : STRING = *Destroy*

The word is used to prefix a step description for the destroy operation.

inspectVerb : STRING = *Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : INTEGER = 60

The order in which a modify step will be executed.

modifyVerb : STRING = *Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : STRING = *id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : INTEGER = 50

The order in which a start step will be executed.

startVerb : STRING = *Start*

The word is used to prefix a step description for the start operation.

stopOrder : INTEGER = 50

The order in which a synchronize after modify stop will be executed.

stopVerb : STRING = *Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : INTEGER = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : INTEGER = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : INTEGER = 65

The order in which a synchronize after modify step will be executed.

createScript : STRING

Python script invoked to deploy a Java EE artifact or create a Java EE resource

destroyScript : STRING

Python script invoked to undeploy a Java EE artifact or destroy a Java EE resource

exposeDeployedApplication : BOOLEAN = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

inspectScript : **STRING**

Python script invoked to inspect a Java EE artifact or Java EE resource

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.ExtensibleDeployedArtifact

Hierarchy [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Base class for all extensible deployed artifact configuration items.

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 or

Hidden Properties

createOrder : INTEGER = 60

The order in which a create step will be executed.

createVerb : STRING = *Create*

The word is used to prefix a step description for the create operation.

destroyOrder : INTEGER = 40

The order in which a destroy step will be executed.

destroyVerb : STRING = *Destroy*

The word is used to prefix a step description for the destroy operation.

inspectVerb : STRING = *Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : INTEGER = 60

The order in which a modify step will be executed.

modifyVerb : STRING = *Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : STRING = *id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : INTEGER = 50

The order in which a start step will be executed.

startVerb : STRING = *Start*

The word is used to prefix a step description for the start operation.

stopOrder : INTEGER = 50

The order in which a synchronize after modify stop will be executed.

stopVerb : STRING = *Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : INTEGER = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : INTEGER = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : INTEGER = 65

The order in which a synchronize after modify step will be executed.

createScript : STRING

Python script invoked to deploy a Java EE artifact or create a Java EE resource

destroyScript : STRING

Python script invoked to undeploy a Java EE artifact or destroy a Java EE resource

exposeDeployedApplication : BOOLEAN = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

inspectScript : **STRING**

Python script invoked to inspect a Java EE artifact or Java EE resource

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.ExtensibleDeployedResource

Hierarchy [was.ExtensibleDeployed](#) >> python.PythonManagedDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

Interfaces udm.Deployed, udm.ConfigurationItem

Base class for all extensible deployed resource configuration items.

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 = 60

The order in which a create step will be executed.

createVerb : STRING = *Create*

The word is used to prefix a step description for the create operation.

destroyOrder : INTEGER = 40

The order in which a destroy step will be executed.

destroyVerb : STRING = *Destroy*

The word is used to prefix a step description for the destroy operation.

inspectVerb : STRING = *Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : INTEGER = 60

The order in which a modify step will be executed.

modifyVerb : STRING = *Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : STRING = *id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : INTEGER = 50

The order in which a start step will be executed.

startVerb : STRING = *Start*

The word is used to prefix a step description for the start operation.

stopOrder : INTEGER = 50

The order in which a synchronize after modify stop will be executed.

stopVerb : STRING = *Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : INTEGER = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : INTEGER = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : INTEGER = 65

The order in which a synchronize after modify step will be executed.

createScript : STRING

Python script invoked to deploy a Java EE artifact or create a Java EE resource

destroyScript : STRING

Python script invoked to undeploy a Java EE artifact or destroy a Java EE resource

exposeDeployedApplication : BOOLEAN = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

inspectScript : **STRING**

Python script invoked to inspect a Java EE artifact or Java EE resource

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.JmsResource

Hierarchy [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >>
python.PythonManagedDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

Interfaces udm.Deployed, udm.ConfigurationItem

Base class for all deployed JMS resource configuration items.

Public Properties



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

The container on which this deployed runs.

jndiName : **STRING**

JNDI name for the resource

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

The deployable that this deployed is derived from.

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jmsProvider, jmsType*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/jms/create-jms-object.py*

Python wsadmin script invoked to create a JMS resource.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/jms/destroy-jms-object.py*

Python wsadmin script invoked to destroy a JMS resource.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/jms/inspect-jms-object.py*

Python wsadmin script invoked to inspect a JMS resource.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : *INTEGER = 65*

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : *INTEGER = 45*

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = *65*

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.ManagedServer

Hierarchy udm.BaseContainer >> udm.BaseConfigurationItem

Interfaces udm.Tagable, [was.WasAppContainer](#), [was.Server](#), python.PythonManagedContainer, [was.WasContainer](#), udm.ConfigurationItem, udm.Container

A WebSphere server managed by a node that is part of a deployment manager (WAS ND)

Public Properties

 **node** : **CI**<*was.NodeAgent*>

Node on which the server runs

tags : **SET_OF_STRING**

The tags to map deployables to containers.

Hidden Properties

restartServerScript : **STRING** = *was/server/restart-server.py*

Restart Server Script

was.ManagedWebServer

Hierarchy udm.BaseConfigurationItem

Interfaces udm.ConfigurationItem

An HTTP server managed by a node that is part of a deployment manager (WAS ND)

Public Properties

 **node** : **CI**<*was.NodeAgent*>

Node on which the http server runs

was.Module

Hierarchy [was.ExtensibleDeployedArtifact](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Base class for all deployed JEE module configuration items.

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 or

roleMappings : [MAP_STRING_STRING](#)

Security role to user/group mapping

sharedLibraries : [SET_OF_CI<was.SharedLibrary>](#)

Shared libraries used by this application

startingWeight : [INTEGER = -1](#)

Specifies the order in which modules are started when the server starts. The module with the lowest starting weight is started first.

Hidden Properties

createOrder : INTEGER = 70

The order in which a create step will be executed.

createScript : STRING = *was/application/deploy-application.py*

Python wsadmin script invoked to create a Java EE artifact.

createVerb : STRING = *Deploy*

The word is used to prefix a step description for the create operation.

destroyOrder : INTEGER = 30

The order in which a destroy step will be executed.

destroyScript : STRING = *was/application/undeploy-application.py*

Python wsadmin script invoked to destroy a Java EE artifact.

destroyVerb : STRING = *Undeploy*

The word is used to prefix a step description for the destroy operation.

inspectVerb : STRING = *Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : INTEGER = 70

The order in which a modify step will be executed.

modifyVerb : STRING = *Upgrade*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : STRING = *id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : INTEGER = 90

The order in which a start step will be executed.

startScript : STRING = *was/application/start-application.py*

Python wsadmin script invoked to start running a Java EE artifact

startVerb : STRING = *Start*

The word is used to prefix a step description for the start operation.

stopOrder : INTEGER = 10

The order in which a stop step will be executed.

stopScript : STRING = *was/application/stop-application.py*

Python wsadmin script invoked to stop running a Java EE artifact.

stopVerb : STRING = *Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : INTEGER = 75

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : INTEGER = 35

Sync After Destroy Order

syncAfterModifyOrder : **INTEGER** = 75

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

inspectScript : **STRING**

Python script invoked to inspect a Java EE artifact or Java EE resource

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

was.Node

null

was.NodeAgent

Hierarchy udm.BaseContainer >> udm.BaseConfigurationItem

Interfaces udm.Taggable, python.PythonManagedContainer, [was.WasContainer](#), udm.ConfigurationItem, [was.Node](#), udm.Container

A WebSphere node agent.

Public Properties

 **cell** : **CI**<was.DeploymentManager>

Deployment manager that manages this node agent

tags : **SET_OF_STRING**

The tags to map deployables to containers.

Hidden Properties

synchronizeScript : **STRING** = *was/base/synchronize-node.py*

Synchronize Script

was.OracleDatasource

Hierarchy [was.Datasource](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> python.PythonManagedDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

Interfaces udm.Deployed, udm.ConfigurationItem

Oracle data source object supplies your application with connections for accessing the database.

Public Properties

URL : **STRING**

JDBC URL



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

The container on which this deployed runs.

datasourceHelperClassname : **STRING**

Specifies the data source helper class

jdbcProvider : **STRING**

Name of the JDBC Provider for this data source

jndiName : **STRING**

JNDI name of the data source

password : **STRING**

Password to use when connecting to the data source

username : **STRING**

Username to use when connecting to the data source

ConnectionPool_connectionTimeout : **INTEGER**

Connection timeout value for a JDBC data source.

ConnectionPool_maxConnections : **INTEGER**

Maximum number of connections for a JDBC data source.

ConnectionPool_minConnections : **INTEGER**

Minimum number of connections for a JDBC data source.

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

The deployable that this deployed is derived from.

description : **STRING**

Description of this data source

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jndiName, jdbcProvider, datasourceHelperClassname, username, password, ConnectionPool*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/datasource/create-datasource.py*

Python wsadmin script invoked to create a JDBC data source.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/datasource/destroy-datasource.py*

Python wsadmin script invoked to destroy a JDBC data source.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/datasource/inspect-datasource.py*

Python wsadmin script invoked to inspect a JDBC data source.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : *INTEGER = 65*

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : **INTEGER** = *45*

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = *65*

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.OracleDataSourceSpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification for an Oracle datasource

Public Properties**ConnectionPool_connectionTimeout** : **STRING**

Connection timeout value for a JDBC data source. (integer)

ConnectionPool_maxConnections : **STRING**

Maximum number of connections for a JDBC data source. (integer)

ConnectionPool_minConnections : **STRING**

Minimum number of connections for a JDBC data source. (integer)

URL : **STRING**

JDBC URL (string)

datasourceHelperClassname : **STRING**

Specifies the data source helper class (string)

description : **STRING**

Description of this data source (string)

jdbcProvider : **STRING**

Name of the JDBC Provider for this data source (string)

jndiName : **STRING**

JNDI name of the data source (string)

password : **STRING**

Password to use when connecting to the data source (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

username : **STRING**

Username to use when connecting to the data source (string)

was.Resource

Hierarchy [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >>
[udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Base class for all deployed JEE resource configuration items.

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 = 60

The order in which a create step will be executed.

createVerb : STRING = *Create*

The word is used to prefix a step description for the create operation.

destroyOrder : INTEGER = 40

The order in which a destroy step will be executed.

destroyVerb : STRING = *Destroy*

The word is used to prefix a step description for the destroy operation.

inspectVerb : STRING = *Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : INTEGER = 60

The order in which a modify step will be executed.

modifyVerb : STRING = *Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : STRING = *id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : INTEGER = 50

The order in which a start step will be executed.

startVerb : STRING = *Start*

The word is used to prefix a step description for the start operation.

stopOrder : INTEGER = 50

The order in which a synchronize after modify stop will be executed.

stopVerb : STRING = *Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : INTEGER = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : INTEGER = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : INTEGER = 65

The order in which a synchronize after modify step will be executed.

createScript : STRING

Python script invoked to deploy a Java EE artifact or create a Java EE resource

destroyScript : STRING

Python script invoked to undeploy a Java EE artifact or destroy a Java EE resource

exposeDeployedApplication : BOOLEAN = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

inspectScript : **STRING**

Python script invoked to inspect a Java EE artifact or Java EE resource

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.Server

null

was.SharedLibrary

Hierarchy [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >>
python.PythonManagedDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

Interfaces udm.Deployed, udm.ConfigurationItem

Container-wide shared library that can be used by deployed applications

Public Properties

classPath : **STRING**

Classpath that contains the JAR files and directories



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

The container on which this deployed runs.

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

The deployable that this deployed is derived from.

nativePath : **STRING**

Optional path to any native libraries (DLL or SO)

Hidden Properties

createOrder : INTEGER = 60

The order in which a create step will be executed.

createScript : STRING = *was/sharedlibrary/create-shared-library.py*

Python wsadmin script invoked to create a shared library.

createVerb : STRING = *Create*

The word is used to prefix a step description for the create operation.

destroyOrder : INTEGER = 40

The order in which a destroy step will be executed.

destroyScript : STRING = *was/sharedlibrary/destroy-shared-library.py*

Python wsadmin script invoked to destroy a shared library.

destroyVerb : STRING = *Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : STRING = *was/sharedlibrary/inspect-shared-library.py*

Python wsadmin script invoked to inspect a shared library.

inspectVerb : STRING = *Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : INTEGER = 60

The order in which a modify step will be executed.

modifyVerb : STRING = *Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : STRING = *id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : INTEGER = 50

The order in which a start step will be executed.

startVerb : STRING = *Start*

The word is used to prefix a step description for the start operation.

stopOrder : INTEGER = 50

The order in which a synchronize after modify stop will be executed.

stopVerb : STRING = *Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : INTEGER = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : INTEGER = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : INTEGER = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.SharedLibrarySpec

Hierarchy [was.Deployable](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

Interfaces [udm.Taggable](#), [udm.Deployable](#), [udm.ConfigurationItem](#)

Specification of a shared library.

Public Properties

classPath : **STRING**

Classpath that contains the JAR files and directories (string)

nativePath : **STRING**

Optional path to any native libraries (DLL or SO) (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

was.SibDestination

Hierarchy [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >>
python.PythonManagedDeployed >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Base class for all deployed SIB destination configuration items.

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

additionalPropertiesNotToExpose : *STRING = sibType, busName*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 58*

The order in which a create step will be executed.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 42*

The order in which a destroy step will be executed.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : *INTEGER = 65*

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : *INTEGER = 45*

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : *INTEGER = 65*

The order in which a synchronize after modify step will be executed.

createScript : *STRING*

Python script invoked to deploy a Java EE artifact or create a Java EE resource

destroyScript : *STRING*

Python script invoked to undeploy a Java EE artifact or destroy a Java EE resource

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

inspectScript : **STRING**

Python script invoked to inspect a Java EE artifact or Java EE resource

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.SibQueue

Hierarchy **was.JmsResource** >> **was.Resource** >> **was.ExtensibleDeployedResource** >> **was.ExtensibleDeployed** >> **python.PythonManagedDeployed** >> **udm.BaseDeployed** >> **udm.BaseConfigurationItem**

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

JMS queue is used as a destination for point-to-point messaging. Use JMS queue destination administrative objects to manage JMS queues for the default messaging provider.

Public Properties



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

The container on which this deployed runs.

deliveryMode : **STRING** = *Application*

The delivery mode for messages. Legal values are 'Application', 'NonPersistent' and 'Persistent'

description : **STRING**

Description of the SIB JMS queue

jndiName : **STRING**

JNDI name for the resource

queueName : **CI**<**was.SibQueueDestination**>

Name of the underlying SIB queue to which the queue points

timeToLive : **STRING**

The time in milliseconds to be used for message expiration

busName : **STRING**

Name of the bus on which the queue resides

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

The deployable that this deployed is derived from.

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jmsProvider, jmsType*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/jms/create-sib-queue.py*

Python wsadmin script invoked to create a JMS SIB queue.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/jms/destroy-sib-queue-or-topic.py*

Python wsadmin script invoked to destroy a JMS SIB queue.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/jms/inspect-sib-queue.py*

Python wsadmin script invoked to inspect a JMS SIB queue.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : *INTEGER = 65*

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : *INTEGER = 45*

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = *65*

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.SibQueueConnectionFactory

Hierarchy [was.JmsResource](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

JMS queue connection factory is used to create connections to the associated JMS provider of JMS queues, for point-to-point messaging. Use queue connection factory administrative objects to manage JMS queue connection factories for the default messaging provider.

Public Properties

busName : **STRING**

Name of the bus on which the queue connection factory resides

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

The container on which this deployed runs.

jndiName : **STRING**

JNDI name for the resource

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

The deployable that this deployed is derived from.

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jmsProvider, jmsType*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/jms/create-sib-connection-factory.py*

Python wsadmin script invoked to create a JMS SIB queue connection factory.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/jms/destroy-sib-connection-factory.py*

Python wsadmin script invoked to destroy a JMS SIB queue connection factory.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/jms/inspect-sib-connection-factory.py*

Python wsadmin script invoked to inspect a JMS SIB queue connection factory.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

jmsType : *STRING = Queue*

JMS type of SIB queue connection factory.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : *INTEGER = 65*

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : **INTEGER** = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.SibQueueConnectionFactorySpec

Hierarchy [was.Deployable](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

Interfaces [udm.Taggable](#), [udm.Deployable](#), [udm.ConfigurationItem](#)

Specification of a SIB queue connection factory.

Public Properties

busName : **STRING**

Name of the bus on which the queue connection factory resides (string)

jndiName : **STRING**

JNDI name for the resource (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

was.SibQueueDestination

Hierarchy [was.SibDestination](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Queue for point-to-point messaging.

Public Properties

busName : `STRING`

Name of the bus on which the queue resides



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

additionalPropertiesNotToExpose : *STRING = sibType, busName*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 58*

The order in which a create step will be executed.

createScript : *STRING = was/jms/create-sib-destination.py*

Python wsadmin script invoked to create a JMS SIB queue destination.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 42*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/jms/destroy-sib-destination.py*

Python wsadmin script invoked to destroy a JMS SIB queue destination.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

sibType : *STRING = Queue*

SIB type of SIB queue destination.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : *INTEGER = 65*

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : *INTEGER = 45*

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = *65*

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

inspectScript : **STRING**

Python script invoked to inspect a Java EE artifact or Java EE resource

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.SibQueueDestinationSpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification of a SIB queue destination.

Public Properties

busName : **STRING**

Name of the bus on which the queue resides (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

was.SibQueueSpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification of a SIB queue.

Public Properties**busName** : *STRING*

Name of the bus on which the queue resides (string)

deliveryMode : *STRING*

The delivery mode for messages. Legal values are 'Application', 'NonPersistent' and 'Persistent' (string)

description : *STRING*

Description of the SIB JMS queue (string)

jndiName : *STRING*

JNDI name for the resource (string)

tags : *SET_OF_STRING*

The tags to map deployables to containers.

timeToLive : *STRING*

The time in milliseconds to be used for message expiration (string)

was.SibTopic

Hierarchy [was.JmsResource](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

JMS topic is used as a destination for publish/subscribe messaging.

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

The container on which this deployed runs.

jndiName : *STRING*

JNDI name for the resource

topicName : *STRING*

Topic to be used inside the topic space (for example, stock/IBM)

topicSpace : *CI<was.SibTopicSpaceDestination>*

Topic space into which to create the topic

busName : *STRING*

Name of the bus on which the topic resides

deployable : *CI<udm.Deployable>*

The deployable that this deployed is derived from.

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jmsProvider, jmsType*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/jms/create-sib-topic.py*

Python wsadmin script invoked to create a JMS SIB topic.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/jms/destroy-sib-queue-or-topic.py*

Python wsadmin script invoked to destroy a JMS SIB topic.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/jms/inspect-sib-topic.py*

Python wsadmin script invoked to inspect a JMS SIB topic.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : *INTEGER = 65*

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : *INTEGER = 45*

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = *65*

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.SibTopicConnectionFactory

Hierarchy **was.JmsResource** >> **was.Resource** >> **was.ExtensibleDeployedResource** >> **was.ExtensibleDeployed** >> **python.PythonManagedDeployed** >> **udm.BaseDeployed** >> **udm.BaseConfigurationItem**

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

JMS topic connection factory is used to create connections to the associated JMS provider of JMS topics, for publish/subscribe messaging. Use topic connection factory administrative objects to manage JMS topic connection factories for the default messaging provider.

Public Properties

busName : **STRING**

Name of the bus on which the topic connection factory resides

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

The container on which this deployed runs.

jndiName : **STRING**

JNDI name for the resource

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

The deployable that this deployed is derived from.

Hidden Properties

additionalPropertiesNotToExpose : *STRING = `jmsProvider, jmsType`*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = `60`*

The order in which a create step will be executed.

createScript : *STRING = `was/jms/create-sib-connection-factory.py`*

Python wsadmin script invoked to create a JMS SIB topic connection factory.

createVerb : *STRING = `Create`*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = `40`*

The order in which a destroy step will be executed.

destroyScript : *STRING = `was/jms/destroy-sib-connection-factory.py`*

Python wsadmin script invoked to destroy a JMS SIB topic connection factory.

destroyVerb : *STRING = `Destroy`*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = `was/jms/inspect-sib-connection-factory.py`*

Python wsadmin script invoked to inspect a JMS SIB topic connection factory.

inspectVerb : *STRING = `Inspect`*

The word is used to prefix a step description for the inspect operation.

jmsType : *STRING = `Topic`*

JMS type of SIB topic connection factory.

modifyOrder : *INTEGER = `60`*

The order in which a modify step will be executed.

modifyVerb : *STRING = `Modify`*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = `id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb`*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = `50`*

The order in which a start step will be executed.

startVerb : *STRING = `Start`*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = `50`*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = `Stop`*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : *INTEGER = `65`*

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : **INTEGER** = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.SibTopicConnectionFactorySpec

Hierarchy [was.Deployable](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

Interfaces [udm.Taggable](#), [udm.Deployable](#), [udm.ConfigurationItem](#)

Specification of a SIB topic connection factory.

Public Properties

busName : **STRING**

Name of the bus on which the topic connection factory resides (string)

jndiName : **STRING**

JNDI name for the resource (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

was.SibTopicSpaceDestination

Hierarchy [was.SibDestination](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Topic space is a location for publish/subscribe messaging.

Public Properties

busName : `STRING`

Name of the bus on which the topic resides



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

additionalPropertiesNotToExpose : *STRING = sibType, busName*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 58*

The order in which a create step will be executed.

createScript : *STRING = was/jms/create-sib-destination.py*

Python wsadmin script invoked to create a JMS SIB topic destination.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 42*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/jms/destroy-sib-destination.py*

Python wsadmin script invoked to destroy a JMS SIB topic destination.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

sibType : *STRING = TopicSpace*

SIB type of SIB topic destination.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : *INTEGER = 65*

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : *INTEGER = 45*

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = *65*

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

inspectScript : **STRING**

Python script invoked to inspect a Java EE artifact or Java EE resource

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.SibTopicSpaceDestinationSpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification of a SIB topic space destination.

Public Properties

busName : **STRING**

Name of the bus on which the topic resides (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

was.SibTopicSpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification of a SIB topic.

Public Properties**busName** : *STRING*

Name of the bus on which the topic resides (string)

jndiName : *STRING*

JNDI name for the resource (string)

tags : *SET_OF_STRING*

The tags to map deployables to containers.

topicName : *STRING*

Topic to be used inside the topic space (for example, stock/IBM) (string)

was.UnmanagedServer**Hierarchy** [was.BaseCell](#) >> udm.BaseContainer >> udm.BaseConfigurationItem**Interfaces** [was.WasAppContainer](#), udm.Tagable, python.PythonManagingContainer, [was.Server](#), python.PythonManagedContainer, [was.WasContainer](#), udm.ConfigurationItem, [was.Cell](#), [was.Node](#), udm.Container

An unmanaged WebSphere Application Server (WAS Base/SA)

Public Properties**cellName** : *STRING*

Name of the WebSphere cell, e.g. MyCell, WASCell, Cell01

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

Host on which the unmanaged WAS server runs

nodeName : *STRING*

Name of the WebSphere node

version : *ENUM [WAS_61, WAS_70, WAS_80]*

Version of WebSphere Application Server.

**wasHome** : *STRING*

Root path of the WebSphere installation path. e.g. /opt/ws/6.1/appserver/profiles/AppSrv01

**password** : *STRING*

Password which is used to login to the WebSphere Administration.

**port** : *INTEGER*

TCP port which is used to login to the WebSphere Administration, default is 8880

tags : *SET_OF_STRING*

The tags to map deployables to containers.

**username** : *STRING*

Username which is used to login to the WebSphere Administration.

Hidden Properties

restartServerScript : **STRING** = *was/server/restart-server.py*

Restart Server Script

runWithDaemon : **BOOLEAN** = *true*

Set to true to execute commands with the Python daemon

was.V5DefaultQueue

Hierarchy [was.JmsResource](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Queue destinations provided for point-to-point messaging by the internal WebSphere JMS provider. Use WebSphere Queue Destination administrative objects to manage queue destinations for the internal WebSphere JMS provider.

NOTE: The queue name must also be added to the list of queue names in the configuration of the JMS server(s) where the queue is to be hosted.

Public Properties



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

The container on which this deployed runs.

jndiName : **STRING**

JNDI name for the resource

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

The deployable that this deployed is derived from.

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jmsProvider, jmsType*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/jms/create-jms-object.py*

Python wsadmin script invoked to create a JMS resource.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/jms/destroy-jms-object.py*

Python wsadmin script invoked to destroy a JMS resource.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/jms/inspect-jms-object.py*

Python wsadmin script invoked to inspect a JMS resource.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

jmsProvider : *STRING = WebSphere JMS Provider*

JMS provider for V5 default queue.

jmsType : *STRING = WASQueue*

JMS type of V5 default queue.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : **INTEGER** = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : **INTEGER** = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.V5DefaultQueueConnectionFactory

Hierarchy [was.JmsResource](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Specifies a topic connection factory, which is used to create connections to the associated JMS provider of JMS queue destinations for point-to-point messaging. Use WebSphere queue connection factory administrative objects to manage queue connection factories for the internal WebSphere JMS provider.

Public Properties

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

The container on which this deployed runs.

jndiName : **STRING**

JNDI name for the resource

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

The deployable that this deployed is derived from.

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jmsProvider, jmsType*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/jms/create-jms-object.py*

Python wsadmin script invoked to create a JMS resource.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/jms/destroy-jms-object.py*

Python wsadmin script invoked to destroy a JMS resource.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/jms/inspect-jms-object.py*

Python wsadmin script invoked to inspect a JMS resource.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

jmsProvider : *STRING = WebSphere JMS Provider*

JMS provider for V5 default queue connection factory.

jmsType : *STRING = WASQueueConnectionFactory*

JMS type of V5 default queue connection factory.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : **INTEGER** = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : **INTEGER** = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.V5DefaultQueueConnectionFactorySpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification of a V5 default queue connection factory.

Public Properties

jndiName : **STRING**

JNDI name for the resource (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

was.V5DefaultQueueSpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification of a V5 default queue.

Public Properties

jndiName : **STRING**

JNDI name for the resource (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

was.V5DefaultTopic

Hierarchy [was.JmsResource](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Specifies the topic destinations for publish and subscribe messaging by the internal WebSphere JMS provider. Use WebSphere topic destination administrative objects to manage topic destinations for the internal WebSphere JMS provider.

Public Properties



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

The container on which this deployed runs.

jndiName : [STRING](#)

JNDI name for the resource

topic : [STRING](#)

String value used to identify the topic

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

The deployable that this deployed is derived from.

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jmsProvider, jmsType*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/jms/create-jms-object.py*

Python wsadmin script invoked to create a JMS resource.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/jms/destroy-jms-object.py*

Python wsadmin script invoked to destroy a JMS resource.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/jms/inspect-jms-object.py*

Python wsadmin script invoked to inspect a JMS resource.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

jmsProvider : *STRING = WebSphere JMS Provider*

JMS provider for V5 default topic.

jmsType : *STRING = WASTopic*

JMS type of V5 default topic.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : **INTEGER** = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : **INTEGER** = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.V5DefaultTopicConnectionFactory

Hierarchy [was.JmsResource](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Specifies a topic connection factory, which is used to create connections to the associated JMS provider of JMS topic destinations for publish and subscribe messaging. Use WebSphere topic connection factory administrative objects to manage topic connection factories for the internal WebSphere JMS provider.

Public Properties

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

The container on which this deployed runs.

jndiName : **STRING**

JNDI name for the resource

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

The deployable that this deployed is derived from.

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jmsProvider, jmsType*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/jms/create-jms-object.py*

Python wsadmin script invoked to create a JMS resource.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/jms/destroy-jms-object.py*

Python wsadmin script invoked to destroy a JMS resource.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/jms/inspect-jms-object.py*

Python wsadmin script invoked to inspect a JMS resource.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

jmsProvider : *STRING = WebSphere JMS Provider*

JMS provider for V5 default topic connection factory.

jmsType : *STRING = WASTopicConnectionFactory*

JMS type of V5 default topic connection factory.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : **INTEGER** = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : **INTEGER** = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.V5DefaultTopicConnectionFactorySpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification of a V5 default topic connection factory.

Public Properties

jndiName : **STRING**

JNDI name for the resource (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

was.V5DefaultTopicSpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification of a V5 default topic.

Public Properties

jndiName : `STRING`

JNDI name for the resource (string)

tags : `SET_OF_STRING`

The tags to map deployables to containers.

topic : `STRING`

String value used to identify the topic (string)

was.VirtualHost

Hierarchy `was.Resource` >> `was.ExtensibleDeployedResource` >> `was.ExtensibleDeployed` >>
`python.PythonManagedDeployed` >> `udm.BaseDeployed` >> `udm.BaseConfigurationItem`

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

Virtual host with a unique set of Web access ports. Such a configuration lets a single host machine resemble multiple host machines. Each virtual host has a logical name and a list of one or more domain name system (DNS) aliases by which it is known.

Public Properties

aliases : `SET_OF_STRING`

Virtual host aliases - enter alias as: hostname:port



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 = 60

The order in which a create step will be executed.

createScript : STRING = *was/virtualhost/create-virtual-host.py*

Python wsadmin script invoked to create a virtual host.

createVerb : STRING = *Create*

The word is used to prefix a step description for the create operation.

destroyOrder : INTEGER = 40

The order in which a destroy step will be executed.

destroyScript : STRING = *was/virtualhost/destroy-virtual-host.py*

Python wsadmin script invoked to destroy a virtual host.

destroyVerb : STRING = *Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : STRING = *was/virtualhost/inspect-virtual-host.py*

Python wsadmin script invoked to inspect a virtual host.

inspectVerb : STRING = *Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : INTEGER = 60

The order in which a modify step will be executed.

modifyVerb : STRING = *Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : STRING = *id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : INTEGER = 50

The order in which a start step will be executed.

startVerb : STRING = *Start*

The word is used to prefix a step description for the start operation.

stopOrder : INTEGER = 50

The order in which a synchronize after modify stop will be executed.

stopVerb : STRING = *Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : INTEGER = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : INTEGER = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : INTEGER = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.VirtualHostSpec

Hierarchy **was.Deployable** >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification of a virtual host.

Public Properties

aliases : **SET_OF_STRING**

Virtual host aliases - enter alias as: hostname:port (set_of_string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

was.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

Java EE WAR archive

Public Properties

contextRoot : *STRING*

Specifies the context root of the Web application (WAR). (string)

excludeFileNamesRegex : *STRING*

Regular expression that matches file names that must be excluded from scanning

placeholders : *SET_OF_STRING*

Placeholders detected in this artifact

preCompileJsps : *STRING*

Specify whether to precompile JavaServer Pages (JSP) files as a part of installation. The default is not to precompile JSP files. (boolean)

roleMappings : *MAP_STRING_STRING*

Security role to user/group mapping (map_string_string)

scanPlaceholders : *BOOLEAN* = *true*

Scan Placeholders

startingWeight : *STRING*

Specifies the order in which modules are started when the server starts. The module with the lowest starting weight is started first. (integer)

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

was.WarModule

Hierarchy *was.Module* >> *was.ExtensibleDeployedArtifact* >> *was.ExtensibleDeployed* >>

python.PythonManagedDeployed >> *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`

Specifies the context root of the Web application (WAR).

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 or

preCompileJsps : `BOOLEAN = false`

Specify whether to precompile JavaServer Pages (JSP) files as a part of installation. The default is not to precompile JSP files.

roleMappings : `MAP_STRING_STRING`

Security role to user/group mapping

sharedLibraries : `SET_OF_CI<was.SharedLibrary>`

Shared libraries used by this application

startingWeight : `INTEGER = -1`

Specifies the order in which modules are started when the server starts. The module with the lowest starting weight is started first.

virtualHost : `CI<was.VirtualHost>`

Specify the virtual host where you want to install the Web modules that are contained in your application. You can install Web modules on the same virtual host or disperse them among several hosts.

webServers : `SET_OF_CI<was.ManagedWebServer>`

Specify the Web servers as targets that serve as routers for requests to this application.

Hidden Properties

createOrder : INTEGER = 70

The order in which a create step will be executed.

createScript : STRING = *was/application/deploy-application.py*

Python wsadmin script invoked to create a Java EE artifact.

createVerb : STRING = *Deploy*

The word is used to prefix a step description for the create operation.

destroyOrder : INTEGER = 30

The order in which a destroy step will be executed.

destroyScript : STRING = *was/application/undeploy-application.py*

Python wsadmin script invoked to destroy a Java EE artifact.

destroyVerb : STRING = *Undeploy*

The word is used to prefix a step description for the destroy operation.

inspectVerb : STRING = *Inspect*

The word is used to prefix a step description for the inspect operation.

modifyOrder : INTEGER = 70

The order in which a modify step will be executed.

modifyVerb : STRING = *Upgrade*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : STRING = *id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : INTEGER = 90

The order in which a start step will be executed.

startScript : STRING = *was/application/start-application.py*

Python wsadmin script invoked to start running a Java EE artifact

startVerb : STRING = *Start*

The word is used to prefix a step description for the start operation.

stopOrder : INTEGER = 10

The order in which a stop step will be executed.

stopScript : STRING = *was/application/stop-application.py*

Python wsadmin script invoked to stop running a Java EE artifact.

stopVerb : STRING = *Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : INTEGER = 75

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : INTEGER = 35

Sync After Destroy Order**syncAfterModifyOrder** : **INTEGER** = 75

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = false

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

inspectScript : **STRING**

Python script invoked to inspect a Java EE artifact or Java EE resource

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

was.WasAppContainer

null

was.WasContainer

null

was.WmqQueue

Hierarchy **was.JmsResource** >> **was.Resource** >> **was.ExtensibleDeployedResource** >> **was.ExtensibleDeployed** >> **python.PythonManagedDeployed** >> **udm.BaseDeployed** >> **udm.BaseConfigurationItem**

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

Queue destinations provided for point-to-point messaging by the WebSphere MQ JMS provider. Use WebSphere MQ queue destination administrative objects to manage queue destinations for the WebSphere MQ JMS provider.

Public Properties**baseQueueName** : **STRING**

Name of the queue to which messages are sent

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

The container on which this deployed runs.

jndiName : **STRING**

JNDI name for the resource

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

The deployable that this deployed is derived from.

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jmsProvider, jmsType*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/jms/create-jms-object.py*

Python wsadmin script invoked to create a JMS resource.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/jms/destroy-jms-object.py*

Python wsadmin script invoked to destroy a JMS resource.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/jms/inspect-jms-object.py*

Python wsadmin script invoked to inspect a JMS resource.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

jmsProvider : *STRING = WebSphere MQ JMS Provider*

JMS provider for WebSphere MQ queue.

jmsType : *STRING = MQQueue*

JMS type of WebSphere MQ queue.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : **INTEGER** = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : **INTEGER** = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.WmqQueueConnectionFactory

Hierarchy [was.JmsResource](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Queue connection factory is used to create connections to the associated JMS provider of JMS queue destinations, for point-to-point messaging. Use WebSphere MQ queue connection factory administrative objects to manage queue connection factories for the WebSphere MQ JMS provider.

Public Properties



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

The container on which this deployed runs.

jndiName : **STRING**

JNDI name for the resource

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

The deployable that this deployed is derived from.

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jmsProvider, jmsType*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/jms/create-jms-object.py*

Python wsadmin script invoked to create a JMS resource.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/jms/destroy-jms-object.py*

Python wsadmin script invoked to destroy a JMS resource.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/jms/inspect-jms-object.py*

Python wsadmin script invoked to inspect a JMS resource.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

jmsProvider : *STRING = WebSphere MQ JMS Provider*

JMS provider for WebSphere MQ queue connection factory

jmsType : *STRING = MQQueueConnectionFactory*

JMS type of WebSphere MQ queue connection factory.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : **INTEGER** = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : **INTEGER** = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.WmqQueueConnectionFactorySpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification of a WMQ queue connection factory.

Public Properties

jndiName : **STRING**

JNDI name for the resource (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

was.WmqQueueSpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification of a WMQ queue.

Public Properties

baseQueueName : **STRING**

Name of the queue to which messages are sent (string)

jndiName : **STRING**

JNDI name for the resource (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

was.WmqTopic

Hierarchy [was.JmsResource](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Topic destinations provided for publish and subscribe messaging by the WebSphere MQ JMS provider. Use WebSphere MQ topic destination administrative objects to manage topic destinations for the WebSphere MQ JMS provider.

Public Properties

baseTopicName : **STRING**

Name of the topic to which messages are sent



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

The container on which this deployed runs.

jndiName : **STRING**

JNDI name for the resource

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

The deployable that this deployed is derived from.

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jmsProvider, jmsType*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/jms/create-jms-object.py*

Python wsadmin script invoked to create a JMS resource.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/jms/destroy-jms-object.py*

Python wsadmin script invoked to destroy a JMS resource.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/jms/inspect-jms-object.py*

Python wsadmin script invoked to inspect a JMS resource.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

jmsProvider : *STRING = WebSphere MQ JMS Provider*

JMS provider for WebSphere MQ topic.

jmsType : *STRING = MQTopic*

JMS type of WebSphere MQ topic.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : **INTEGER** = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : **INTEGER** = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.WmqTopicConnectionFactory

Hierarchy [was.JmsResource](#) >> [was.Resource](#) >> [was.ExtensibleDeployedResource](#) >> [was.ExtensibleDeployed](#) >> [python.PythonManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Topic connection factory is used to create connections to the associated JMS provider of JMS topic destinations, for publish and subscribe messaging. Use WebSphere MQ topic connection factory administrative objects to manage topic connection factories for the WebSphere MQ JMS provider.

Public Properties

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

The container on which this deployed runs.

jndiName : **STRING**

JNDI name for the resource

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

The deployable that this deployed is derived from.

Hidden Properties

additionalPropertiesNotToExpose : *STRING = jmsProvider, jmsType*

Properties that are not exposed to any python wsadmin script.

createOrder : *INTEGER = 60*

The order in which a create step will be executed.

createScript : *STRING = was/jms/create-jms-object.py*

Python wsadmin script invoked to create a JMS resource.

createVerb : *STRING = Create*

The word is used to prefix a step description for the create operation.

destroyOrder : *INTEGER = 40*

The order in which a destroy step will be executed.

destroyScript : *STRING = was/jms/destroy-jms-object.py*

Python wsadmin script invoked to destroy a JMS resource.

destroyVerb : *STRING = Destroy*

The word is used to prefix a step description for the destroy operation.

inspectScript : *STRING = was/jms/inspect-jms-object.py*

Python wsadmin script invoked to inspect a JMS resource.

inspectVerb : *STRING = Inspect*

The word is used to prefix a step description for the inspect operation.

jmsProvider : *STRING = WebSphere MQ JMS Provider*

JMS type of WebSphere MQ topic connection factory.

jmsType : *STRING = MQTopicConnectionFactory*

JMS type of WebSphere MQ topic connection factory.

modifyOrder : *INTEGER = 60*

The order in which a modify step will be executed.

modifyVerb : *STRING = Modify*

The word is used to prefix a step description for the modify operation.

standardPropertiesNotToExpose : *STRING = id, type, properties, deployable, container, createOrder, createScript, createVerb, syncAfterCreateOrder, modifyOrder, modifyScript, modifyVerb, syncAfterModifyOrder, destroyOrder, destroyScript, securityPermissions, inheritPermissions, exposeDeployedApplication, destroyVerb, syncAfterDestroyOrder, startOrder, startScript, startVerb, stopOrder, stopScript, stopVerb, inspectScript, inspectVerb*

Standard properties that are not exposed to any python wsadmin script.

startOrder : *INTEGER = 50*

The order in which a start step will be executed.

startVerb : *STRING = Start*

The word is used to prefix a step description for the start operation.

stopOrder : *INTEGER = 50*

The order in which a synchronize after modify stop will be executed.

stopVerb : *STRING = Stop*

The word is used to prefix a step description for the stop operation.

syncAfterCreateOrder : **INTEGER** = 65

The order in which a synchronize after create step will be executed.

syncAfterDestroyOrder : **INTEGER** = 45

The order in which a synchronize after destroy step will be executed.

syncAfterModifyOrder : **INTEGER** = 65

The order in which a synchronize after modify step will be executed.

exposeDeployedApplication : **BOOLEAN** = *false*

flag to indicate whether the deployed application CI is to be injected to the python script execution context.

modifyScript : **STRING**

Python script invoked to upgrade a Java EE artifact or modify a Java EE resource

startScript : **STRING**

Python script invoked to start a Java EE artifact or Java EE resource

stopScript : **STRING**

Python script invoked to stop a Java EE artifact or Java EE resource

was.WmqTopicConnectionFactorySpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification of a WMQ topic connection factory.

Public Properties

jndiName : **STRING**

JNDI name for the resource (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.

was.WmqTopicSpec

Hierarchy [was.Deployable](#) >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification of a WMQ topic.

Public Properties

baseTopicName : **STRING**

Name of the topic to which messages are sent (string)

jndiName : **STRING**

JNDI name for the resource (string)

tags : **SET_OF_STRING**

The tags to map deployables to containers.