

Deployit Oracle Glassfish Server Plugin Manual

Version 3.9.0

Table of Contents

Table of Contents	2
Preface	3
Overview	3
Features	3
Requirements	3
Usage in Deployment Packages	3
Using the deployables and deployed	4
Deployable vs. Container table	4
Deployed Actions Table	4
Deploying to Glassfish	4
Discovery	5
Extension points	5
Adding Additional Properties	5
Extending the plugin with custom control task	6
CI Reference	6
Configuration Item Overview	6
Deployables	6
Deployeds	6
Containers	7
Other Configuration Items	7
Configuration Item Details	8
glassfish.ApplicationRef	8
glassfish.ApplicationRefSpec	9
glassfish.Artifact	10
glassfish.CliBasedContainer	11
glassfish.CliManagedDeployed	12
glassfish.CliManagedDeployedArtifact	13
glassfish.Cluster	15
glassfish.Domain	16
glassfish.Domain_create	17
glassfish.Ear	17
glassfish.EarModule	18
glassfish.EjbJar	20
glassfish.EjbJarModule	21
glassfish.JMSConnectionFactory	23
glassfish.JMSDestination	25
glassfish.JMSResources	27
glassfish.JdbcConnectionPool	29
glassfish.JdbcConnectionPoolSpec	31
glassfish.JdbcResource	32
glassfish.JdbcResourceSpec	34
glassfish.JeeJdbcResource	35
glassfish.Queue	36
glassfish.QueueConnectionFactory	38
glassfish.QueueConnectionFactorySpec	40
glassfish.QueueSpec	41
glassfish.ResourceRef	41
glassfish.ResourceRefSpec	42
glassfish.StandaloneServer	43
glassfish.StandaloneServer_create	43
glassfish.Topic	44
glassfish.TopicConnectionFactory	45
glassfish.TopicConnectionFactorySpec	47
glassfish.TopicSpec	48
glassfish.War	48
glassfish.WarModule	49

Preface

This document describes the functionality provided by the Glassfish plugin.

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

Overview

The Glassfish plugin is a Deployit plugin that adds the capability to manage deployments and resources on Glassfish application server 3.x. The plugin has the capability of managing application artifacts, datasource and JMS resources via the Glassfish Cli, and can easily be extended to support more deployment options or management of new artifacts/resources on Glassfish.

Features

- Deploying to Domain, Standalone servers or clusters
- Deployment of application artifacts
 - Enterprise application (EAR)
 - Web application (WAR)
 - Enterprise Java beans (EJB)
 - Artifact references
- Deployment of resources
 - JDBC Connection Pools
 - JDBC Resources
 - JMS Connection Factories
 - JMS Queues
 - JMS Topics
 - Resource references
- Control tasks to
 - Create, destroy, start and stop domains and standalone servers
- Discovery of Domains, Clusters and Standalone servers

Requirements

- **Deployit requirements**
 - **Deployit:** version 3.9+
- **Infrastructural requirements**
 - **Glassfish versions:** 3.x
 - **Linux server**
 - **User credentials** for accessing the Host and Glassfish Cli.

Usage in Deployment Packages

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

The following is a sample deployit-manifest.xml file that can be used to create a Glassfish specific deployment package. It contains declarations for an [War](#), a [connection pool](#), and a [Jdbc resources](#). It also contains references to be able to target the deployables to specific containers.

```

<?xml version="1.0" encoding="UTF-8"?>
<udm:DeploymentPackage version="1.0" application="MyApp">
  <deployables>

    <glassfish.War name="myWarFile" file="myWarFile/PetClinic-1.0.war">
      <scanPlaceholders>false</scanPlaceholders>
    </glassfish.War>
    <glassfish.ApplicationRefSpec name="myWarRef">
      <applicationName>myWarFile</applicationName>
    </glassfish.ApplicationRefSpec>

    <glassfish.JdbcConnectionPoolSpec name="connPool">
      <datasourceclassname>com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource</datasourceclassname>
      <restype>javax.sql.DataSource</restype>
    </glassfish.JdbcConnectionPoolSpec>

    <glassfish.JdbcResourceSpec name="myJDBCResource">
      <jndiName>myJDBCResource</jndiName>
      <poolName>connPool</poolName>
    </glassfish.JdbcResourceSpec>
    <glassfish.ResourceRefSpec name="MyJDBCResourceRef">
      <resourceName>myJDBCResource</resourceName>
    </glassfish.ResourceRefSpec>

  </deployables>
</udm:DeploymentPackage>

```

Using the deployables and deployed

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

Deployable vs. Container table

Deployable	Container	Generated deployed
jee.Ear glassfish.Ear	glassfish.Domain	glassfish.EarModule
jee.War glassfish.War	glassfish.Domain	glassfish.WarModule
jee.EjbJar glassfish.EjbJar	glassfish.Domain	glassfish.EjbJarModule
glassfish.JdbcConnectionPoolSpec	glassfish.Domain	glassfish.JdbcConnectionPool
jee.DataSourceSpec glassfish.JdbcResourceSpec	glassfish.Domain	glassfish.JdbcResource
jee.ResourceSpec glassfish.QueueConnectionFactorySpec	glassfish.Domain	glassfish.QueueConnectionFactory
jee.ResourceSpec glassfish.TopicConnectionFactorySpec	glassfish.Domain	glassfish.TopicConnectionFactory
jee.ResourceSpec glassfish.QueueSpec	glassfish.Domain	glassfish.Queue
jee.ResourceSpec glassfish.TopicSpec	glassfish.Domain	glassfish.Topic
glassfish.ApplicationRefSpec	glassfish.RefContainer	glassfish.ApplicationRef
glassfish.ResourceRefSpec	glassfish.RefContainer	glassfish.ResourceRef

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

Deployed Actions Table

Deployed	Actions performed for operations		
	Create	Destroy	Modify
glassfish.EarModule glassfish.WarModule glassfish.EjbJarModule	<ul style="list-style-type: none"> upload artifact deploy application 	<ul style="list-style-type: none"> check if there are no references to the application undeploy old application version 	<ul style="list-style-type: none"> redploy application version
glassfish.JdbcConnectionPool glassfish.JdbcResource glassfish.QueueConnectionFactory glassfish.TopicConnectionFactory glassfish.Queue glassfish.Topic	<ul style="list-style-type: none"> create resource 	<ul style="list-style-type: none"> delete resource 	<ul style="list-style-type: none"> set properties
glassfish.JdbcConnectionPool glassfish.JdbcResource	<ul style="list-style-type: none"> create resource 	<ul style="list-style-type: none"> delete resource 	<ul style="list-style-type: none"> create resource delete resource

Deploying to Glassfish

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

Glassfish manages all the artifacts and resources in the domain. Therefore all artifacts and

resources must be deployed directly to the domain. To target an application or resource to a specific container, you can use references. There are two types of deployables that can be used to deploy references:

- **ApplicationRefSpec** This deployable can be used to target applications to containers.
- **ResourceRefSpec** This deployable can be used to target resources to containers.

The CI name for all deployables will be used as identifier for the application or resource in glassfish. The applications and resources are referenced by name.

- A application can only be undeployed when there are no references to it. So please undeploy all references to the application as well if you want to undeploy an application. The plugin checks if there are references, and if so it will give an error.

Discovery

The plugin supports discovery of Domains, Clusters, and Standalone Servers.

The Domain can be discovered through the Host that runs the Domain. Notice that the name of the CI should match the name of the Domain, Cluster or Standalone Server. The name of the container CI is used for the `--target` parameter of the Glassfish Cli.

- Deployit will never discover cluster members. You are able to deploy any kind of deployable directly to the cluster, so Deployit does not need to know about the instances of a cluster.
- Deployit will always discover the default Standalone Server of the domain called server.
- Deployit will only discover infrastructure CIs. No deployed CIs will be discovered.

Extension points

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

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

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

There also exists an advanced method to extend the plugin, but the implementation of this form of extension needs to be written in the Java programming language and consists of writing so-called `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. Also refer to the Overthere documentation for working with remote files.

Adding Additional Properties

Glassfish artifacts and resources support the concept of **Additional Properties**. These properties are normally specified by using the `--properties` argument of Glassfish Cli commands. Deployit can be extended very easily to add one or more **Additional Properties**. You can add them by extending a type synthetically. You need to add the property into the category "Additional Properties".

For example:

```
<type-modification type="glassfish.WarModule">
  <property name="keepSessions" kind="boolean" category="Additional Properties" default="true"/>
</type-modification>
```

This will result in adding the **Additional Property** `keepSessions` to be available on the CI, with a default value of `true`. This will result in deploying the application with the Glassfish Cli argument `--properties keepSessions=true`.

Extending the plugin with custom control task

The plugin has the capability to add control tasks to [CliManagedDeployed](#) or [CliManagedContainer](#). The control task can be specified as a jython script that will be executed on the Deployit Server. The jython script will execute `asadmin` commands on the remote host.

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

Synthetic.xml snippet

```
<type-modification type="glassfish.Domain">
  <method name="listClusters" label="List clusters" delegate="asadmin" script="list-clusters.py" >
</type-modification>
```

list-clusters.py snippet

```
logOutput("Listing clusters")
result = executeCmd('list-clusters')
logOutput(result.output)
logOutput("Done.")
```

The script will execute the `list-clusters` command using `asadmin` on the remote host, and print the result.

CI Reference

Configuration Item Overview

Deployables

CI	Description
glassfish.ApplicationRefSpec	Specification of an Application Reference
glassfish.Ear	A JEE EAR archive
glassfish.EjbJar	An EjbJar archive
glassfish.JdbcConnectionPoolSpec	A Glassfish jdbc connectionPool (deployable)
glassfish.JdbcResourceSpec	JDBC Resource
glassfish.QueueConnectionFactorySpec	a Glassfish QCF (deployable)
glassfish.QueueSpec	a Glassfish Queue (deployable)
glassfish.ResourceRefSpec	Specification of a Resource Reference
glassfish.TopicConnectionFactorySpec	a Glassfish QCF (deployable)
glassfish.TopicSpec	A Glassfish Topic (deployable)
glassfish.War	A JEE WAR archive

Deployeds

CI	Description
glassfish.ApplicationRef	Application reference is used to refer an application
glassfish.Artifact	Description unavailable
glassfish.CliManagedDeployed	Base for all deployed that utilize the Glassfish Cli for configuration
glassfish.CliManagedDeployedArtifact	Base for all deployed artifacts that utilize the Glassfish Cli for configuration
glassfish.EarModule	Ear with values configured for a deployment
glassfish.EjbJarModule	EjbJar with values configured for a deployment
glassfish.JMSConnectionFactory	Base class for all the JMS connection factories
glassfish.JMSDestination	Base class for all the JMS destinations
glassfish.JMSResources	Base class for all the JMS resources
glassfish.JdbcConnectionPool	A Glassfish jdbc connectionPool
glassfish.JdbcResource	JDBC Resource
glassfish.JeeJdbcResource	JDBC resource
glassfish.Queue	a Glassfish Queue
glassfish.QueueConnectionFactory	a Glassfish QCF
glassfish.ResourceRef	Resource reference is used to refer a glassfish resource
glassfish.Topic	A Glassfish Topic
glassfish.TopicConnectionFactory	a Glassfish QCF
glassfish.WarModule	War with values configured for a deployment

Containers

CI	Description
glassfish.CliBasedContainer	GlassFish Cli Managed Container
glassfish.Cluster	Glassfish Cluster
glassfish.Domain	Description unavailable
glassfish.StandaloneServer	Glassfish StandaloneServer

Other Configuration Items

CI	Description
glassfish.ApplicationRef	Application reference is used to refer an application
glassfish.ApplicationRefSpec	Specification of an Application Reference
glassfish.Artifact	Description unavailable
glassfish.CliBasedContainer	GlassFish Cli Managed Container
glassfish.CliManagedDeployed	Base for all deployed that utilize the Glassfish Cli for configuration
glassfish.CliManagedDeployedArtifact	Base for all deployed artifacts that utilize the Glassfish Cli for configuration
glassfish.Cluster	Glassfish Cluster
glassfish.Domain	Description unavailable
glassfish.Domain_create	Generated type for method create
glassfish.Ear	A JEE EAR archive
glassfish.EarModule	Ear with values configured for a deployment
glassfish.EjbJar	An EjbJar archive
glassfish.EjbJarModule	EjbJar with values configured for a deployment
glassfish.JMSConnectionFactory	Base class for all the JMS connection factories
glassfish.JMSDestination	Base class for all the JMS destinations
glassfish.JMSResources	Base class for all the JMS resources
glassfish.JdbcConnectionPool	A Glassfish jdbc connectionPool
glassfish.JdbcConnectionPoolSpec	A Glassfish jdbc connectionPool (deployable)
glassfish.JdbcResource	JDBC Resource
glassfish.JdbcResourceSpec	JDBC Resource
glassfish.JeeJdbcResource	JDBC resource
glassfish.Queue	a Glassfish Queue
glassfish.QueueConnectionFactory	a Glassfish QCF
glassfish.QueueConnectionFactorySpec	a Glassfish QCF (deployable)
glassfish.QueueSpec	a Glassfish Queue (deployable)
glassfish.ResourceRef	Resource reference is used to refer a glassfish resource
glassfish.ResourceRefSpec	Specification of a Resource Reference
glassfish.StandaloneServer	Glassfish StandaloneServer
glassfish.StandaloneServer_create	Generated type for method create
glassfish.Topic	A Glassfish Topic
glassfish.TopicConnectionFactory	a Glassfish QCF
glassfish.TopicConnectionFactorySpec	a Glassfish QCF (deployable)
glassfish.TopicSpec	A Glassfish Topic (deployable)
glassfish.War	A JEE WAR archive
glassfish.WarModule	War with values configured for a deployment


Configuration Item Details

glassfish.ApplicationRef

Type Hierarchy [glassfish.CliManagedDeployed](#) >> udm.BaseDeployed >> udm.BaseConfigurationItem

Interfaces udm.EmbeddedDeployedContainer, udm.Deployed, udm.ConfigurationItem

Application reference is used to refer an application.

Parent
 container : CI<udm.Container> The container on which this deployed runs.

Hidden Properties	
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 75	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/applicationref/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Create	Create Verb
* destroyOrder : INTEGER = 25	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/applicationref/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Destroy	Destroy Verb
* inspectScript : STRING = glassfish/applicationref/inspect.py	Classpath to the script used to inspect the generic container.
* modifyOrder : INTEGER = 75	The order of the step in the step list for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
libraryScripts : LIST_OF_STRING	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
modifyScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.ApplicationRefSpec

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

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

Specification of an Application Reference.

Public Properties	
applicationName :	STRING Specifies name of the application to be referred (string)
tags :	SET_OF_STRING If set, this deployable will only be mapped automatically to containers with the same tag.

glassfish.Artifact

Virtual Type

Type Hierarchy [glassfish.CliManagedDeployedArtifact](#) >>
[glassfish.CliManagedDeployed](#) >> [udm.BaseDeployed](#) >>
[udm.BaseConfigurationItem](#)

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

Description unavailable








Parent	
* container :	CI<udm.Container> The container on which this deployed runs.
Public Properties	
availabilityEnabled :	BOOLEAN = false Controls whether availability is enabled for web sessions and for stateful session bean (SFSB) checkpointing and potentially passivation
deployable :	CI<udm.Deployable> The deployable that this deployed is derived from.
description :	STRING Specifies the description of this application
force :	BOOLEAN = false Forces redeployment even if this application has already been deployed or already exists.
keepState :	BOOLEAN = false Retains web sessions, SFSB instances, and persistently created EJB timers between redeployments
lbenabled :	BOOLEAN = true This option controls whether the deployed application is available for load balancing. The default is true
libraries :	STRING A comma-separated list of library JAR files. Specify the library JAR files by their relative or absolute paths.
placeholders :	MAP_STRING_STRING A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>
verify :	BOOLEAN = false Verifies the syntax and semantics of the deployment descriptor. Verifier packages must be installed

Hidden Properties	
* applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 70	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/application/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Deploy	Create Verb
* destroyOrder : INTEGER = 30	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/application/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Undeploy	Destroy Verb
* inspectScript : STRING = glassfish/application/inspect-app.py	Classpath to the script used to inspect the generic container.
* libraryScripts : LIST_OF_STRING = [glassfish/application/application-lib.py]	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
* modifyOrder : INTEGER = 70	The order of the step in the step list for the modify operation.
* modifyScript : STRING = glassfish/application/modify.py	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
* modifyVerb : STRING = Update	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
enabled : BOOLEAN = true	Allows users to access the application.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.CliBasedContainer

Virtual Type
Type Hierarchy udm.BaseContainer >> udm.BaseConfigurationItem
Interfaces udm.Tagable, glassfish.CliManagingContainer, udm.ConfigurationItem, glassfish.CliManagedContainer, udm.Container, overthere.HostContainer


GlassFish Cli Managed Container

Parent	
 *	host : CI<overthere.Host> Host
Public Properties	
 *	home : STRING GlassFish home directory
 *	username : STRING Username which is used to login to DAS.
	adminHostAddress : STRING = localhost Host which is used to login to DAS, default is localhost
	password : STRING Password which is used to login to DAS.
	port : INTEGER = 4848 TCP port which is used to login to DAS, default is 4848
	secure : BOOLEAN = false Connection to DAS should be secure.
	tags : SET_OF_STRING If set, only deployables with the same tag will be automatically mapped to this container.
Hidden Properties	
	libraryScripts : LIST_OF_STRING = [glassfish/library/runtime.py , glassfish/library/commandHelper.py] List of python library scripts that should be automatically loaded when using a GlassFish CLI script.

glassfish.CliManagedDeployed

Virtual Type
Type Hierarchy udm.BaseDeployed >> udm.BaseConfigurationItem
Interfaces udm.EmbeddedDeployedContainer, udm.Deployed, udm.ConfigurationItem

Base for all deployed that utilize the Glassfish Cli for configuration

Parent	
 *	container : CI<udm.Container> The container on which this deployed runs.
Public Properties	
	deployable : CI<udm.Deployable> The deployable that this deployed is derived from.

Hidden Properties	
* applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 60	The order of the step in the step list for the create operation.
* createScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Create	Create Verb
* destroyOrder : INTEGER = 40	The order of the step in the step list for the destroy operation.
* destroyVerb : STRING = Destroy	Destroy Verb
* modifyOrder : INTEGER = 60	The order of the step in the step list for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
destroyScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
libraryScripts : LIST_OF_STRING	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
modifyScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.CliManagedDeployedArtifact

Virtual Type

Type Hierarchy [glassfish.CliManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

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

Parent

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

The container on which this deployed runs.

Hidden Properties	
* applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 60	The order of the step in the step list for the create operation.
* createScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Create	Create Verb
* destroyOrder : INTEGER = 40	The order of the step in the step list for the destroy operation.
* destroyVerb : STRING = Destroy	Destroy Verb
* modifyOrder : INTEGER = 60	The order of the step in the step list for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
destroyScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
libraryScripts : LIST_OF_STRING	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
modifyScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.Cluster

Type Hierarchy udm.BaseContainer >> udm.BaseConfigurationItem
Interfaces glassfish.RefContainer, udm.Taggable, glassfish.CliManagedContainer, udm.ConfigurationItem, udm.Container, overthere.HostContainer

Glassfish Cluster








Parent	
* domain : <code>CI<glassfish.Domain ></code>	Domain to which the server group belongs.
Public Properties	
tags : <code>SET_OF_STRING</code>	If set, only deployables with the same tag will be automatically mapped to this container.
Hidden Properties	
* restartOrder : <code>INTEGER = 80</code>	Restart Order
* startOrder : <code>INTEGER = 80</code>	Start Order
* startScript : <code>STRING = glassfish/container/start-cluster.py</code>	Start Script
* stopOrder : <code>INTEGER = 20</code>	Stop Order
* stopScript : <code>STRING = glassfish/container/stop-cluster.py</code>	Stop Script
Control task	Parameter CI Attributes Description
start	script = glassfish/container/start-cluster.py, delegate = asadmin Starts the Server instance
stop	script = glassfish/container/stop-cluster.py, delegate = asadmin Stops the Server instance

glassfish.Domain

Type Hierarchy [glassfish.CliBasedContainer](#) >> [udm.BaseContainer](#) >> [udm.BaseConfigurationItem](#)

Interfaces [udm.Taggable](#), [glassfish.CliManagingContainer](#), [glassfish.CliManagedContainer](#), [udm.ConfigurationItem](#), [udm.Container](#), [overthere.HostContainer](#)

Description unavailable

Parent			
	host	CI <overthere.Host>	Host
Children			
	clusters	SET_OF_CI <glassfish.Cluster >	StandaloneServer groups defined in domain
	servers	SET_OF_CI <glassfish.StandaloneServer >	Profiles defined in domain
Public Properties			
	home	STRING	GlassFish home directory
	username	STRING	Username which is used to login to DAS.
	adminHostAddress	STRING = localhost	Host which is used to login to DAS, default is localhost
	password	STRING	Password which is used to login to DAS.
	port	INTEGER = 4848	TCP port which is used to login to DAS, default is 4848
	secure	BOOLEAN = false	Connection to DAS should be secure.
	tags	SET_OF_STRING	If set, only deployables with the same tag will be automatically mapped to this container.

Hidden Properties			
<p>* inspectScript : STRING = glassfish/container/inspect-domain.py Classpath to the script used to inspect the domain.</p> <p>libraryScripts : LIST_OF_STRING = [glassfish/library/runtime.py, glassfish/library/commandHelper.py] List of python library scripts that should be automatically loaded when using a GlassFish CLI script.</p>			
Control task	Parameter CI	Attributes	Description
create	glassfish.Domain_create	script = glassfish/container/create-domain.py, delegate = asadmin	Creates a domain
delete		script = glassfish/container/delete-domain.py, delegate = asadmin	Deletes a domain
start		script = glassfish/container/start-domain.py, delegate = asadmin	Starts the Domain Administration Server of the specified domain
stop		script = glassfish/container/stop-domain.py, delegate = asadmin	Stops the Domain Administration Server of the specified domain

glassfish.Domain_create

Type Hierarchy udm.Parameters >> udm.BaseConfigurationItem

Interfaces udm.ConfigurationItem

Generated type for method create.

Public Properties	
domainProperties :	STRING Setting the optional name/value pairs overrides the default values for the properties of the domain to be created
instancePort :	INTEGER This HTTP port specifies where the web application context roots are available for a web browser to connect to

glassfish.Ear

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

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

A JEE EAR archive

Public Properties	
availabilityEnabled :	STRING Controls whether availability is enabled for web sessions and for stateful session bean (SFSB) checkpointing and potentially passivation (boolean)
checksum :	STRING The checksum used to detect differences on the artifact. If not provided, it will be calculated by Deployit.
compatibility :	STRING Supports the backward compatibility of JAR visibility in v2 instead of the stricter Java EE 6 requirements implemented in v3. (boolean)
description :	STRING Specifies the description of this application (string)
excludeFileNamesRegex :	STRING Regular expression that matches file names that must be excluded from scanning
force :	STRING Forces redeployment even if this application has already been deployed or already exists. (boolean)
javaWebStartEnabled :	STRING Specifies whether Java Web Start access is permitted for an application client module. (boolean)
keepState :	STRING Retains web sessions, SFSB instances, and persistently created EJB timers between redeployments (boolean)
lbenabled :	STRING This option controls whether the deployed application is available for load balancing. The default is true (boolean)
libraries :	STRING A comma-separated list of library JAR files. Specify the library JAR files by their relative or absolute paths. (string)
placeholders :	SET_OF_STRING Placeholders detected in this artifact
precompileJsp :	STRING Precompiles JSP pages during deployment. (boolean)
scanPlaceholders :	BOOLEAN = false Whether to scan this artifact for placeholders when it is imported
tags :	SET_OF_STRING If set, this deployable will only be mapped automatically to containers with the same tag.
verify :	STRING Verifies the syntax and semantics of the deployment descriptor. Verifier packages must be installed (boolean)
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
delimiters :	STRING = {{ }} The delimiters used indicate placeholders, defaults to '{{ }}'. This is a 5 character string with a space in the middle, the first two are the leading delimiter, the last two are the closing delimiter

glassfish.EarModule

Type Hierarchy [glassfish.Artifact](#) >> [glassfish.CliManagedDeployedArtifact](#) >> [glassfish.CliManagedDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Ear with values configured for a deployment

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

Public Properties	
availabilityEnabled : BOOLEAN = false	Controls whether availability is enabled for web sessions and for stateful session bean (SFSB) checkpointing and potentially passivation
compatibility : BOOLEAN = false	Supports the backward compatibility of JAR visibility in v2 instead of the stricter Java EE 6 requirements implemented in v3.
deployable : CI<udm.Deployable>	The deployable that this deployed is derived from.
description : STRING	Specifies the description of this application
force : BOOLEAN = false	Forces redeployment even if this application has already been deployed or already exists.
javaWebStartEnabled : BOOLEAN = true	Specifies whether Java Web Start access is permitted for an application client module.
keepState : BOOLEAN = false	Retains web sessions, SFSB instances, and persistently created EJB timers between redeployments
lbenabled : BOOLEAN = true	This option controls whether the deployed application is available for load balancing. The default is true
libraries : STRING	A comma-separated list of library JAR files. Specify the library JAR files by their relative or absolute paths.
placeholders : MAP_STRING_STRING	A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>
precompileJsp : BOOLEAN = false	Precompiles JSP pages during deployment.
verify : BOOLEAN = false	Verifies the syntax and semantics of the deployment descriptor. Verifier packages must be installed

Hidden Properties	
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed.
STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.	
* createOrder : INTEGER = 70	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/application/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Deploy	Create Verb
* destroyOrder : INTEGER = 30	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/application/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Undeploy	Destroy Verb
* extension : STRING = ear	Extension
* inspectScript : STRING = glassfish/application/inspect-app.py	Classpath to the script used to inspect the generic container.
* libraryScripts : LIST_OF_STRING = [glassfish/application/application-lib.py]	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
* modifyOrder : INTEGER = 70	The order of the step in the step list for the modify operation.
* modifyScript : STRING = glassfish/application/modify.py	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
* modifyVerb : STRING = Update	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
enabled : BOOLEAN = true	Allows users to access the application.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.EjbJar

Type 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

An EjbJar archive

Public Properties	
availabilityEnabled : STRING	Controls whether availability is enabled for web sessions and for stateful session bean (SFSB) checkpointing and potentially passivation (boolean)
checksum : STRING	The checksum used to detect differences on the artifact. If not provided, it will be calculated by Deployit.
compatibility : STRING	Supports the backward compatibility of JAR visibility in v2 instead of the stricter Java EE 6 requirements implemented in v3. (boolean)
description : STRING	Specifies the description of this application (string)
excludeFileNamesRegex : STRING	Regular expression that matches file names that must be excluded from scanning
force : STRING	Forces redeployment even if this application has already been deployed or already exists. (boolean)
keepState : STRING	Retains web sessions, SFSB instances, and persistently created EJB timers between redeployments (boolean)
lbenabled : STRING	This option controls whether the deployed application is available for load balancing. The default is true (boolean)
libraries : STRING	A comma-separated list of library JAR files. Specify the library JAR files by their relative or absolute paths. (string)
placeholders : SET_OF_STRING	Placeholders detected in this artifact
scanPlaceholders : BOOLEAN = false	Whether to scan this artifact for placeholders when it is imported
tags : SET_OF_STRING	If set, this deployable will only be mapped automatically to containers with the same tag.
verify : STRING	Verifies the syntax and semantics of the deployment descriptor. Verifier packages must be installed (boolean)
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
delimiters : STRING = {{ }}	The delimiters used indicate placeholders, defaults to '{{ }}'. This is a 5 character string with a space in the middle, the first two are the leading delimiter, the last two are the closing delimiter

glassfish.EjbJarModule

Type Hierarchy	glassfish.Artifact >> glassfish.CliManagedDeployedArtifact >> glassfish.CliManagedDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem
Interfaces	udm.EmbeddedDeployedContainer, udm.Artifact, udm.Deployed, udm.ConfigurationItem, udm.DerivedArtifact

EjbJar with values configured for a deployment

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

Public Properties	
availabilityEnabled : BOOLEAN = false	Controls whether availability is enabled for web sessions and for stateful session bean (SFSB) checkpointing and potentially passivation
compatibility : BOOLEAN = false	Supports the backward compatibility of JAR visibility in v2 instead of the stricter Java EE 6 requirements implemented in v3.
deployable : CI<udm.Deployable>	The deployable that this deployed is derived from.
description : STRING	Specifies the description of this application
force : BOOLEAN = false	Forces redeployment even if this application has already been deployed or already exists.
keepState : BOOLEAN = false	Retains web sessions, SFSB instances, and persistently created EJB timers between redeployments
lbenabled : BOOLEAN = true	This option controls whether the deployed application is available for load balancing. The default is true
libraries : STRING	A comma-separated list of library JAR files. Specify the library JAR files by their relative or absolute paths.
placeholders : MAP_STRING_STRING	A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>
verify : BOOLEAN = false	Verifies the syntax and semantics of the deployment descriptor. Verifier packages must be installed

Hidden Properties	
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 70	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/application/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Deploy	Create Verb
* destroyOrder : INTEGER = 30	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/application/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Undeploy	Destroy Verb
* extension : STRING = jar	Extension
* inspectScript : STRING = glassfish/application/inspect-app.py	Classpath to the script used to inspect the generic container.
* libraryScripts : LIST_OF_STRING = [glassfish/application/application-lib.py]	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
* modifyOrder : INTEGER = 70	The order of the step in the step list for the modify operation.
* modifyScript : STRING = glassfish/application/modify.py	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
* modifyVerb : STRING = Update	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
enabled : BOOLEAN = true	Allows users to access the application.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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


glassfish.JMSConnectionFactory

Virtual Type

Type Hierarchy [glassfish.JMSResources](#) >> [glassfish.CliManagedDeployed](#) >>
udm.BaseDeployed >> udm.BaseConfigurationItem

Interfaces udm.EmbeddedDeployedContainer, udm.Deployed,
udm.ConfigurationItem

Base class for all the JMS connection factories

Parent	
	container : CI<udm.Container> The container on which this deployed runs.
Public Properties	
	AddressList : STRING A comma-separated list of message queue addresses that specify the host names (and, optionally, port numbers) of a message broker instance
	ClientId : STRING A client ID for a connection factory that will be used by a durable subscriber
	Name : STRING The name of the physical destination to which the resource will refer
	Password : STRING The password for the connection factory
	UserName : STRING The user name for the connection factory
	deployable : CI<udm.Deployable> The deployable that this deployed is derived from.
	description : STRING Text providing details about the JMS resource

Hidden Properties	
* applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 60	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/jmsconnectionfactory/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Create	Create Verb
* destroyOrder : INTEGER = 40	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/jmsconnectionfactory/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Destroy	Destroy Verb
* inspectScript : STRING = glassfish/jmsconnectionfactory/inspect.py	Classpath to the script used to inspect the generic container.
* modifyOrder : INTEGER = 60	The order of the step in the step list for the modify operation.
* modifyScript : STRING = glassfish/jmsconnectionfactory/modify.py	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
libraryScripts : LIST_OF_STRING	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.JMSDestination

Virtual Type

Type Hierarchy [glassfish.JMSResources](#) >> [glassfish.CliManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Base class for all the JMS destinations

Parent

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

Hidden Properties	
* applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 60	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/jmsdestination/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Create	Create Verb
* destroyOrder : INTEGER = 40	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/jmsdestination/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Destroy	Destroy Verb
* inspectScript : STRING = glassfish/jmsdestination/inspect.py	Classpath to the script used to inspect the generic container.
* modifyOrder : INTEGER = 60	The order of the step in the step list for the modify operation.
* modifyScript : STRING = glassfish/jmsdestination/modify.py	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
libraryScripts : LIST_OF_STRING	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.JMSResources

Virtual Type

Type Hierarchy [glassfish.CliManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Base class for all the JMS resources

Parent

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

Hidden Properties	
* applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 60	The order of the step in the step list for the create operation.
* createScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Create	Create Verb
* destroyOrder : INTEGER = 40	The order of the step in the step list for the destroy operation.
* destroyVerb : STRING = Destroy	Destroy Verb
* modifyOrder : INTEGER = 60	The order of the step in the step list for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
destroyScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
libraryScripts : LIST_OF_STRING	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
modifyScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.JdbcConnectionPool

Type Hierarchy [glassfish.CliManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

A Glassfish jdbc connectionPool

Parent

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

Hidden Properties	
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 60	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/jdbcconnectionpool/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Create	Create Verb
* destroyOrder : INTEGER = 70	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/jdbcconnectionpool/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Destroy	Destroy Verb
* inspectScript : STRING = glassfish/jdbcconnectionpool/inspect.py	Classpath to the script used to inspect the generic container.
* libraryScripts : LIST_OF_STRING = [glassfish/jdbcconnectionpool/jdbc-connectionpool-lib.py]	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
* modifyOrder : INTEGER = 60	The order of the step in the step list for the modify operation.
* modifyScript : STRING = glassfish/jdbcconnectionpool/modify.py	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* propertyCategory : STRING = Properties	Property Category
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.JdbcConnectionPoolSpec

Type Hierarchy `jee.ResourceSpec >> udm.BaseDeployable >> udm.BaseConfigurationItem`

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

A Glassfish jdbc connectionPool (deployable)

Public Properties	
databaseName : STRING	Specifies the database for this connection pool (string)
datasourceName : STRING	Specifies an underlying XADataSource, or a ConnectionPoolDataSource if connection pooling is done (string)
datasourceclassname : STRING	Vendor-specific classname that implements the DataSource and/or XADataSource APIs (string)
description : STRING	Description of the JDBC Connection Pool (string)
driverclassname : STRING	Vendor-specific classname that implements the java.sql.Driver interface. (string)
idletimeout : STRING	Maximum time that connection can remain idle in the pool(in seconds) (integer)
isolationguaranteed : STRING	All connections use same isolation level; requires Transaction Isolation (boolean)
maxpoolsize : STRING	Maximum number of connections that can be created to satisfy client requests (integer)
maxwait : STRING	Amount of time caller waits before connection timeout is sent(in milliseconds) (integer)
nontransactionalconnections : STRING	Returns non-transactional connections (boolean)
password : STRING	Specifies the password for connecting to the database (string)
ping : STRING	When enabled, the pool is pinged during creation or reconfiguration to identify and warn of any erroneous values for its attributes (boolean)
poolresize : STRING	Number of connections to be removed when pool idle timeout expires (integer)
port : STRING	Specifies the port on which the database server listens for requests (string)
restype : STRING	Must be specified if the datasource class implements more than 1 of the interface(javax.sql.DataSource, javax.sql.XADataSource, or javax.sql.ConnectionPoolDataSource). (enum)
serverName : STRING	Specifies the database server for this connection pool (string)
steadypoolsize : STRING	Minimum and initial number of connections maintained in the pool (integer)
tags : SET_OF_STRING	If set, this deployable will only be mapped automatically to containers with the same tag.
url : STRING	Specifies the URL for this connection pool. Although this is not a standard property, it is commonly used (string)
user : STRING	Specifies the user name for connecting to the database (string)

glassfish.JdbcResource

Type Hierarchy `glassfish.JeeJdbcResource >> glassfish.CliManagedDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem`

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

JDBC Resource. This deployed is used when a glassfish.JdbcResourceSpec is specified in a package.

Parent
<p>* container : CI<udm.Container> The container on which this deployed runs.</p>
Public Properties
<p>* jndiName : STRING Specifies the JNDI name for the JDBC resource</p>
<p>* poolName : STRING The name of the JDBC connection pool. If two or more JDBC resource elements point to the same connection pool element, they use the same pool connection at runtime.</p>
<p>deployable : CI<udm.Deployable> The deployable that this deployed is derived from.</p>

Hidden Properties	
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 65	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/jdbcresource/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Create	Create Verb
* destroyOrder : INTEGER = 35	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/jdbcresource/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Destroy	Destroy Verb
* inspectScript : STRING = glassfish/jdbcresource/inspect.py	Classpath to the script used to inspect the generic container.
* modifyOrder : INTEGER = 60	The order of the step in the step list for the modify operation.
* modifyScript : STRING = glassfish/jdbcresource/modify.py	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* propertyCategory : STRING = Properties	Property Category
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
libraryScripts : LIST_OF_STRING	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.JdbcResourceSpec

Type Hierarchy	jee.DataSourceSpec >> jee.JndiResourceSpec >> jee.ResourceSpec >> udm.BaseDeployable >> udm.BaseConfigurationItem
Interfaces	udm.Taggable, udm.Deployable, udm.ConfigurationItem

JDBC Resource. This deployed is used when a `glassfish.JdbcResourceSpec` is specified in a package. (deployable)


Public Properties	
jndiName :	STRING Specifies the JNDI name for the JDBC resource (string)
poolName :	STRING The name of the JDBC connection pool. If two or more JDBC resource elements point to the same connection pool element, they use the same pool connection at runtime. (string)
tags :	SET_OF_STRING If set, this deployable will only be mapped automatically to containers with the same tag.

glassfish.JeeJdbcResource

Type Hierarchy [glassfish.CliManagedDeployed](#) >> `udm.BaseDeployed` >> `udm.BaseConfigurationItem`

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

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

Parent	
 container :	CI<udm.Container> The container on which this deployed runs.

Hidden Properties	
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 65	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/jdbcresource/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Create	Create Verb
* destroyOrder : INTEGER = 35	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/jdbcresource/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Destroy	Destroy Verb
* inspectScript : STRING = glassfish/jdbcresource/inspect.py	Classpath to the script used to inspect the generic container.
* modifyOrder : INTEGER = 60	The order of the step in the step list for the modify operation.
* modifyScript : STRING = glassfish/jdbcresource/modify.py	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* propertyCategory : STRING = Properties	Property Category
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
libraryScripts : LIST_OF_STRING	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.Queue

Type Hierarchy [glassfish.JMSDestination](#) >> [glassfish.JMSResources](#) >>
[glassfish.CliManagedDeployed](#) >> [udm.BaseDeployed](#) >>
[udm.BaseConfigurationItem](#)

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

a Glassfish Queue

Parent	
* container :	CI<udm.Container> The container on which this deployed runs.
Public Properties	
* jndiName :	STRING Specifies the JNDI name for the resource
AddressList :	STRING A comma-separated list of message queue addresses that specify the host names (and, optionally, port numbers) of a message broker instance
ClientId :	STRING A client ID for a connection factory that will be used by a durable subscriber
Name :	STRING The name of the physical destination to which the resource will refer
Password :	STRING The password for the connection factory
UserName :	STRING The user name for the connection factory
deployable :	CI<udm.Deployable> The deployable that this deployed is derived from.


Hidden Properties	
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 60	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/jmsdestination/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Create	Create Verb
* destroyOrder : INTEGER = 40	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/jmsdestination/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Destroy	Destroy Verb
* inspectScript : STRING = glassfish/jmsdestination/inspect.py	Classpath to the script used to inspect the generic container.
* modifyOrder : INTEGER = 60	The order of the step in the step list for the modify operation.
* modifyScript : STRING = glassfish/jmsdestination/modify.py	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* restype : STRING = javax.jms.Queue	The JMS resource type, which is javax.jms.Queue
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
libraryScripts : LIST_OF_STRING	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.QueueConnectionFactory

Type Hierarchy [glassfish.JMSConnectionFactory](#) >> [glassfish.JMSResources](#) >>
[glassfish.CliManagedDeployed](#) >> [udm.BaseDeployed](#) >>
[udm.BaseConfigurationItem](#)

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

a Glassfish QCF

Parent	
	container : CI<udm.Container> The container on which this deployed runs.
Public Properties	
AddressList :	STRING A comma-separated list of message queue addresses that specify the host names (and, optionally, port numbers) of a message broker instance
ClientId :	STRING A client ID for a connection factory that will be used by a durable subscriber
Name :	STRING The name of the physical destination to which the resource will refer
Password :	STRING The password for the connection factory
UserName :	STRING The user name for the connection factory
deployable :	CI<udm.Deployable> The deployable that this deployed is derived from.
description :	STRING Text providing details about the JMS resource

Hidden Properties	
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 60	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/jmsconnectionfactory/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Create	Create Verb
* destroyOrder : INTEGER = 40	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/jmsconnectionfactory/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Destroy	Destroy Verb
* inspectScript : STRING = glassfish/jmsconnectionfactory/inspect.py	Classpath to the script used to inspect the generic container.
* modifyOrder : INTEGER = 60	The order of the step in the step list for the modify operation.
* modifyScript : STRING = glassfish/jmsconnectionfactory/modify.py	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* restype : STRING = javax.jms.QueueConnectionFactory	The JMS resource type, which is javax.jms.QueueConnectionFactory
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
libraryScripts : LIST_OF_STRING	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.QueueConnectionFactorySpec

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

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

a Glassfish QCF (deployable)

Public Properties	
AddressList :	STRING A comma-separated list of message queue addresses that specify the host names (and, optionally, port numbers) of a message broker instance (string)
ClientId :	STRING A client ID for a connection factory that will be used by a durable subscriber (string)
Name :	STRING The name of the physical destination to which the resource will refer (string)
Password :	STRING The password for the connection factory (string)
UserName :	STRING The user name for the connection factory (string)
description :	STRING Text providing details about the JMS resource (string)
tags :	SET_OF_STRING If set, this deployable will only be mapped automatically to containers with the same tag.

glassfish.QueueSpec

Type Hierarchy `jee.ResourceSpec >> udm.BaseDeployable >> udm.BaseConfigurationItem`

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

a Glassfish Queue (deployable)

Public Properties	
AddressList :	STRING A comma-separated list of message queue addresses that specify the host names (and, optionally, port numbers) of a message broker instance (string)
ClientId :	STRING A client ID for a connection factory that will be used by a durable subscriber (string)
Name :	STRING The name of the physical destination to which the resource will refer (string)
Password :	STRING The password for the connection factory (string)
UserName :	STRING The user name for the connection factory (string)
jndiName :	STRING Specifies the JNDI name for the resource (string)
tags :	SET_OF_STRING If set, this deployable will only be mapped automatically to containers with the same tag.

glassfish.ResourceRef

Type Hierarchy `glassfish.CliManagedDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem`

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

Resource reference is used to refer a glassfish resource

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

Hidden Properties	
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 70	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/resourceref/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Create	Create Verb
* destroyOrder : INTEGER = 25	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/resourceref/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Destroy	Destroy Verb
* inspectScript : STRING = glassfish/resourceref/inspect.py	Classpath to the script used to inspect the generic container.
* modifyOrder : INTEGER = 65	The order of the step in the step list for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
libraryScripts : LIST_OF_STRING	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
modifyScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.ResourceRefSpec

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

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

Specification of a Resource Reference.

Public Properties	
resourceName :	STRING Specifies the name/jndiname of the glassfish resource to be referred (string)
tags :	SET_OF_STRING If set, this deployable will only be mapped automatically to containers with the same tag.

glassfish.StandaloneServer

Type Hierarchy udm.BaseContainer >> udm.BaseConfigurationItem

Interfaces glassfish.RefContainer, udm.Taggable, glassfish.CliManagedContainer, udm.ConfigurationItem, udm.Container, overthere.HostContainer

Glassfish StandaloneServer

Parent	
* domain :	CI<glassfish.Domain > Domain to which the server group belongs.
Public Properties	
host :	CI<overthere.Host> Host on which server resides. When empty, the host on which the domain resides is used.
tags :	SET_OF_STRING If set, only deployables with the same tag will be automatically mapped to this container.
Hidden Properties	
* restartOrder :	INTEGER = 80 Restart Order
* startOrder :	INTEGER = 80 Start Order
* startScript :	STRING = glassfish/container/start-instance.py Start Script
* stopOrder :	INTEGER = 20 Stop Order
* stopScript :	STRING = glassfish/container/stop-instance.py Stop Script

Control task	Parameter CI	Attributes	Description
create	glassfish.StandaloneServer_create	script = glassfish/container/create-instance.py, delegate = asadmin	Creates a server instance
delete		script = glassfish/container/delete-instance.py, delegate = asadmin	Deletes a server instance
start		script = glassfish/container/start-instance.py, delegate = asadmin	Starts the Server instance
stop		script = glassfish/container/stop-instance.py, delegate = asadmin	Stops the Server instance

glassfish.StandaloneServer_create

Type Hierarchy udm.Parameters >> udm.BaseConfigurationItem

Interfaces udm.ConfigurationItem

Generated type for method create.

Public Properties	
* node : STRING	Name of the node on which the instance will reside
config : STRING	Specifies the named configuration that the instance references.uses the default configuration, if you do not provide any.
systemProperties : STRING	Setting the optional name/value pairs overrides the default values for the properties of the server to be created

glassfish.Topic

Type Hierarchy [glassfish.JMSDestination](#) >> [glassfish.JMSResources](#) >> [glassfish.CliManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

A Glassfish Topic

Parent	
* container : CI<udm.Container>	The container on which this deployed runs.
Public Properties	
* jndiName : STRING	Specifies the JNDI name for the resource
AddressList : STRING	A comma-separated list of message queue addresses that specify the host names (and, optionally, port numbers) of a message broker instance
ClientId : STRING	A client ID for a connection factory that will be used by a durable subscriber
Name : STRING	The name of the physical destination to which the resource will refer
Password : STRING	The password for the connection factory
UserName : STRING	The user name for the connection factory
deployable : CI<udm.Deployable>	The deployable that this deployed is derived from.


Hidden Properties	
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 60	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/jmsdestination/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Create	Create Verb
* destroyOrder : INTEGER = 40	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/jmsdestination/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Destroy	Destroy Verb
* inspectScript : STRING = glassfish/jmsdestination/inspect.py	Classpath to the script used to inspect the generic container.
* modifyOrder : INTEGER = 60	The order of the step in the step list for the modify operation.
* modifyScript : STRING = glassfish/jmsdestination/modify.py	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* restype : STRING = javax.jms.Topic	The JMS resource type, which is javax.jms.Topic
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
libraryScripts : LIST_OF_STRING	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.TopicConnectionFactory

Type Hierarchy [glassfish.JMSConnectionFactory](#) >> [glassfish.JMSResources](#) >> [glassfish.CliManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

a Glassfish QCF

Parent	
	container : CI<udm.Container> The container on which this deployed runs.
Public Properties	
AddressList :	STRING A comma-separated list of message queue addresses that specify the host names (and, optionally, port numbers) of a message broker instance
ClientId :	STRING A client ID for a connection factory that will be used by a durable subscriber
Name :	STRING The name of the physical destination to which the resource will refer
Password :	STRING The password for the connection factory
UserName :	STRING The user name for the connection factory
deployable :	CI<udm.Deployable> The deployable that this deployed is derived from.
description :	STRING Text providing details about the JMS resource

Hidden Properties	
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 60	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/jmsconnectionfactory/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Create	Create Verb
* destroyOrder : INTEGER = 40	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/jmsconnectionfactory/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Destroy	Destroy Verb
* inspectScript : STRING = glassfish/jmsconnectionfactory/inspect.py	Classpath to the script used to inspect the generic container.
* modifyOrder : INTEGER = 60	The order of the step in the step list for the modify operation.
* modifyScript : STRING = glassfish/jmsconnectionfactory/modify.py	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* restype : STRING = javax.jms.TopicConnectionFactory	The JMS resource type, which can be javax.jms.TopicConnectionFactory
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
libraryScripts : LIST_OF_STRING	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

glassfish.TopicConnectionFactorySpec

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

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

a Glassfish QCF (deployable)

Public Properties	
AddressList :	STRING A comma-separated list of message queue addresses that specify the host names (and, optionally, port numbers) of a message broker instance (string)
ClientId :	STRING A client ID for a connection factory that will be used by a durable subscriber (string)
Name :	STRING The name of the physical destination to which the resource will refer (string)
Password :	STRING The password for the connection factory (string)
UserName :	STRING The user name for the connection factory (string)
description :	STRING Text providing details about the JMS resource (string)
tags :	SET_OF_STRING If set, this deployable will only be mapped automatically to containers with the same tag.

glassfish.TopicSpec

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

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

A Glassfish Topic (deployable)

Public Properties	
AddressList :	STRING A comma-separated list of message queue addresses that specify the host names (and, optionally, port numbers) of a message broker instance (string)
ClientId :	STRING A client ID for a connection factory that will be used by a durable subscriber (string)
Name :	STRING The name of the physical destination to which the resource will refer (string)
Password :	STRING The password for the connection factory (string)
UserName :	STRING The user name for the connection factory (string)
jndiName :	STRING Specifies the JNDI name for the resource (string)
tags :	SET_OF_STRING If set, this deployable will only be mapped automatically to containers with the same tag.

glassfish.War

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

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

A JEE WAR archive

Public Properties	
availabilityEnabled :	STRING Controls whether availability is enabled for web sessions and for stateful session bean (SFSB) checkpointing and potentially passivation (boolean)
checksum :	STRING The checksum used to detect differences on the artifact. If not provided, it will be calculated by Deployit.
contextRoot :	STRING Path relative to server's base URL. (string)
description :	STRING Specifies the description of this application (string)
excludeFileNamesRegex :	STRING Regular expression that matches file names that must be excluded from scanning
force :	STRING Forces redeployment even if this application has already been deployed or already exists. (boolean)
keepState :	STRING Retains web sessions, SFSB instances, and persistently created EJB timers between redeployments (boolean)
lbenabled :	STRING This option controls whether the deployed application is available for load balancing. The default is true (boolean)
libraries :	STRING A comma-separated list of library JAR files. Specify the library JAR files by their relative or absolute paths. (string)
placeholders :	SET_OF_STRING Placeholders detected in this artifact
precompileJsp :	STRING Precompiles JSP pages during deployment. (boolean)
scanPlaceholders :	BOOLEAN = false Whether to scan this artifact for placeholders when it is imported
tags :	SET_OF_STRING If set, this deployable will only be mapped automatically to containers with the same tag.
verify :	STRING Verifies the syntax and semantics of the deployment descriptor. Verifier packages must be installed (boolean)
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
delimiters :	STRING = {{ }} The delimiters used indicate placeholders, defaults to '{{ }}'. This is a 5 character string with a space in the middle, the first two are the leading delimiter, the last two are the closing delimiter

glassfish.WarModule

Type Hierarchy [glassfish.Artifact](#) >> [glassfish.CliManagedDeployedArtifact](#) >> [glassfish.CliManagedDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

War with values configured for a deployment

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

Public Properties	
availabilityEnabled : BOOLEAN = false	Controls whether availability is enabled for web sessions and for stateful session bean (SFSB) checkpointing and potentially passivation
contextRoot : STRING	Path relative to server's base URL.
deployable : CI<udm.Deployable>	The deployable that this deployed is derived from.
description : STRING	Specifies the description of this application
force : BOOLEAN = false	Forces redeployment even if this application has already been deployed or already exists.
keepState : BOOLEAN = false	Retains web sessions, SFSB instances, and persistently created EJB timers between redeployments
lbenabled : BOOLEAN = true	This option controls whether the deployed application is available for load balancing. The default is true
libraries : STRING	A comma-separated list of library JAR files. Specify the library JAR files by their relative or absolute paths.
placeholders : MAP_STRING_STRING	A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>
precompileJsp : BOOLEAN = false	Precompiles JSP pages during deployment.
verify : BOOLEAN = false	Verifies the syntax and semantics of the deployment descriptor. Verifier packages must be installed

Hidden Properties	
* containerRestartStrategy : ENUM [NONE, STOP_START, RESTART] = NONE	The logical stop/start sequence for the container affected by the deployed. STOP_START : stop container, undeploy, deploy, start container. RESTART : undeploy, deploy, stop container, start container.
* createOrder : INTEGER = 70	The order of the step in the step list for the create operation.
* createScript : STRING = glassfish/application/deploy.py	Classpath to the script that is uploaded and executed on the generic container for the create operation.
* createVerb : STRING = Deploy	Create Verb
* destroyOrder : INTEGER = 30	The order of the step in the step list for the destroy operation.
* destroyScript : STRING = glassfish/application/undeploy.py	Classpath to the script that is uploaded and executed on the generic container for the destroy operation.
* destroyVerb : STRING = Undeploy	Destroy Verb
* extension : STRING = war	Extension
* inspectScript : STRING = glassfish/application/inspect-app.py	Classpath to the script used to inspect the generic container.
* libraryScripts : LIST_OF_STRING = [glassfish/application/application-lib.py]	List of python library scripts that should be automatically loaded when using a Glassfish CLI script.
* modifyOrder : INTEGER = 70	The order of the step in the step list for the modify operation.
* modifyScript : STRING = glassfish/application/modify.py	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
* modifyVerb : STRING = Update	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* restartStrategy : ENUM [NONE, STOP, START, STOP_START, RESTART] = NONE	The logical stop/start sequence for the deployed. STOP_START : stop, undeploy, deploy, start. STOP: stop, undeploy, deploy. START: undeploy, deploy, start.
* startOrder : INTEGER = 90	The order in which a start step will be executed.
* startVerb : STRING = Start	The word that is used to prefix a step description for the start operation.
* stopOrder : INTEGER = 10	The order in which a stop step will be executed.
* stopVerb : STRING = Stop	The word that is used to prefix a step description for the stop operation.
applyRestartStrategyOnNoop : BOOLEAN = false	When true, the restart strategy is apply for a NOOP operation
enabled : BOOLEAN = true	Allows users to access the application.
noopScript : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
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

