

# Deployit Tomcat Plugin Manual

Version 3.9.0

## Table of Contents

Table of Contents	2
Preface	4
Overview	4
Features	4
Requirements	4
Usage in Deployment Packages	4
Tomcat Topology	5
Tomcat Server	5
Tomcat Common Context	5
Tomcat Virtual Host	5
Using the deployables and deployed	5
Deployable vs. Container table	5
Deployed Actions Table	5
Deploying Web Applications	6
Deploying Resources	6
Extension points	6
Adding additional context attributes	6
Adding new context elements	7
CI Reference	7
Configuration Item Overview	7
Deployables	7
Deployeds	8
Containers	9
Other Configuration Items	9
Configuration Item Details	11
jee.DataSourceSpec	11
jee.Ear	11
jee.EjbJar	12
jee.JndiResourceSpec	12
jee.MailSessionSpec	12
jee.QueueConnectionFactorySpec	12
jee.QueueSpec	13
jee.ResourceSpec	13
jee.TopicConnectionFactorySpec	13
jee.TopicSpec	13
jee.War	14
tomcat.ActiveMqConnectionFactory	14
tomcat.ActiveMqConnectionFactorySpec	16
tomcat.ActiveMqQueue	16
tomcat.ActiveMqQueueSpec	18
tomcat.ActiveMqTopic	18
tomcat.ActiveMqTopicSpec	20
tomcat.CommonContext	20
tomcat.ContextContainer	20
tomcat.ContextElement	21
tomcat.ContextWarModule	22
tomcat.DataSource	25
tomcat.DataSourceLink	27
tomcat.DataSourceLinkSpec	29
tomcat.DataSourceSpec	29
tomcat.JeeActiveMqConnectionFactory	30
tomcat.JeeActiveMqQueue	32
tomcat.JeeActiveMqTopic	34
tomcat.JeeWebsphereMqQueue	36
tomcat.JeeWebsphereMqQueueConnectionFactory	38
tomcat.JeeWebsphereMqTopic	40
tomcat.JeeWebsphereMqTopicConnectionFactory	42

tomcat.JndiContextElement	44
tomcat.MailSession	46
tomcat.MailSessionSpec	48
tomcat.ResourceLink	48
tomcat.ResourceLinkSpec	50
tomcat.Server	50
tomcat.VirtualHost	51
tomcat.WarModule	52
tomcat.WebsphereMqQueue	54
tomcat.WebsphereMqQueueConnectionFactory	56
tomcat.WebsphereMqQueueConnectionFactorySpec	58
tomcat.WebsphereMqQueueSpec	58
tomcat.WebsphereMqTopic	58
tomcat.WebsphereMqTopicConnectionFactory	61
tomcat.WebsphereMqTopicConnectionFactorySpec	63
tomcat.WebsphereMqTopicSpec	63

## Preface

This document describes the functionality provided by the Tomcat plugin.

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

## Overview

The Tomcat plugin is a Deployit plugin that adds capability for managing deployments and resources to a Tomcat Server. Standard support includes deploying/undeploying web applications, datasources, mail sessions, resource links, ActiveMQ and WebsphereMQFeatures resources. The plugin can easily be extended to support more deployment options or management of new artifacts/resources on Tomcat.

## Features

- Deployment of web applications (WAR) to a Tomcat virtual host.
- Deployment of resources to an application context on a Tomcat virtual host.
- Deployment of resources to the common Tomcat context ie. \$TOMCAT\_HOME/conf/context.xml.
- Deployment of resource links.
- Supported JEE Resources
  - Datasource
  - JMS Queue
  - JMS Queue Connection Factory
  - JMS Topic
  - JMS Topic Connection Factory
  - Mail Session
- Supported Messaging Middleware
  - Active MQ
  - Websphere MQ
- Support for Tomcat Database Connection Pool (DBCP) Configurations
- Stopping and starting a Tomcat Server via control tasks.

## Requirements

- **Deployit requirements**
  - **Deployit:** version 3.6+
  - **Tomcat versions:** 5.5.x, 6.x, 7.x (Unix and Windows)
- **Infrastructural requirements**
  - **User credentials** for accessing the Host running the Tomcat Server.

## Usage in Deployment Packages

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

The following is a sample MANIFEST.MF file that can be used to create a Tomcat specific deployment package. It contain declarations for a [war](#), a [datasource](#) and a couple of JMS resources.

```
Manifest-Version: 1.0
Deployit-Package-Format-Version: 1.3
CI-Application: SampleApp
CI-Version: 1.0

Name: SampleApp-1.0.war
CI-Name: SampleApp
CI-Type: jee.War

Name: testDatasource
```

```

CI-Type: tomcat.DataSourceSpec
CI-jndiName: jdbc/sampleDatasource
CI-url: jdbc:mysql://localhost/test
CI-driverClassName: com.mysql.jdbc.Driver
CI-username: {{DATABASE_USERNAME}}
CI-password: {{DATABASE_PASSWORD}}

Name: sampleQueue
CI-Type: jee.QueueSpec
CI-jndiNames: jms/testQueue

Name: sampleQCf
CI-Type: jee.QueueConnectionFactorySpec
CI-jndiNames: jms/sampleQCf

```

## Tomcat Topology

The Tomcat plugin allows for the modelling of a Tomcat installation into the Deployit infrastructure as a hierarchical set of containers. [Tomcat Server](#), [Tomcat Common Context](#) and [Tomcat Virtual Hosts](#) containers can then be included into Deployit environment definitions, to which deployables can be deployed to.

### Tomcat Server

Models a Tomcat installation running on a host. This container must be defined under any [Host](#) in the Deployit infrastructure. The container supports OS specific stop and start commands used to control the stopping and starting of the Tomcat Server. In addition, stop and start wait times can be specified. Deployit will wait for the specified amount of time to elapse after the execution of a stop/start command.

### Tomcat Common Context

Models the context.xml file present in the \$TOMCAT\_HOME/conf directory. This container must be defined under a [Tomcat Server](#) container. Only a single instance should be defined. Any [Tomcat Resource](#) can be deployed to this container.

### Tomcat Virtual Host

Models a Tomcat virtual host definition, ie. \$TOMCAT\_HOME/conf/[enginename]/[hostname]. This container must be defined under a [Tomcat Server](#) container. Multiple instances can be defined. [Web Applications](#) and [Tomcat Resource](#) can be deployed to this container.

## Using the deployables and deployments

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

### Deployable vs. Container table

Deployable	Container	Deployed
jee.War	tomcat.VirtualHost	tomcat.WarModule
tomcat.ResourceLinkSpec	tomcat.VirtualHost tomcat.CommonContext	tomcat.ResourceLink
tomcat.DataSourceLinkSpec	tomcat.VirtualHost tomcat.CommonContext	tomcat.DataSourceLink
jee.DataSourceSpec	tomcat.VirtualHost tomcat.CommonContext	tomcat.DataSource
jee.MailSessionSpec	tomcat.VirtualHost tomcat.CommonContext	tomcat.MailSession
jee.QueueConnectionFactorySpec tomcat.ActiveMqConnectionFactorySpec tomcat.WebsphereMqQueueConnectionFactorySpec	tomcat.VirtualHost tomcat.CommonContext	tomcat.ActiveMqConnectionFactory tomcat.WebsphereMqQueueConnectionFactory
jee.TopicConnectionFactorySpec tomcat.WebsphereMqTopicConnectionFactorySpec	tomcat.VirtualHost tomcat.CommonContext	tomcat.WebsphereMqTopicConnectionFactory
jee.QueueSpec tomcat.ActiveMqQueueSpec tomcat.WebsphereMqQueueSpec	tomcat.VirtualHost tomcat.CommonContext	tomcat.ActiveMqQueue tomcat.WebsphereMqQueue
jee.TopicSpec tomcat.ActiveMqTopicSpec tomcat.WebsphereMqTopicSpec	tomcat.VirtualHost tomcat.CommonContext	tomcat.ActiveMqTopic tomcat.WebsphereMqTopic

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

### Deployed Actions Table

Deployed	Actions performed for operations		
	Create	Destroy	Modify
tomcat.WarModule	<ul style="list-style-type: none"> <li>Stop Tomcat</li> <li>Create a context xml using the context root name specified on the war in the virtual host's directory. eg.</li> </ul>	<ul style="list-style-type: none"> <li>Stop Tomcat</li> <li>Delete war from</li> </ul>	<ul style="list-style-type: none"> <li>Stop Tomcat</li> <li>Delete war from virtual host's application base directory.</li> </ul>

	<ul style="list-style-type: none"> <li>• \$TOMCAT_HOME/conf/Catalina/localhost</li> <li>• Upload war to virtual host's application base directory. eg. \$TOMCAT_HOME/webapps</li> <li>• Start Tomcat</li> </ul>	<ul style="list-style-type: none"> <li>• virtual host's application base directory.</li> <li>• Delete exploded war folder from virtual host's application base directory.</li> <li>• Remove attributes, e.g. docBase, from the Context element in the application specific context xml</li> <li>• Start Tomcat</li> </ul>	<ul style="list-style-type: none"> <li>• Delete exploded war folder from virtual host's application base directory.</li> <li>• Remove attributes, e.g. docBase, from the Context element in the application specific context xml</li> <li>• Upload war to virtual host's application base directory. eg. \$TOMCAT_HOME/webapps</li> <li>• Start Tomcat</li> </ul>
Any tomcat.JndiContextElement (tomcat.DataSource, tomcat.ActiveMqQueue, tomcat.ResourceLink, etc)	<ul style="list-style-type: none"> <li>• Stop Tomcat</li> <li>• Create xml fragment, e.g. &lt;Resource name='mail/Session' auth='Container' type='javax.mail.Session' mail.smtp.host='localhost'/&gt;</li> <li>• Insert xml fragment into context xml</li> <li>• Start Tomcat</li> </ul>	<ul style="list-style-type: none"> <li>• Stop Tomcat</li> <li>• Find xml fragment in context xml using the name attribute</li> <li>• Delete xml fragment from context xml</li> <li>• Create xml fragment, e.g. &lt;Resource name='mail/Session' auth='Container' type='javax.mail.Session' mail.smtp.host='localhost'/&gt;</li> <li>• Insert xml fragment into context xml</li> <li>• Start Tomcat</li> </ul>	<ul style="list-style-type: none"> <li>• Stop Tomcat</li> <li>• Find xml fragment in context xml using the name attribute</li> <li>• Delete xml fragment from context xml</li> <li>• Create xml fragment, e.g. &lt;Resource name='mail/Session' auth='Container' type='javax.mail.Session' mail.smtp.host='localhost'/&gt;</li> <li>• Insert xml fragment into context xml</li> <li>• Start Tomcat</li> </ul>

## Deploying Web Applications

[Web Applications](#) can be deployed to a [Tomcat Virtual Host](#). The context root must be specified. The context root is used to create the corresponding application specific context xml in the virtual host.

## Deploying Resources

[Tomcat Resource](#) can be deployed to a [Tomcat Virtual Host](#) or [Tomcat Common Context](#). Every resource has an optional context property. This property is always set to *context* when the resource is deployed to [Tomcat Common Context](#). When the resource is deployed to a [Tomcat Virtual Host](#), the user can specify the name of the context to which the resource must be defined. When left blank, the plugin can default the value to that of the war in the current deployment. There must only be a single war in the current deployment for the defaulting to work. Example : deploying a war with context root set to *sample*, and a datasource to a virtual host will automatically result in the *context* property of the datasource been set to *sample*.

## Extension points

The Tomcat plugin is designed to be extended through Deployit's Plugin API type system. Refer to *Customization Manual* for an explanation of the type system.

## Adding additional context attributes

Tomcat context's have serveral attributes that can be set for a war. By default, the Tomcat Plugin only sets the unpackWAR and docBase attributes. When additional attributes are required, performing a type modification on the corresponding tomcat.WarModule can be done. For example, adding support for the `antiResourceLocking` attribute in `synthetic.xml`:

```
<type-modification type="tomcat.WarModule">
  <property name="antiResourceLocking" default="true" hidden="true"/>
  <property name="contextElementPropertyMapping" kind="map_string_string" hidden="true"
    default="unpackWAR:, docBase:, antiResourceLocking:"
  </type-modification>
```

An additional hidden property `antiResourceLocking` with a default value of "true" is added to the deployed. The property is added to the

`contextElementPropertyMapping` property's default value. This property is used to translate properties on the deployed to their xml equivalent in the Tomcat context. The left hand represents the property name in the deployed, while the right hand side represents the attribute name in the xml. When the right hand side is blank, the property name and xml equivalent are the same.

The same technique can be used to add attributes to other Tomcat resources like `datasource`, `queues`, etc.

## Adding new context elements

Tomcat context's have several nested components or special features (Context Parameters, Lifecycle Listeners, etc.), that can be defined in the xml.

It is possible to model these features in the Tomcat Plugin and perform subsequence deployments of these resources. For example, adding support for the Context Parameter feature.

```
<Context>
...
<Parameter name="companyName" value="My Company, Incorporated"
  override="false"/>
...
</Context>
```

Add to `synthetic.xml`:

```
<type type="tomcat.ContextParameter" extends="tomcat.ContextElement"
  deployable-type="tomcat.ContextParameterSpec">
  <generate-deployable type="tomcat.ContextParameterSpec" extends="jee.ResourceSpec"/>
  <property name="paramName" description="The name of the context parameter"/>
  <property name="value"/>
  <property name="override" kind="boolean" required="false" default="false"/>

  <!--inherited hidden -->
  <property name="elementTag" default="Parameter" hidden="true"/>
  <property name="elementName" default="paramName" hidden="true"/>
  <property name="elementPropertyMapping" kind="map_string_string" hidden="true"
    default="paramName:name, value:, override:"/>
</type>
```

## CI Reference

### Configuration Item Overview

#### Deployables

CI	Description
<a href="#">jee.DataSourceSpec</a>	Deployable Java EE DataSource accessible via JNDI
<a href="#">jee.Ear</a>	Deployable EAR artifact
<a href="#">jee.EjbJar</a>	Deployable EJB JAR artifact
<a href="#">jee.JndiResourceSpec</a>	Deployable Java EE Resource accessible via JNDI
<a href="#">jee.MailSessionSpec</a>	Deployable Java EE Mail Session accessible via JNDI
<a href="#">jee.QueueConnectionFactorySpec</a>	Deployable Java EE JMS QueueConnectionFactory accessible via JNDI
<a href="#">jee.QueueSpec</a>	Deployable Java EE JMS Queue accessible via JNDI
<a href="#">jee.ResourceSpec</a>	Description unavailable
<a href="#">jee.TopicConnectionFactorySpec</a>	Deployable Java EE JMS TopicConnectionFactory accessible via JNDI
<a href="#">jee.TopicSpec</a>	Deployable Java EE JMS Topic accessible via JNDI
<a href="#">jee.War</a>	Deployable WAR artifact
<a href="#">tomcat.ActiveMqConnectionFactorySpec</a>	ActiveMq Connection Factory installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.ActiveMqQueueSpec</a>	ActiveMq queue installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.ActiveMqTopicSpec</a>	ActiveMq topic installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.DataSourceLinkSpec</a>	To create a datasource link to a global JNDI resource
<a href="#">tomcat.DataSourceSpec</a>	DataSource installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.MailSessionSpec</a>	Mail Session installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.ResourceLinkSpec</a>	To create a link to a global JNDI resource
<a href="#">tomcat.WebsphereMqQueueConnectionFactorySpec</a>	WebsphereMq Queue Connection Factory installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.WebsphereMqQueueSpec</a>	WebsphereMq queue installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.WebsphereMqTopicConnectionFactorySpec</a>	WebsphereMq Topic Connection Factory installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.WebsphereMqTopicSpec</a>	WebsphereMq topic installed to a Tomcat Virtual Host or the Common Context (deployable)

## Deployeds



CI	Description
<a href="#">tomcat.ActiveMqConnectionFactory</a>	ActiveMq Connection Factory installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.ActiveMqQueue</a>	ActiveMq queue installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.ActiveMqTopic</a>	ActiveMq topic installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.ContextElement</a>	Base type for all Tomcat resources
<a href="#">tomcat.ContextWarModule</a>	War Module that has an associated context
<a href="#">tomcat.DataSource</a>	DataSource installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.DataSourceLink</a>	To create a datasoure link to a global JNDI resource
<a href="#">tomcat.JeeActiveMqConnectionFactory</a>	ActiveMq Connection Factory installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.JeeActiveMqQueue</a>	ActiveMq queue installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.JeeActiveMqTopic</a>	ActiveMq topic installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.JeeWebsphereMqQueue</a>	WebsphereMq queue installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.JeeWebsphereMqQueueConnectionFactory</a>	WebsphereMq Queue Connection Factory installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.JeeWebsphereMqTopic</a>	WebsphereMq topic installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.JeeWebsphereMqTopicConnectionFactory</a>	WebsphereMq Topic Connection Factory installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.JndiContextElement</a>	Resource installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.MailSession</a>	Mail Session installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.ResourceLink</a>	To create a link to a global JNDI resource
<a href="#">tomcat.WarModule</a>	War installed to a Tomcat Virtual Host
<a href="#">tomcat.WebsphereMqQueue</a>	WebsphereMq queue installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.WebsphereMqQueueConnectionFactory</a>	WebsphereMq Queue Connection Factory installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.WebsphereMqTopic</a>	WebsphereMq topic installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.WebsphereMqTopicConnectionFactory</a>	WebsphereMq Topic Connection Factory installed to a Tomcat Virtual Host or the Common Context

## Containers

CI	Description
<a href="#">tomcat.CommonContext</a>	The Context element information will be loaded by all webapps
<a href="#">tomcat.ContextContainer</a>	Models a Tomcat context on a Tomcat server in Deployit's Infrastructure
<a href="#">tomcat.Server</a>	Tomcat server on host
<a href="#">tomcat.VirtualHost</a>	represents a virtual host, which is an association of a network name for a server (such as 'www

## Other Configuration Items

CI	Description
<a href="#">jee.DataSourceSpec</a>	Deployable Java EE DataSource accessible via JNDI
<a href="#">jee.Ear</a>	Deployable EAR artifact
<a href="#">jee.EjbJar</a>	Deployable EJB JAR artifact
<a href="#">jee.JndiResourceSpec</a>	Deployable Java EE Resource accessible via JNDI
<a href="#">jee.MailSessionSpec</a>	Deployable Java EE Mail Session accessible via JNDI
<a href="#">jee.QueueConnectionFactorySpec</a>	Deployable Java EE JMS QueueConnectionFactory accessible via JNDI
<a href="#">jee.QueueSpec</a>	Deployable Java EE JMS Queue accessible via JNDI
<a href="#">jee.ResourceSpec</a>	Description unavailable
<a href="#">jee.TopicConnectionFactorySpec</a>	Deployable Java EE JMS TopicConnectionFactory accessible via JNDI
<a href="#">jee.TopicSpec</a>	Deployable Java EE JMS Topic accessible via JNDI
<a href="#">jee.War</a>	Deployable WAR artifact
<a href="#">tomcat.ActiveMqConnectionFactory</a>	ActiveMq Connection Factory installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.ActiveMqConnectionFactorySpec</a>	ActiveMq Connection Factory installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.ActiveMqQueue</a>	ActiveMq queue installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.ActiveMqQueueSpec</a>	ActiveMq queue installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.ActiveMqTopic</a>	ActiveMq topic installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.ActiveMqTopicSpec</a>	ActiveMq topic installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.CommonContext</a>	The Context element information will be loaded by all webapps
<a href="#">tomcat.ContextContainer</a>	Models a Tomcat context on a Tomcat server in Deployit's Infrastructure
<a href="#">tomcat.ContextElement</a>	Base type for all Tomcat resources
<a href="#">tomcat.ContextWarModule</a>	War Module that has an associated context
<a href="#">tomcat.DataSource</a>	DataSource installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.DataSourceLink</a>	To create a datasource link to a global JNDI resource
<a href="#">tomcat.DataSourceLinkSpec</a>	To create a datasource link to a global JNDI resource
<a href="#">tomcat.DataSourceSpec</a>	DataSource installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.JeeActiveMqConnectionFactory</a>	ActiveMq Connection Factory installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.JeeActiveMqQueue</a>	ActiveMq queue installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.JeeActiveMqTopic</a>	ActiveMq topic installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.JeeWebSphereMqQueue</a>	WebSphereMq queue installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.JeeWebSphereMqQueueConnectionFactory</a>	WebSphereMq Queue Connection Factory installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.JeeWebSphereMqTopic</a>	WebSphereMq topic installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.JeeWebSphereMqTopicConnectionFactory</a>	WebSphereMq Topic Connection Factory installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.JndiContextElement</a>	Resource installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.MailSession</a>	Mail Session installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.MailSessionSpec</a>	Mail Session installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.ResourceLink</a>	To create a link to a global JNDI resource

<a href="#">tomcat.ResourceLinkSpec</a>	To create a link to a global JNDI resource
<a href="#">tomcat.Server</a>	Tomcat server on host
<a href="#">tomcat.VirtualHost</a>	represents a virtual host, which is an association of a network name for a server (such as 'www
<a href="#">tomcat.WarModule</a>	War installed to a Tomcat Virtual Host
<a href="#">tomcat.WebsphereMqQueue</a>	WebsphereMq queue installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.WebsphereMqQueueConnectionFactory</a>	WebsphereMq Queue Connection Factory installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.WebsphereMqQueueConnectionFactorySpec</a>	WebsphereMq Queue Connection Factory installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.WebsphereMqQueueSpec</a>	WebsphereMq queue installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.WebsphereMqTopic</a>	WebsphereMq topic installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.WebsphereMqTopicConnectionFactory</a>	WebsphereMq Topic Connection Factory installed to a Tomcat Virtual Host or the Common Context
<a href="#">tomcat.WebsphereMqTopicConnectionFactorySpec</a>	WebsphereMq Topic Connection Factory installed to a Tomcat Virtual Host or the Common Context (deployable)
<a href="#">tomcat.WebsphereMqTopicSpec</a>	WebsphereMq topic installed to a Tomcat Virtual Host or the Common Context (deployable)

## Configuration Item Details

### jee.DataSourceSpec

**Type Hierarchy** [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

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

Deployable Java EE DataSource accessible via JNDI.

Public Properties	
<b>jndiName</b> :	<a href="#">STRING</a> Name used to lookup this resource in JNDI
<b>tags</b> :	<a href="#">SET_OF_STRING</a> If set, this deployable will only be mapped automatically to containers with the same tag.

### jee.Ear

**Type Hierarchy** [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](#)

Deployable EAR artifact.

**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

**jee.EjbJar**

**Type Hierarchy** `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`

Deployable EJB JAR artifact.

**Public Properties**

**checksum** : **STRING**

The checksum used to detect differences on the artifact. If not provided, it will be calculated by Deployit.

**excludeFileNamesRegex** : **STRING**

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

**placeholders** : **SET\_OF\_STRING**

Placeholders detected in this artifact

**scanPlaceholders** : **BOOLEAN** = `false`

Whether to scan this artifact for placeholders when it is imported

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

If set, this deployable will only be mapped automatically to containers with the same tag.

**Hidden Properties**

\* **textFileNamesRegex** : **STRING** = `.*\.(cfg | conf | config | ini | properties | props | txt | asp | aspx | htm | html | jsf | jsp | xht | xhtml | sql | xml | xsd | xsl | xslt)`  
Regular expression that matches file names of text files

**jee.JndiResourceSpec**

**Virtual Type**

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

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

Deployable Java EE Resource accessible via JNDI.

**Public Properties**

**jndiName** : **STRING**

Name used to lookup this resource in JNDI

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

If set, this deployable will only be mapped automatically to containers with the same tag.

**jee.MailSessionSpec**

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

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

Deployable Java EE Mail Session accessible via JNDI.

**Public Properties**

**jndiName** : **STRING**

Name used to lookup this resource in JNDI

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

If set, this deployable will only be mapped automatically to containers with the same tag.

**jee.QueueConnectionFactorySpec**

**Type Hierarchy** [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#)  
 >> [udm.BaseConfigurationItem](#)

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

Deployable Java EE JMS QueueConnectionFactory accessible via JNDI.

Public Properties	
<b>jndiName</b> :	<b>STRING</b> Name used to lookup this resource in JNDI
<b>tags</b> :	<b>SET_OF_STRING</b> If set, this deployable will only be mapped automatically to containers with the same tag.

## jee.QueueSpec

**Type Hierarchy** [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#)  
 >> [udm.BaseConfigurationItem](#)

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

Deployable Java EE JMS Queue accessible via JNDI.

Public Properties	
<b>jndiName</b> :	<b>STRING</b> Name used to lookup this resource in JNDI
<b>tags</b> :	<b>SET_OF_STRING</b> If set, this deployable will only be mapped automatically to containers with the same tag.

## jee.ResourceSpec

**Virtual Type**

**Type Hierarchy** [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

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

Description unavailable

Public Properties	
<b>tags</b> :	<b>SET_OF_STRING</b> If set, this deployable will only be mapped automatically to containers with the same tag.

## jee.TopicConnectionFactorySpec

**Type Hierarchy** [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#)  
 >> [udm.BaseConfigurationItem](#)

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

Deployable Java EE JMS TopicConnectionFactory accessible via JNDI.

Public Properties	
<b>jndiName</b> :	<b>STRING</b> Name used to lookup this resource in JNDI
<b>tags</b> :	<b>SET_OF_STRING</b> If set, this deployable will only be mapped automatically to containers with the same tag.

## jee.TopicSpec

**Type Hierarchy** [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#)  
 >> [udm.BaseConfigurationItem](#)

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

Deployable Java EE JMS Topic accessible via JNDI.

Public Properties	
<b>jndiName</b> :	<b>STRING</b> Name used to lookup this resource in JNDI
<b>tags</b> :	<b>SET_OF_STRING</b> If set, this deployable will only be mapped automatically to containers with the same tag.

**jee.War**

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

Deployable WAR artifact.

Public Properties	
<b>checksum</b> : <b>STRING</b>	The checksum used to detect differences on the artifact. If not provided, it will be calculated by Deployit.
<b>excludeFileNamesRegex</b> : <b>STRING</b>	Regular expression that matches file names that must be excluded from scanning
<b>placeholders</b> : <b>SET_OF_STRING</b>	Placeholders detected in this artifact
<b>scanPlaceholders</b> : <b>BOOLEAN</b> = <b>false</b>	Whether to scan this artifact for placeholders when it is imported
<b>tags</b> : <b>SET_OF_STRING</b>	If set, this deployable will only be mapped automatically to containers with the same tag.
Hidden Properties	
<b>* textFileNamesRegex</b> : <b>STRING</b> = <b>.\.(cfg   conf   config   ini   properties   props   txt   asp   aspx   htm   html   jsf   jsp   xht   xhtml   sql   xml   xsd   xsl   xslt)</b>	Regular expression that matches file names of text files

**tomcat.ActiveMqConnectionFactory**

<b>Type Hierarchy</b>	<a href="#">tomcat.JeeActiveMqConnectionFactory</a> >> <a href="#">tomcat.JndiContextElement</a> >> <a href="#">tomcat.ContextElement</a> >> generic.ProcessedTemplate >> generic.AbstractDeployedArtifact >> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem
<b>Interfaces</b>	udm.EmbeddedDeployedContainer, udm.Deployed, udm.ConfigurationItem

ActiveMq Connection Factory installed to a Tomcat Virtual Host or the Common Context

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

Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, description=, factory=, brokerUrl=brokerURL, brokerName=}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* factory</b> : <b>STRING</b> = org.apache.activemq.jndi.JNDIReferenceFactory	Factory
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = org.apache.activemq.ActiveMQConnectionFactory	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

**restartRequired** : **BOOLEAN** = **true**

Restart Required

**restartRequiredForNoop** : **BOOLEAN** = **false**

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## tomcat.ActiveMqConnectionFactorySpec

**Type Hierarchy** [jee.QueueConnectionFactorySpec](#) >> [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

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

ActiveMq Connection Factory installed to a Tomcat Virtual Host or the Common Context (deployable)

Public Properties	
<b>brokerName</b> : <b>STRING</b>	Broker Name (string)
<b>brokerUrl</b> : <b>STRING</b>	Broker Url (string)
<b>context</b> : <b>STRING</b>	The name of the context that this resource is associated with (string)
<b>description</b> : <b>STRING</b>	Description (string)
<b>jndiName</b> : <b>STRING</b>	Name used to lookup the resource in JNDI. (string)
<b>tags</b> : <b>SET_OF_STRING</b>	If set, this deployable will only be mapped automatically to containers with the same tag.

## tomcat.ActiveMqQueue

**Type Hierarchy** [tomcat.JeeActiveMqQueue](#) >> [tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >> [generic.ProcessedTemplate](#) >> [generic.AbstractDeployedArtifact](#) >> [generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

ActiveMq queue installed to a Tomcat Virtual Host or the Common Context

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



Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, description=, factory=, physicalName=}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* factory</b> : <b>STRING</b> = org.apache.activemq.jndi.JNDIReferenceFactory	Factory
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = org.apache.activemq.command.ActiveMQQueue	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

**restartRequired** : **BOOLEAN** = true

Restart Required

**restartRequiredForNoop** : **BOOLEAN** = false

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = true

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## tomcat.ActiveMqQueueSpec

**Type Hierarchy** [jee.QueueSpec](#) >> [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)**Interfaces** [udm.Taggable](#), [udm.Deployable](#), [udm.ConfigurationItem](#)

ActiveMq queue installed to a Tomcat Virtual Host or the Common Context (deployable)

### Public Properties

**context** : **STRING**

The name of the context that this resource is associated with (string)

**description** : **STRING**

Description (string)

**jndiName** : **STRING**

Name used to lookup the resource in JNDI. (string)

**physicalName** : **STRING**

Physical Name (string)

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

If set, this deployable will only be mapped automatically to containers with the same tag.

## tomcat.ActiveMqTopic

**Type Hierarchy** [tomcat.JeeActiveMqTopic](#) >> [tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >> [generic.ProcessedTemplate](#) >> [generic.AbstractDeployedArtifact](#) >> [generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)**Interfaces** [udm.EmbeddedDeployedContainer](#), [udm.Deployed](#), [udm.ConfigurationItem](#)

ActiveMq topic installed to a Tomcat Virtual Host or the Common Context

### Parent

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

The container on which this deployed runs.

Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, description=, factory=, physicalName=}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* factory</b> : <b>STRING</b> = org.apache.activemq.jndi.JNDIReferenceFactory	Factory
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = org.apache.activemq.command.ActiveMQTopic	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

**restartRequired** : **BOOLEAN** = **true**  
Restart Required

**restartRequiredForNoop** : **BOOLEAN** = **false**  
The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**  
Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## tomcat.ActiveMqTopicSpec

**Type Hierarchy** [jee.TopicSpec](#) >> [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

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

ActiveMq topic installed to a Tomcat Virtual Host or the Common Context (deployable)

Public Properties	
<b>context</b> : <b>STRING</b>	The name of the context that this resource is associated with (string)
<b>description</b> : <b>STRING</b>	Description (string)
<b>jndiName</b> : <b>STRING</b>	Name used to lookup the resource in JNDI. (string)
<b>physicalName</b> : <b>STRING</b>	Physical Name (string)
<b>tags</b> : <b>SET_OF_STRING</b>	If set, this deployable will only be mapped automatically to containers with the same tag.

## tomcat.CommonContext

**Type Hierarchy** [tomcat.ContextContainer](#) >> [generic.NestedContainer](#) >> [udm.BaseContainer](#) >> [udm.BaseConfigurationItem](#)

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

The Context element information will be loaded by all webapps.

Parent	
* <b>server</b> : <b>CI&lt;tomcat.Server &gt;</b>	Server
Public Properties	
<b>envVars</b> : <b>MAP_STRING_STRING</b>	Environment variables for container
<b>tags</b> : <b>SET_OF_STRING</b>	If set, only deployables with the same tag will be automatically mapped to this container.
Hidden Properties	
* <b>contextDirectory</b> : <b>STRING</b> = <b>\${container.server.home}/conf</b>	Context Directory
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

## tomcat.ContextContainer

**Virtual Type**

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

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

Models a Tomcat context on a Tomcat server in Deployit's Infrastructure.

Public Properties	
<b>* contextDirectory</b> : <b>STRING</b>	Context Directory
<b>envVars</b> : <b>MAP_STRING_STRING</b>	Environment variables for container
<b>tags</b> : <b>SET_OF_STRING</b>	If set, only deployables with the same tag will be automatically mapped to this container.
Hidden Properties	
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

## tomcat.ContextElement

**Virtual Type**

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

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

Base type for all Tomcat resources

Parent	
<b>* container</b> : <b>CI&lt;udm.Container&gt;</b>	The container on which this deployed runs.
Public Properties	
<b>context</b> : <b>STRING</b>	The name of the context that this resource is associated with
<b>deployable</b> : <b>CI&lt;udm.Deployable&gt;</b>	The deployable that this deployed is derived from.
<b>targetFile</b> : <b>STRING</b>	Name of the artifact on the generic server.

Hidden Properties	
<b>* createOptions</b> : SET_OF_STRING = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : INTEGER = 50	The order of the step in the step list for the create operation.
<b>* createVerb</b> : STRING = Create	Create Verb
<b>* destroyOptions</b> : SET_OF_STRING = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : INTEGER = 40	The order of the step in the step list for the destroy operation.
<b>* destroyVerb</b> : STRING = Destroy	Destroy Verb
<b>* modifyOptions</b> : SET_OF_STRING = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : INTEGER = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : STRING = Modify	Modify Verb
<b>* noopOptions</b> : SET_OF_STRING = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : INTEGER = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : STRING = Modify	Noop Verb
<b>* targetDirectory</b> : STRING	Path to which artifact must be copied to on the generic server.
<b>* template</b> : STRING	Classpath to the freemarker template used to generate the content of the final text based artifact.
<b>createTargetDirectory</b> : BOOLEAN = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : STRING	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : SET_OF_STRING	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartRequired</b> : BOOLEAN = false	The generic container requires a restart for the action performed by this deployed.
<b>restartRequiredForNoop</b> : BOOLEAN = false	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : BOOLEAN = true	Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## tomcat.ContextWarModule

**Virtual Type**

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

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

War Module that has an associated context.xml

Parent	
<b>* container</b> : <code>CI&lt;udm.Container&gt;</code>	The container on which this deployed runs.
Public Properties	
<b>* contextRoot</b> : <code>STRING = \${deployed.name}</code>	Context root under which the web application can be access
<b>deployable</b> : <code>CI&lt;udm.Deployable&gt;</code>	The deployable that this deployed is derived from.
<b>placeholders</b> : <code>MAP_STRING_STRING</code>	A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>
<b>targetFile</b> : <code>STRING</code>	Name of the artifact on the generic server.

Hidden Properties	
<b>* contextSubject</b> : <b>STRING</b> = war settings	Context Subject
<b>* contextTemplate</b> : <b>STRING</b>	Freemarker template used to create the context xml for the War
<b>* contextXmlTargetDirectory</b> : <b>STRING</b>	Target directory to which the context xml must be copied to on the target system
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the create operation.
<b>* createOrderOfContextXml</b> : <b>INTEGER</b> = 60	The order of the create context step in the step list for the create operation.
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	The order of the step in the step list for the destroy operation.
<b>* destroyOrderOfContextXml</b> : <b>INTEGER</b> = 40	The order of the destroy context step in the step list for the create operation.
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyOrderOfContextXml</b> : <b>INTEGER</b> = 60	The order of the rename context root step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* targetDirectory</b> : <b>STRING</b>	Path to which artifact must be copied to on the generic server.
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>deleteGeneratedResources</b> : <b>SET_OF_STRING</b>	Absolute paths to files on the target system that was generated and must be cleaned up during a destroy operation
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>preserveExistingFiles</b> : <b>BOOLEAN</b> = false	if true, preserve the existing files on the remote host (do not delete the copied files)



during the destroy operation).

**restartRequired** : **BOOLEAN** = **false**

The generic container requires a restart for the action performed by this deployed.

**restartRequiredForNoop** : **BOOLEAN** = **false**

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

**targetPathSharedSubDirectories** : **BOOLEAN** = **false**

The sub directories on the target machine are not deleted if files other than that copied by Deployit are present. Please note that setting this option to true will cause the removal process to be a bit slower.

## tomcat.DataSource

**Type Hierarchy** [tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >> [generic.ProcessedTemplate](#) >> [generic.AbstractDeployedArtifact](#) >> [generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

DataSource installed to a Tomcat Virtual Host or the Common Context

Parent	
* <b>container</b> : <a href="#">CI&lt;udm.Container&gt;</a>	The container on which this deployed runs.
Public Properties	
* <b>driverClassName</b> : <b>STRING</b>	The fully qualified Java class name of the JDBC driver to be used.
* <b>jndiName</b> : <b>STRING</b>	Name used to lookup the resource in JNDI.
* <b>password</b> : <b>STRING</b>	The connection password to be passed to our JDBC driver to establish a connection.
* <b>url</b> : <b>STRING</b>	The connection URL to be passed to our JDBC driver to establish a connection.
* <b>username</b> : <b>STRING</b>	The connection username to be passed to our JDBC driver to establish a connection.
<b>connectionProperties</b> : <b>MAP_STRING_STRING</b>	The connection properties that will be sent to our JDBC driver when establishing new connections.
<b>context</b> : <b>STRING</b>	The name of the context that this resource is associated with
<b>deployable</b> : <a href="#">CI&lt;udm.Deployable&gt;</a>	The deployable that this deployed is derived from.
<b>initialSize</b> : <b>INTEGER</b> = <b>0</b>	The initial number of connections that are created when the pool is started.
<b>maxActive</b> : <b>INTEGER</b> = <b>8</b>	The maximum number of active connections that can be allocated from this pool at the same time, or negative for no limit.
<b>maxIdle</b> : <b>INTEGER</b> = <b>8</b>	The maximum number of connections that can remain idle in the pool, without extra ones being released, or negative for no limit.
<b>maxWait</b> : <b>INTEGER</b> = <b>-1</b>	The maximum number of milliseconds that the pool will wait (when there are no available connections) for a connection to be returned before throwing an exception, or -1 to wait indefinitely.
<b>minIdle</b> : <b>INTEGER</b> = <b>0</b>	The minimum number of connections that can remain idle in the pool, without extra ones being created, or zero to create none.

Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, driverClassName=, url=, username=, password=, maxActive=, maxIdle=, minIdle=, maxWait=, initialSize=, connectionProperties=}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = javax.sql.DataSource	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartRequired</b> : <b>BOOLEAN</b> = true	

**Restart Required****restartRequiredForNoop** : **BOOLEAN** = **false**

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

**tomcat.DataSourceLink**

**Type Hierarchy** [tomcat.ResourceLink](#) >> [tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >> generic.ProcessedTemplate >> generic.AbstractDeployedArtifact >> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem

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

To create a datasoure link to a global JNDI resource. Doing a JNDI lookup on the link name will then return the linked global resource. Installed to a Tomcat Virtual Host or the Common Context

**Parent**

\* **container** : [CI<udm.Container>](#)  
The container on which this deployed runs.

**Public Properties**

\* **globalName** : **STRING**  
The name of the linked global resource in the global JNDI context.

\* **jndiName** : **STRING**  
The name of the resource link to be created, relative to the java:comp/env context.

**context** : **STRING**  
The name of the context that this resource is associated with

**deployable** : [CI<udm.Deployable>](#)  
The deployable that this deployed is derived from.

Hidden Properties	
<b>* createOptions</b> : SET_OF_STRING = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : INTEGER = 60	Create Order
<b>* createVerb</b> : STRING = Create	Create Verb
<b>* destroyOptions</b> : SET_OF_STRING = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : INTEGER = 40	Destroy Order
<b>* destroyVerb</b> : STRING = Destroy	Destroy Verb
<b>* elementName</b> : STRING = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : MAP_STRING_STRING = {jndiName=name, resourceType=type, globalName=}	Element Property Mapping
<b>* elementTag</b> : STRING = ResourceLink	Element Tag
<b>* modifyOptions</b> : SET_OF_STRING = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : INTEGER = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : STRING = Modify	Modify Verb
<b>* noopOptions</b> : SET_OF_STRING = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : INTEGER = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : STRING = Modify	Noop Verb
<b>* resourceType</b> : STRING = javax.sql.DataSource	Resource Type
<b>* targetDirectory</b> : STRING = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : STRING = \${deployed.context}.xml	Target File
<b>* template</b> : STRING = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : BOOLEAN = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : STRING	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : SET_OF_STRING	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartRequired</b> : BOOLEAN = true	Restart Required
<b>restartRequiredForNoop</b> : BOOLEAN = false	The generic container requires a restart for the NOOP action performed by this

deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## tomcat.DataSourceLinkSpec

**Type Hierarchy** [tomcat.ResourceLinkSpec](#) >> [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

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

To create a datasoure link to a global JNDI resource. Doing a JNDI lookup on the link name will then return the linked global resource. Installed to a Tomcat Virtual Host or the Common Context (deployable)

### Public Properties

**context** : **STRING**

The name of the context that this resource is associated with (string)

**globalName** : **STRING**

The name of the linked global resource in the global JNDI context. (string)

**jndiName** : **STRING**

The name of the resource link to be created, relative to the java:comp/env context. (string)

**resourceType** : **STRING**

The fully qualified Java class name expected by the web application when it performs a lookup for this resource link. (string)

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

If set, this deployable will only be mapped automatically to containers with the same tag.

## tomcat.DataSourceSpec

**Type Hierarchy** [jee.DataSourceSpec](#) >> [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

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

DataSource installed to a Tomcat Virtual Host or the Common Context (deployable)

Public Properties	
<b>connectionProperties</b> :	<b>MAP_STRING_STRING</b> The connection properties that will be sent to our JDBC driver when establishing new connections. (map_string_string)
<b>context</b> :	<b>STRING</b> The name of the context that this resource is associated with (string)
<b>driverClassName</b> :	<b>STRING</b> The fully qualified Java class name of the JDBC driver to be used. (string)
<b>initialSize</b> :	<b>STRING</b> The initial number of connections that are created when the pool is started. (integer)
<b>jndiName</b> :	<b>STRING</b> Name used to lookup the resource in JNDI. (string)
<b>maxActive</b> :	<b>STRING</b> The maximum number of active connections that can be allocated from this pool at the same time, or negative for no limit. (integer)
<b>maxIdle</b> :	<b>STRING</b> The maximum number of connections that can remain idle in the pool, without extra ones being released, or negative for no limit. (integer)
<b>maxWait</b> :	<b>STRING</b> The maximum number of milliseconds that the pool will wait (when there are no available connections) for a connection to be returned before throwing an exception, or -1 to wait indefinitely. (integer)
<b>minIdle</b> :	<b>STRING</b> The minimum number of connections that can remain idle in the pool, without extra ones being created, or zero to create none. (integer)
<b>password</b> :	<b>STRING</b> The connection password to be passed to our JDBC driver to establish a connection. (string)
<b>tags</b> :	<b>SET_OF_STRING</b> If set, this deployable will only be mapped automatically to containers with the same tag.
<b>url</b> :	<b>STRING</b> The connection URL to be passed to our JDBC driver to establish a connection. (string)
<b>username</b> :	<b>STRING</b> The connection username to be passed to our JDBC driver to establish a connection. (string)

### tomcat.JeeActiveMqConnectionFactory

<b>Type Hierarchy</b>	<a href="#">tomcat.JndiContextElement</a> >> <a href="#">tomcat.ContextElement</a> >> <a href="#">generic.ProcessedTemplate</a> >> <a href="#">generic.AbstractDeployedArtifact</a> >> <a href="#">generic.AbstractDeployed</a> >> <a href="#">udm.BaseDeployed</a> >> <a href="#">udm.BaseConfigurationItem</a>
<b>Interfaces</b>	<a href="#">udm.EmbeddedDeployedContainer</a> , <a href="#">udm.Deployed</a> , <a href="#">udm.ConfigurationItem</a>

ActiveMq Connection Factory installed to a Tomcat Virtual Host or the Common Context

Parent	
* <b>container</b> :	<b>CI&lt;udm.Container&gt;</b> The container on which this deployed runs.
Public Properties	
* <b>brokerName</b> :	<b>STRING</b> Broker Name
* <b>brokerUrl</b> :	<b>STRING</b> Broker Url
* <b>description</b> :	<b>STRING = JMS Connection Factory</b> Description
* <b>jndiName</b> :	<b>STRING</b> Name used to lookup the resource in JNDI.
<b>context</b> :	<b>STRING</b> The name of the context that this resource is associated with
<b>deployable</b> :	<b>CI&lt;udm.Deployable&gt;</b> The deployable that this deployed is derived from.

Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, description=, factory=, brokerUrl=brokerURL, brokerName=}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* factory</b> : <b>STRING</b> = org.apache.activemq.jndi.JNDIReferenceFactory	Factory
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = org.apache.activemq.ActiveMQConnectionFactory	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

**restartRequired** : **BOOLEAN** = **true**

Restart Required

**restartRequiredForNoop** : **BOOLEAN** = **false**

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

### tomcat.JeeActiveMqQueue

**Type Hierarchy** [tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >> [generic.ProcessedTemplate](#) >> [generic.AbstractDeployedArtifact](#) >> [generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

ActiveMq queue installed to a Tomcat Virtual Host or the Common Context

#### Parent

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

The container on which this deployed runs.



Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, description=, factory=, physicalName=}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* factory</b> : <b>STRING</b> = org.apache.activemq.jndi.JNDIReferenceFactory	Factory
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = org.apache.activemq.command.ActiveMQQueue	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

**restartRequired** : **BOOLEAN** = **true**  
Restart Required

**restartRequiredForNoop** : **BOOLEAN** = **false**  
The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**  
Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

### tomcat.JeeActiveMqTopic

**Type Hierarchy** [tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >>  
generic.ProcessedTemplate >> generic.AbstractDeployedArtifact >>  
generic.AbstractDeployed >> udm.BaseDeployed >>  
udm.BaseConfigurationItem

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

ActiveMq topic installed to a Tomcat Virtual Host or the Common Context

#### Parent

\* **container** : **CI**<[udm.Container](#)>  
The container on which this deployed runs.

Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, description=, factory=, physicalName=}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* factory</b> : <b>STRING</b> = org.apache.activemq.jndi.JNDIReferenceFactory	Factory
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = org.apache.activemq.command.ActiveMQTopic	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

**restartRequired** : **BOOLEAN** = **true**

Restart Required

**restartRequiredForNoop** : **BOOLEAN** = **false**

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## tomcat.JeeWebSphereMqQueue

**Type Hierarchy** [tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >> [generic.ProcessedTemplate](#) >> [generic.AbstractDeployedArtifact](#) >> [generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

WebSphereMq queue installed to a Tomcat Virtual Host or the Common Context

### Parent

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

The container on which this deployed runs.

Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, description=, factory=, physicalName=QU}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* factory</b> : <b>STRING</b> = com.ibm.mq.jms.MQQueueFactory	Factory
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = com.ibm.mq.jms.MQQueue	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

**restartRequired** : **BOOLEAN** = **true**

Restart Required

**restartRequiredForNoop** : **BOOLEAN** = **false**

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

**tomcat.JeeWebSphereMqQueueConnectionFactory**

**Type Hierarchy** [tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >>  
[generic.ProcessedTemplate](#) >> [generic.AbstractDeployedArtifact](#) >>  
[generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >>  
[udm.BaseConfigurationItem](#)

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

WebSphereMq Queue Connection Factory installed to a Tomcat Virtual Host or the Common Context

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

The container on which this deployed runs.

Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, factory=, description=, server=HOST, port=PORT, channel=CHAN, transport=TRAN, queueManager=QMGR}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* factory</b> : <b>STRING</b> = com.ibm.mq.jms.MQQueueConnectionFactoryFactory	Factory
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = com.ibm.mq.jms.MQQueueConnectionFactory	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>* transport</b> : <b>STRING</b> = 1	Transport
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	

Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

**restartRequired** : **BOOLEAN** = **true**  
Restart Required

**restartRequiredForNoop** : **BOOLEAN** = **false**  
The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**  
Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## tomcat.JeeWebSphereMqTopic

**Type Hierarchy** [tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >> [generic.ProcessedTemplate](#) >> [generic.AbstractDeployedArtifact](#) >> [generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

WebSphereMq topic installed to a Tomcat Virtual Host or the Common Context

Parent	
<b>* container</b> : <a href="#">CI&lt;udm.Container&gt;</a>	The container on which this deployed runs.
Public Properties	
<b>* description</b> : <a href="#">STRING</a>	Description
<b>* jndiName</b> : <a href="#">STRING</a>	Name used to lookup the resource in JNDI.
<b>* physicalName</b> : <a href="#">STRING</a>	Physical Name
<b>context</b> : <a href="#">STRING</a>	The name of the context that this resource is associated with
<b>deployable</b> : <a href="#">CI&lt;udm.Deployable&gt;</a>	The deployable that this deployed is derived from.



Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, description=, factory=, physicalName=TOP}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* factory</b> : <b>STRING</b> = com.ibm.mq.jms.MQTopicFactory	Factory
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = com.ibm.mq.jms.MQTopic	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

**restartRequired** : **BOOLEAN** = **true**

Restart Required

**restartRequiredForNoop** : **BOOLEAN** = **false**

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

**tomcat.JeeWebSphereMqTopicConnectionFactory**

**Type Hierarchy** [tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >>  
[generic.ProcessedTemplate](#) >> [generic.AbstractDeployedArtifact](#) >>  
[generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >>  
[udm.BaseConfigurationItem](#)

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

WebSphereMq Topic Connection Factory installed to a Tomcat Virtual Host or the Common Context

**Parent**✱ **container** : [CI](#)[udm.Container](#)

The container on which this deployed runs.

Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, factory=, description=, server=HOST, port=PORT, channel=CHAN, transport=TRAN, queueManager=QMGR}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* factory</b> : <b>STRING</b> = com.ibm.mq.jms.MQTopicConnectionFactoryFactory	Factory
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = com.ibm.mq.jms.MQTopicConnectionFactory	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>* transport</b> : <b>STRING</b> = 1	Transport
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	

Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

**restartRequired** : **BOOLEAN** = **true**

Restart Required

**restartRequiredForNoop** : **BOOLEAN** = **false**

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## tomcat.JndiContextElement

### Virtual Type

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

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

Resource installed to a Tomcat Virtual Host or the Common Context

### Parent

\* **container** : [CI<udm.Container>](#)  
The container on which this deployed runs.

### Public Properties

\* **jndiName** : **STRING**  
Name used to lookup the resource in JNDI.

**context** : **STRING**  
The name of the context that this resource is associated with

**deployable** : [CI<udm.Deployable>](#)  
The deployable that this deployed is derived from.

Hidden Properties	
<b>* createOptions</b> : SET_OF_STRING = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : INTEGER = 60	Create Order
<b>* createVerb</b> : STRING = Create	Create Verb
<b>* destroyOptions</b> : SET_OF_STRING = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : INTEGER = 40	Destroy Order
<b>* destroyVerb</b> : STRING = Destroy	Destroy Verb
<b>* elementName</b> : STRING = \${deployed.jndiName}	Element Name
<b>* elementTag</b> : STRING = Resource	Element Tag
<b>* modifyOptions</b> : SET_OF_STRING = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : INTEGER = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : STRING = Modify	Modify Verb
<b>* noopOptions</b> : SET_OF_STRING = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : INTEGER = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : STRING = Modify	Noop Verb
<b>* targetDirectory</b> : STRING = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : STRING = \${deployed.context}.xml	Target File
<b>* template</b> : STRING = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : BOOLEAN = false	Create the target directory on the generic server if it does not exist.
<b>elementPropertyMapping</b> : MAP_STRING_STRING	Mapping of the CI properties to there factory equivalent. A blank value assumes the same name as the key.
<b>inspectClasspathResources</b> : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : STRING	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : SET_OF_STRING	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script.The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartRequired</b> : BOOLEAN = true	Restart Required
<b>restartRequiredForNoop</b> : BOOLEAN = false	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : BOOLEAN = true	

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## tomcat.MailSession

**Type Hierarchy** [tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >> [generic.ProcessedTemplate](#) >> [generic.AbstractDeployedArtifact](#) >> [generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

Mail Session installed to a Tomcat Virtual Host or the Common Context

Parent	
* <b>container</b> :	<a href="#">CI</a> < <a href="#">udm.Container</a> > The container on which this deployed runs.
Public Properties	
* <b>jndiName</b> :	<a href="#">STRING</a> Name used to lookup the resource in JNDI.
* <b>smtpHost</b> :	<a href="#">STRING</a> Points at the server that provides SMTP service for your network.
<b>context</b> :	<a href="#">STRING</a> The name of the context that this resource is associated with
<b>deployable</b> :	<a href="#">CI</a> < <a href="#">udm.Deployable</a> > The deployable that this deployed is derived from.

Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, smtpHost=mail.smtp.host}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = javax.mail.Session	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script.The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartRequired</b> : <b>BOOLEAN</b> = true	Restart Required

**restartRequiredForNoop** : **BOOLEAN** = false

The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = true

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## tomcat.MailSessionSpec

**Type Hierarchy** [jee.MailSessionSpec](#) >> [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

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

Mail Session installed to a Tomcat Virtual Host or the Common Context (deployable)

### Public Properties

**context** : **STRING**

The name of the context that this resource is associated with (string)

**jndiName** : **STRING**

Name used to lookup the resource in JNDI. (string)

**smtpHost** : **STRING**

Points at the server that provides SMTP service for your network. (string)

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

If set, this deployable will only be mapped automatically to containers with the same tag.

## tomcat.ResourceLink

**Type Hierarchy** [tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >> [generic.ProcessedTemplate](#) >> [generic.AbstractDeployedArtifact](#) >> [generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

To create a link to a global JNDI resource. Doing a JNDI lookup on the link name will then return the linked global resource. Installed to a Tomcat Virtual Host or the Common Context

### Parent

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

The container on which this deployed runs.



Hidden Properties	
<b>* createOptions</b> : SET_OF_STRING = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : INTEGER = 60	Create Order
<b>* createVerb</b> : STRING = Create	Create Verb
<b>* destroyOptions</b> : SET_OF_STRING = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : INTEGER = 40	Destroy Order
<b>* destroyVerb</b> : STRING = Destroy	Destroy Verb
<b>* elementName</b> : STRING = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : MAP_STRING_STRING = {jndiName=name, resourceType=type, globalName=}	Element Property Mapping
<b>* elementTag</b> : STRING = ResourceLink	Element Tag
<b>* modifyOptions</b> : SET_OF_STRING = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : INTEGER = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : STRING = Modify	Modify Verb
<b>* noopOptions</b> : SET_OF_STRING = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : INTEGER = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : STRING = Modify	Noop Verb
<b>* targetDirectory</b> : STRING = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : STRING = \${deployed.context}.xml	Target File
<b>* template</b> : STRING = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : BOOLEAN = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : STRING	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : SET_OF_STRING	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script.The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartRequired</b> : BOOLEAN = true	Restart Required
<b>restartRequiredForNoop</b> : BOOLEAN = false	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : BOOLEAN = true	

Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## tomcat.ResourceLinkSpec

**Type Hierarchy** [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

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

To create a link to a global JNDI resource. Doing a JNDI lookup on the link name will then return the linked global resource. Installed to a Tomcat Virtual Host or the Common Context (deployable)

Public Properties	
<b>context</b> : <a href="#">STRING</a>	The name of the context that this resource is associated with (string)
<b>globalName</b> : <a href="#">STRING</a>	The name of the linked global resource in the global JNDI context. (string)
<b>jndiName</b> : <a href="#">STRING</a>	The name of the resource link to be created, relative to the java:comp/env context. (string)
<b>resourceType</b> : <a href="#">STRING</a>	The fully qualified Java class name expected by the web application when it performs a lookup for this resource link. (string)
<b>tags</b> : <a href="#">SET_OF_STRING</a>	If set, this deployable will only be mapped automatically to containers with the same tag.

## tomcat.Server

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

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

Tomcat server on host.

Parent	
* <b>host</b> : <a href="#">CI&lt;overthere.Host&gt;</a>	Host upon which the container resides
Public Properties	
* <b>home</b> : <a href="#">STRING</a>	Home directory for Tomcat
* <b>startCommand</b> : <a href="#">STRING</a>	OS specific command used to start Tomcat
* <b>startWaitTime</b> : <a href="#">INTEGER</a> = 0	Duration (in secs) to wait after the start server step has been executed
* <b>stopCommand</b> : <a href="#">STRING</a>	OS specific command used to stop Tomcat
* <b>stopWaitTime</b> : <a href="#">INTEGER</a> = 0	Duration (in secs) to wait after the stop server step has been executed
<b>envVars</b> : <a href="#">MAP_STRING_STRING</a>	Environment variables for container
<b>tags</b> : <a href="#">SET_OF_STRING</a>	If set, only deployables with the same tag will be automatically mapped to this container.

Hidden Properties			
<b>* restartOrder</b>	: INTEGER = 90		The order of the restart container step in the step list.
<b>* serverXml</b>	: STRING = \${container.home}/conf/server.xml		Server Xml
<b>* startOrder</b>	: INTEGER = 90		The order of the start container step in the step list.
<b>* startScript</b>	: STRING = tc/server/start-tc		Start Script
<b>* stopOrder</b>	: INTEGER = 10		The order of the stop container step in the step list.
<b>* stopScript</b>	: STRING = tc/server/stop-tc		Stop Script
<b>inspectClasspathResources</b>	: SET_OF_STRING		Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b>	: STRING		Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b>	: SET_OF_STRING		Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartScript</b>	: STRING		Classpath to the script used to restart the generic container.
<b>restartWaitTime</b>	: INTEGER = 0		The time to wait in seconds for a container restart action.
Control task	Parameter CI	Attributes	Description
<b>start</b>			Start the Tomcat.
<b>stop</b>			Stop the Tomcat.

## tomcat.VirtualHost

**Type Hierarchy** [tomcat.ContextContainer](#) >> generic.NestedContainer >> udm.BaseContainer >> udm.BaseConfigurationItem

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

represents a virtual host, which is an association of a network name for a server (such as 'www.mycompany.com' with the particular server on which Catalina is running.

Parent	
<b>* server</b>	: CI<tomcat.Server > Server
Public Properties	
<b>* appBase</b>	: STRING = webapps The Application Base directory for this virtual host. This is the pathname of a directory that may contain web applications to be deployed on this virtual host. You may specify a pathname that is relative to the Tomcat Server home directory.
<b>* hostName</b>	: STRING = localhost Host Name
<b>envVars</b>	: MAP_STRING_STRING Environment variables for container
<b>tags</b>	: SET_OF_STRING If set, only deployables with the same tag will be automatically mapped to this container.

Hidden Properties	
<b>* appBaseAbsolutePath</b> : <b>STRING</b> = <b>\${container.server.home}/\${container.appBase}</b>	App Base Absolute Path
<b>* contextDirectory</b> : <b>STRING</b> = <b>\${container.server.home}/conf/Catalina/\${container.hostName}</b>	Context Directory
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

## tomcat.WarModule

**Type Hierarchy** [tomcat.ContextWarModule](#) >> [generic.CopiedArtifact](#) >> [generic.AbstractDeployedArtifact](#) >> [generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

War installed to a Tomcat Virtual Host

Parent	
<b>* container</b> : <b>CI&lt;udm.Container&gt;</b>	The container on which this deployed runs.
Public Properties	
<b>* contextRoot</b> : <b>STRING</b> = <b>\${deployed.name}</b>	Context root under which the web application can be access
<b>deployable</b> : <b>CI&lt;udm.Deployable&gt;</b>	The deployable that this deployed is derived from.
<b>placeholders</b> : <b>MAP_STRING_STRING</b>	A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>

Hidden Properties	
<b>* contextElementPropertyMapping</b> : MAP_STRING_STRING = {unpackWAR=, docBase=}	Mapping of the Context attributes. A blank value assumes the same name as the key.
<b>* contextSubject</b> : STRING = war settings	Context Subject
<b>* contextTemplate</b> : STRING = tc/context/context-attributes.ftl	Context Template
<b>* contextXmlTargetDirectory</b> : STRING = \${deployed.container.contextDirectory}	Context Xml Target Directory
<b>* createOptions</b> : SET_OF_STRING = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : INTEGER = 70	Create Order
<b>* createOrderOfContextXml</b> : INTEGER = 60	The order of the create context step in the step list for the create operation.
<b>* createVerb</b> : STRING = Create	Create Verb
<b>* destroyOptions</b> : SET_OF_STRING = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : INTEGER = 30	Destroy Order
<b>* destroyOrderOfContextXml</b> : INTEGER = 40	The order of the destroy context step in the step list for the create operation.
<b>* destroyVerb</b> : STRING = Destroy	Destroy Verb
<b>* docBase</b> : STRING = \${deployed.targetFile}	Doc Base
<b>* modifyOptions</b> : SET_OF_STRING = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : INTEGER = 50	The order of the step in the step list for the modify operation.
<b>* modifyOrderOfContextXml</b> : INTEGER = 60	The order of the rename context root step in the step list for the modify operation.
<b>* modifyVerb</b> : STRING = Modify	Modify Verb
<b>* noopOptions</b> : SET_OF_STRING = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : INTEGER = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : STRING = Modify	Noop Verb
<b>* targetDirectory</b> : STRING = \${deployed.container.appBaseAbsolutePath}	Target Directory
<b>* targetFile</b> : STRING = \${deployed.contextRoot}.war	Target File
<b>createTargetDirectory</b> : BOOLEAN = false	Create the target directory on the generic server if it does not exist.
<b>deleteGeneratedResources</b> : SET_OF_STRING	Absolute paths to files on the target system that was generated and must be clean up during a destroy operation
<b>inspectClasspathResources</b> : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : STRING	

Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b> Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>preserveExistingFiles</b> : <b>BOOLEAN</b> = <b>false</b> if true, preserve the existing files on the remote host (do not delete the copied files during the destroy operation).
<b>restartRequired</b> : <b>BOOLEAN</b> = <b>true</b> Restart Required
<b>restartRequiredForNoop</b> : <b>BOOLEAN</b> = <b>false</b> The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <b>BOOLEAN</b> = <b>true</b> Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.
<b>targetPathSharedSubDirectories</b> : <b>BOOLEAN</b> = <b>false</b> The sub directories on the target machine are not deleted if files other than that copied by Deployit are present. Please note that setting this option to true will cause the removal process to be a bit slower.
<b>unpackWAR</b> : <b>BOOLEAN</b> = <b>true</b> Unpack W A R

### tomcat.WebsphereMqQueue

<b>Type Hierarchy</b>	<a href="#">tomcat.JeeWebsphereMqQueue</a> >> <a href="#">tomcat.JndiContextElement</a> >> <a href="#">tomcat.ContextElement</a> >> <a href="#">generic.ProcessedTemplate</a> >> <a href="#">generic.AbstractDeployedArtifact</a> >> <a href="#">generic.AbstractDeployed</a> >> <a href="#">udm.BaseDeployed</a> >> <a href="#">udm.BaseConfigurationItem</a>
<b>Interfaces</b>	<a href="#">udm.EmbeddedDeployedContainer</a> , <a href="#">udm.Deployed</a> , <a href="#">udm.ConfigurationItem</a>

WebsphereMq queue installed to a Tomcat Virtual Host or the Common Context

Parent
<b>* container</b> : <a href="#">CI&lt;udm.Container&gt;</a> The container on which this deployed runs.
Public Properties
<b>* description</b> : <b>STRING</b> Description
<b>* jndiName</b> : <b>STRING</b> Name used to lookup the resource in JNDI.
<b>* physicalName</b> : <b>STRING</b> Physical Name
<b>context</b> : <b>STRING</b> The name of the context that this resource is associated with
<b>deployable</b> : <a href="#">CI&lt;udm.Deployable&gt;</a> The deployable that this deployed is derived from.

Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, description=, factory=, physicalName=QU}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* factory</b> : <b>STRING</b> = com.ibm.mq.jms.MQQueueFactory	Factory
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = com.ibm.mq.jms.MQQueue	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

**restartRequired** : **BOOLEAN** = **true**  
Restart Required

**restartRequiredForNoop** : **BOOLEAN** = **false**  
The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**  
Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

### tomcat.WebsphereMqQueueConnectionFactory

**Type Hierarchy** [tomcat.JeeWebsphereMqQueueConnectionFactory](#) >> [tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >> [generic.ProcessedTemplate](#) >> [generic.AbstractDeployedArtifact](#) >> [generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

WebsphereMq Queue Connection Factory installed to a Tomcat Virtual Host or the Common Context

#### Parent

\* **container** : [CI<udm.Container>](#)  
The container on which this deployed runs.



Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, factory=, description=, server=HOST, port=PORT, channel=CHAN, transport=TRAN, queueManager=QMGR}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* factory</b> : <b>STRING</b> = com.ibm.mq.jms.MQQueueConnectionFactoryFactory	Factory
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = com.ibm.mq.jms.MQQueueConnectionFactory	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>* transport</b> : <b>STRING</b> = 1	Transport
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	

Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

**restartRequired** : **BOOLEAN** = **true**  
Restart Required

**restartRequiredForNoop** : **BOOLEAN** = **false**  
The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**  
Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## tomcat.WebsphereMqQueueConnectionFactorySpec

**Type Hierarchy** [jee.QueueConnectionFactorySpec](#) >> [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

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

WebsphereMq Queue Connection Factory installed to a Tomcat Virtual Host or the Common Context (deployable)

Public Properties	
<b>channel</b> : <b>STRING</b>	Channel (string)
<b>context</b> : <b>STRING</b>	The name of the context that this resource is associated with (string)
<b>description</b> : <b>STRING</b>	Description (string)
<b>jndiName</b> : <b>STRING</b>	Name used to lookup the resource in JNDI. (string)
<b>port</b> : <b>STRING</b>	Port (integer)
<b>queueManager</b> : <b>STRING</b>	Queue Manager (string)
<b>server</b> : <b>STRING</b>	Server (string)
<b>tags</b> : <b>SET_OF_STRING</b>	If set, this deployable will only be mapped automatically to containers with the same tag.

## tomcat.WebsphereMqQueueSpec

**Type Hierarchy** [jee.QueueSpec](#) >> [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

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

WebsphereMq queue installed to a Tomcat Virtual Host or the Common Context (deployable)

Public Properties	
<b>context</b> : <b>STRING</b>	The name of the context that this resource is associated with (string)
<b>description</b> : <b>STRING</b>	Description (string)
<b>jndiName</b> : <b>STRING</b>	Name used to lookup the resource in JNDI. (string)
<b>physicalName</b> : <b>STRING</b>	Physical Name (string)
<b>tags</b> : <b>SET_OF_STRING</b>	If set, this deployable will only be mapped automatically to containers with the same tag.

## tomcat.WebsphereMqTopic

**Type Hierarchy** [tomcat.JeeWebSphereMqTopic](#) >> [tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >> [generic.ProcessedTemplate](#) >> [generic.AbstractDeployedArtifact](#) >> [generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

WebSphereMq topic installed to a Tomcat Virtual Host or the Common Context

Parent	
* <b>container</b> :	<a href="#">CI&lt;udm.Container&gt;</a> The container on which this deployed runs.
Public Properties	
* <b>description</b> :	<a href="#">STRING</a> Description
* <b>jndiName</b> :	<a href="#">STRING</a> Name used to lookup the resource in JNDI.
* <b>physicalName</b> :	<a href="#">STRING</a> Physical Name
<b>context</b> :	<a href="#">STRING</a> The name of the context that this resource is associated with
<b>deployable</b> :	<a href="#">CI&lt;udm.Deployable&gt;</a> The deployable that this deployed is derived from.

Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, description=, factory=, physicalName=TOP}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* factory</b> : <b>STRING</b> = com.ibm.mq.jms.MQTopicFactory	Factory
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = com.ibm.mq.jms.MQTopic	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

**restartRequired** : **BOOLEAN** = **true**  
Restart Required

**restartRequiredForNoop** : **BOOLEAN** = **false**  
The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**  
Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

### tomcat.WebsphereMqTopicConnectionFactory

**Type Hierarchy** [tomcat.JeeWebsphereMqTopicConnectionFactory](#) >>  
[tomcat.JndiContextElement](#) >> [tomcat.ContextElement](#) >>  
 generic.ProcessedTemplate >> generic.AbstractDeployedArtifact >>  
 generic.AbstractDeployed >> udm.BaseDeployed >>  
 udm.BaseConfigurationItem

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

WebsphereMq Topic Connection Factory installed to a Tomcat Virtual Host or the Common Context

#### Parent

\* **container** : **CI**<[udm.Container](#)>  
The container on which this deployed runs.

Hidden Properties	
<b>* auth</b> : <b>STRING</b> = Container	Auth
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = 60	Create Order
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	Destroy Order
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<b>* elementName</b> : <b>STRING</b> = \${deployed.jndiName}	Element Name
<b>* elementPropertyMapping</b> : <b>MAP_STRING_STRING</b> = {jndiName=name, auth=, resourceType=type, factory=, description=, server=HOST, port=PORT, channel=CHAN, transport=TRAN, queueManager=QMGR}	Element Property Mapping
<b>* elementTag</b> : <b>STRING</b> = Resource	Element Tag
<b>* factory</b> : <b>STRING</b> = com.ibm.mq.jms.MQTopicConnectionFactoryFactory	Factory
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = Modify	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [none]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = Modify	Noop Verb
<b>* resourceType</b> : <b>STRING</b> = com.ibm.mq.jms.MQTopicConnectionFactory	Resource Type
<b>* targetDirectory</b> : <b>STRING</b> = \${deployed.container.contextDirectory}	Target Directory
<b>* targetFile</b> : <b>STRING</b> = \${deployed.context}.xml	Target File
<b>* template</b> : <b>STRING</b> = tc/context/context-element.ftl	Template
<b>* transport</b> : <b>STRING</b> = 1	Transport
<b>createTargetDirectory</b> : <b>BOOLEAN</b> = false	Create the target directory on the generic server if it does not exist.
<b>inspectClasspathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <b>STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <b>SET_OF_STRING</b>	

Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.

**restartRequired** : **BOOLEAN** = **true**  
Restart Required

**restartRequiredForNoop** : **BOOLEAN** = **false**  
The generic container requires a restart for the NOOP action performed by this deployed.

**targetDirectoryShared** : **BOOLEAN** = **true**  
Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

## tomcat.WebsphereMqTopicConnectionFactorySpec

**Type Hierarchy** [jee.TopicConnectionFactorySpec](#) >> [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

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

WebsphereMq Topic Connection Factory installed to a Tomcat Virtual Host or the Common Context (deployable)

Public Properties	
<b>channel</b> : <b>STRING</b>	Channel (string)
<b>context</b> : <b>STRING</b>	The name of the context that this resource is associated with (string)
<b>description</b> : <b>STRING</b>	Description (string)
<b>jndiName</b> : <b>STRING</b>	Name used to lookup the resource in JNDI. (string)
<b>port</b> : <b>STRING</b>	Port (integer)
<b>queueManager</b> : <b>STRING</b>	Queue Manager (string)
<b>server</b> : <b>STRING</b>	Server (string)
<b>tags</b> : <b>SET_OF_STRING</b>	If set, this deployable will only be mapped automatically to containers with the same tag.

## tomcat.WebsphereMqTopicSpec

**Type Hierarchy** [jee.TopicSpec](#) >> [jee.JndiResourceSpec](#) >> [jee.ResourceSpec](#) >> [udm.BaseDeployable](#) >> [udm.BaseConfigurationItem](#)

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

WebsphereMq topic installed to a Tomcat Virtual Host or the Common Context (deployable)

Public Properties	
<b>context</b> : <b>STRING</b>	The name of the context that this resource is associated with (string)
<b>description</b> : <b>STRING</b>	Description (string)
<b>jndiName</b> : <b>STRING</b>	Name used to lookup the resource in JNDI. (string)
<b>physicalName</b> : <b>STRING</b>	Physical Name (string)
<b>tags</b> : <b>SET_OF_STRING</b>	If set, this deployable will only be mapped automatically to containers with the same tag.