

# **Deployit Generic Model Plugin Manual**

**Version 3.9.1**

## Table of Contents

Table of Contents	2
Preface	3
Overview	3
Features	3
Requirements	3
Plugin Concepts	3
Container	3
NestedContainer	3
Copied Artifact	3
Executed Script	3
Manual Process	4
Executed Folder	4
Processed Template	4
Templating	4
Control Task Delegates	4
shellScript Delegate	5
shellScripts Delegate	5
Using the deployables and deployed	6
Deployable vs. Container Table	6
Deployed Actions Table	6
Sample Usage Scenario - Deploying a new middleware platform	6
Defining the Container	7
Defining a Configuration File	7
Defining a WAR	7
Defining a Datasource	7
Discovery	8
Encoding	8
Property Inspection	8
Inspecting Set properties	9
Inspecting Map properties	9
Configuration Item Discovery	9
Mail server setup	9
Configuring TLS	9
CI Reference	10
Configuration Item Overview	10
Deployables	10
Deployeds	10
Containers	10
Other Configuration Items	10
Configuration Item Details	10
generic.AbstractDeployed	11
generic.AbstractDeployedArtifact	12
generic.Archive	13
generic.BaseGenericContainer	14
generic.Container	15
generic.CopiedArtifact	16
generic.ExecutedFolder	19
generic.ExecutedScript	20
generic.ExecutedScriptWithDerivedArtifact	23
generic.File	25
generic.Folder	25
generic.ManualProcess	25
generic.NestedContainer	27
generic.ProcessedTemplate	29
generic.Resource	30
mail.SmtServer	31

## Preface

This document describes the functionality provided by the Generic Model plugin.

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

## Overview

Deployit supports a number of middleware platforms. Sometimes, though, it is necessary to extend Deployit with new middleware support. The Generic Model plugin provides a way to do this, without having to write Java code. Instead, using Deployit's flexible type system and the base CIs from the Generic Model plugin, new CIs can be defined by writing XML and providing scripts for functionality.

Several of Deployit's standard plugins are also built on top of the Generic Model plugin.

## Features

- Define custom containers
  - Stop, start, restart capabilities
- Define and copy custom artifacts to a custom container
- Define, copy and execute custom scripts and folders on a custom container
- Define resources to be processed by a template and copied to a custom container
- Define and execute control tasks on containers and deployments
- Flexible templating engine

## Requirements

The plugin requires:

- **Deployit:** version 3.5+

## Plugin Concepts

The Generic Model plugin provides several CIs that can be used as base classes for creating Deployit extensions. There are base CIs for each of Deployit's CI types (deployables, deployments and containers). A typical usage scenario is to create custom, synthetic CIs (based on one of the provided CIs) and using it to invoke the required behavior (scripts) in a deployment plan.

Note: since Deployit version 3.6, the deployments in the Generic Model Plugin can target containers that implement the [HostContainer](#) interface. In addition to the [Container](#) and derived CIs, this means they can also be targeted to CIs derived from [Host](#).

## Container

A [Container](#) is a topology CI and models middleware in your infrastructure. This would typically be used to model middleware for Deployit does not have out of the box support or that is custom to your environment. The other CIs in the plugin can be deployed to (subclasses of) the container. The behavior of the container in a deployment is configured by specifying scripts to be executed when it is started, stopped or restarted. Deployit will invoke these scripts as needed.

## NestedContainer

A [Nested Container](#) is a topology CI and models middleware in your infrastructure. The nested container allows for the modelling of abstract middleware concepts as containers to which items can be deployed.

## Copied Artifact

A [CopiedArtifact](#) is an artifact as copied over to a [Container](#). It manages the copying of any generic artifact ([File](#), [Folder](#), [Archive](#), [Resource](#)) in the deployment package to the container. It is possible to indicate that this copied artifact requires a container restart.

## Executed Script

An [ExecutedScript](#) encapsulates a script that is executed on a [Container](#). The script is

processed by the templating engine (see below) before being copied to the target container. The behavior of the script is configured by specifying scripts to be executed when it is deployed, upgraded or undeployed.

## Manual Process

A [ManualProcess](#) entails a script containing manual instructions for the operator to perform before the deployment can continue. The script is processed by the templating engine (see below) and is displayed to the operator in the step logs. Once the instructions have been carried out, the operator can continue the deployment. The instructions can also be automatically emailed.

## Executed Folder

An [ExecutedFolder](#) encapsulates a folder containing installation and rollback scripts that are executed on a [Container](#). Installation scripts are executed when the folder is deployed or updated, rollback scripts are executed when it is undeployed. Execution of the scripts happens in order. Scripts are processed by the templating engine (see below) before being copied to the target container.

## Processed Template

A [ProcessedTemplate](#) is a [Freemarker](#) template that is processed by the templating engine (see below), then copied to a [Container](#).

## Templating

When defining and using CIs with the Generic Model plugin, the need arises to use variables in certain CI properties and scripts. The most obvious use is to include properties from the deployment itself, such as the names or locations of files in the deployment package. Deployit uses the [Freemarker](#) templating engine for this.

When performing a deployment using the Generic Model plugin, all CIs and scripts are processed in Freemarker. This means that placeholders can be used in CI properties and scripts to make them more flexible. Freemarker resolves placeholders using a *context*, a set of objects defining the template's environment. This context depends on the type of CI being deployed.

For deployed CIs, the context *deployed* refers to the current CI instance. For example:

```
<type type="tc.DeployedDataSource" extends="generic.ProcessedTemplate" deployable-type="tc.DataSource"
  container-type="tc.Server">
  ...
  <property name="targetFile" default="${deployed.name}-ds.xml" hidden="true"/>
  ...
</type>
```

For container CIs, the context *container* refers to the current container instance. For example:

```
<type type="tc.Server" extends="generic.Container">
  <property name="home" default="/tmp/tomcat"/>
  <property name="targetDirectory" default="${container.home}/webapps" hidden="true"/>
</type>
```

A special case is when referring to an artifact in the placeholder. For example, when deploying a CI representing a WAR file, the following placeholder can be used to refer to that file (assuming there is a *file* property on the deployable):

```
${deployed.deployable.file}
```

In this case, Deployit will copy the referred artifact to the target container so that the file is available to the executing script. A script containing a command like the following would therefore copy the file represented by the deployable to its installation path on the remote machine:

```
cp ${deployed.deployable.file} /install/path
```

## Control Task Delegates

The Generic Model Plugin has pre defined control task delegates that have the ability to

execute scripts on a target host. The delegates can be used to define control tasks on any configuration item defined in Deployit's type system.

## shellScript Delegate

An *shellScript* delegate has the capability of executing a single script on a target host.

Arguments
<p><b>* script:</b> <b>STRING</b></p> <p>The classpath to the Freemarker template that will generate the script</p>
<p><b>host:</b> <b>STRING</b></p> <p>The target host on which to execute the script. This argument takes an expression in the form <code>\${..}</code> which indicates the property to use as the host. Example : <code>\${thisCi.parent.host}</code>, <code>\${thisCi.delegateToHost}</code>. In the absence of this argument, the delegate will try to resolve the host as follows:</p> <ul style="list-style-type: none"> <li>For <code>udm.Deployed</code> derived configuration items, the container property is used as the target host if it is an <code>overthere.HostContainer</code></li> <li>For <code>udm.Container</code> derived configuration items, the configuration item itself is used as the target host if it is an <code>overthere.HostContainer</code></li> <li>In all other cases, this argument is required.</li> </ul>
<p><b>classpathResources:</b> <b>LIST_OF_STRING</b></p> <p>Comma separated string of additional classpath resources that should be uploaded to the working directory before executing the script.</p>
<p><b>templateClasspathResources :</b> <b>LIST_OF_STRING</b></p> <p>Comma separated string of additional template classpath resources that should be uploaded to the working directory before executing the 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.</p>

Example :

```
<type type="tc.DeployedDataSource" extends="generic.ProcessedTemplate" deployable-type="tc.DataSource"
  container-type="tc.Server">
  <generate-deployable type="tc.DataSource" extends="generic.Resource"/>
  ...
  <method name="ping" delegate="shellScript" script="tc/ping.sh" classpathResources="tc/ping.py"/>
</type>
```

## shellScripts Delegate

An *shellScripts* delegate has the capability of executing multiple scripts on a target host.

Arguments
<p><b>* scripts:</b> <b>LIST_OF_STRING</b></p> <p>Comma separated string of the classpath to the Freemarker templates that will generate the scripts. In addition, each template can be prefixed with an alias. The format of the alias is <i>alias:path</i>. The alias can be used to define <code>classpathResources</code> and <code>templateClasspathResources</code> attributes that should be uploaded for the specific script. Example : <i>aliasClasspathResources</i> and <i>aliasTemplateClasspathResources</i></p>
<p><b>host:</b> <b>STRING</b></p> <p>The target host on which to execute the script. This argument takes an expression in the form <code>\${..}</code> which indicates the property to use as the host. Example : <code>\${thisCi.parent.host}</code>, <code>\${thisCi.delegateToHost}</code>. In the absence of this argument, the delegate will try to resolve the host as follows:</p> <ul style="list-style-type: none"> <li>For <code>udm.Deployed</code> derived configuration items, the container property is used as the target host if it is an <code>overthere.HostContainer</code></li> <li>For <code>udm.Container</code> derived configuration items, the configuration item itself is used as the target host if it is an <code>overthere.HostContainer</code></li> <li>In all other cases, this argument is required.</li> </ul>
<p><b>classpathResources:</b> <b>LIST_OF_STRING</b></p> <p>Comma separated string of additional classpath resources that should be uploaded to the working directory before executing the script. These resources are uploaded for all scripts.</p>
<p><b>templateClasspathResources :</b> <b>LIST_OF_STRING</b></p> <p>Comma separated string of additional template classpath resources that should be uploaded to the working directory before executing the 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. These resources are uploaded for all scripts.</p>

Example :

```

<type type="tc.Server" extends="generic.Container">
  ...
  <method name="startAndWait" delegate="shellScripts" scripts="start:tc/start.sh,tc/tailLog.sh"
startClasspathResources="tc/start.jar" startTemplateClasspathResources="tc/password.xml"
classpathResources="common.jar"/>
</type>

```

## Using the deployables and deployed

### Deployable vs. Container Table

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

Deployable	Containers	Generated deployed
generic.File generic.Archive	overthere.HostContainer	generic.CopiedArtifact
any deployable	overthere.HostContainer	generic.ExecutedScript
any deployable	overthere.HostContainer	generic.ManualProcess
any folder deployable	overthere.HostContainer	generic.ExecutedFolder
any deployable	overthere.HostContainer	generic.ProcessedTemplate

### Deployed Actions Table

The following table describes the effect a deployed has on its container.

Deployed	Create	Destroy	Modify
generic.CopiedArtifact	<ul style="list-style-type: none"> <li>Create target path on host, if needed</li> <li>Copy file to target path on host</li> </ul>	<ul style="list-style-type: none"> <li>Delete file from host (if the preserveExistingFiles property if set to false)</li> </ul>	<ul style="list-style-type: none"> <li>Delete old file from host (if the preserveExistingFiles property if set to false)</li> <li>Copy modified file to target path on host</li> </ul>
generic.ExecutedScript	<ul style="list-style-type: none"> <li>Run script through template engine</li> <li>Copy create script to container</li> <li>Execute script</li> </ul>	<ul style="list-style-type: none"> <li>Run script through template engine</li> <li>Copy destroy script to container</li> <li>Execute script</li> </ul>	<ul style="list-style-type: none"> <li>Run script through template engine</li> <li>Copy modify script to container</li> <li>Execute script</li> </ul>
generic.ManualProcess	<ul style="list-style-type: none"> <li>Run instructions script through template engine</li> <li>Mail instructions if mail properties set and target container has a mail.SMTPServer</li> <li>Display in step logs</li> <li>Pause step</li> <li>When operator retries step, step continues as successful</li> </ul>	<ul style="list-style-type: none"> <li>Same as Create</li> </ul>	<ul style="list-style-type: none"> <li>Same as Create</li> </ul>
generic.ExecutedFolder	<p>For each installation script in the folder (ordered alphabetically by name, ascending):</p> <ul style="list-style-type: none"> <li>Run script through template engine</li> <li>Copy create script to container</li> <li>Execute script</li> </ul>	<p>For each rollback script in the folder (ordered alphabetically by name, descending):</p> <ul style="list-style-type: none"> <li>Run script through template engine</li> <li>Copy destroy script to container</li> <li>Execute script</li> </ul>	<p>For each installation script in the folder that was not part of the deployment being upgraded (ordered alphabetically by name, ascending):</p> <ul style="list-style-type: none"> <li>Run script through template engine</li> <li>Copy modify script to container</li> <li>Execute script</li> </ul>
generic.ProcessedTemplate	<ul style="list-style-type: none"> <li>Run script through template engine</li> <li>Copy template to container</li> </ul>	<ul style="list-style-type: none"> <li>Run script through template engine</li> <li>Delete template from container</li> </ul>	<ul style="list-style-type: none"> <li>Run script through template engine</li> <li>Delete template from container</li> <li>Copy new template to container</li> </ul>

### Sample Usage Scenario - Deploying a new middleware platform

This section describes an example of using the Generic Model plugin to implement support for a simple middleware platform. Deployment to this platform is done by simply copying a WAR archive to the right directory on the container. Resources are created by copying configuration files into the container's configuration directory. The Tomcat application server works in a very similar manner.

By defining a container and several other CIs based on CIs from the Generic Model plugin, it is possible to add support for deploying to this platform to Deployit.

## Defining the Container

To use any of the CIs in the Generic Model plugin, they need to be targeted to a [Container](#).

This snippet shows how to define a generic container as a synthetic type:

```
<type type="tc.Server" extends="generic.Container">
  <property name="home" default="/tmp/tomcat"/>
</type>

<type type="tc.UnmanagedServer" extends="tc.Server">
  <property name="startScript" default="tc/start.sh" hidden="true"/>
  <property name="stopScript" default="tc/stop.sh" hidden="true"/>
  <property name="restartScript" default="tc/restart.sh" hidden="true"/>
</type>
```

Note that the *tc.UnmanagedServer* CI defines a start, stop and restart script. Deployit Server reads these scripts from the classpath. When targetting a deployment to the *tc.UnmanagedServer*, Deployit will include steps executing the start, stop and restart scripts in appropriate places in the deployment plan.

## Defining a Configuration File

The following snippet defines a CI based on the [CopiedArtifact](#). The *tc.DeployedFile* CI can be targeted to the *tc.Server*. The target directory is specified as a hidden property. Note the placeholder syntax used here.

```
<type type="tc.DeployedFile" extends="generic.CopiedArtifact" deployable-type="tc.File"
  container-type="tc.Server">
  <generate-deployable type="tc.File" extends="generic.File"/>
  <property name="targetDirectory" default="${deployed.container.home}/conf" hidden="true"/>
</type>
```

Using the above snippet, it is possible to create a package with a *tc.File* deployable and deploy it to an environment containing a *tc.UnmanagedServer*. This will result in a *tc.DeployedFile* deployed.

## Defining a WAR

To deploy a WAR file to the *tc.Server*, one possibility is to define a *tc.DeployedWar* CI that extends the [ExecutedScript](#). The *tc.DeployedWar* CI is generated when deploying a *jee.War* to the *tc.Server* CI. This is what the XML looks like:

```
<type type="tc.DeployedWar" extends="generic.ExecutedScript" deployable-type="jee.War"
  container-type="tc.Server">
  <generate-deployable type="tc.War" extends="jee.War"/>
  <property name="createScript" default="tc/install-war" hidden="true"/>
  <property name="modifyScript" default="tc/reinstall-war" hidden="true" required="false"/>
  <property name="destroyScript" default="tc/uninstall-war" hidden="true"/>
</type>
```

When performing an initial deployment, the create script, *tc/install-war* is executed on the target container. Inside the script, a reference to the *file* property is replaced by the actual archive. Note that the script files do not have an extension. Depending on the target platform, the extension *sh* (Unix) or *bat* (Windows) is used.

The WAR file is referenced from the script as follows:

```
echo Installing WAR ${deployed.deployable.file} in ${deployed.container.home}
```

## Defining a Datasource

Configuration files can be deployed by creating a CI based on the [ProcessedTemplate](#). By including a [Resource](#) in the package that is a Freemarker template, a configuration file can be generated during the deployment and copied to the container. This snippet defines such a CI, *tc.DeployedDataSource*:

```
<type type="tc.DeployedDataSource" extends="generic.ProcessedTemplate" deployable-type="tc.DataSource"
  container-type="tc.Server">
  <generate-deployable type="tc.DataSource" extends="generic.Resource"/>

  <property name="jdbcUrl"/>
  <property name="port" kind="integer"/>
  <property name="targetDirectory" default="${deployed.container.home}/webapps" hidden="true"/>
  <property name="targetFile" default="${deployed.name}-ds.xml" hidden="true"/>
  <property name="template" default="tc/datasource.ftl" hidden="true"/>
</type>
```

The `template` property specifies the Freemarker template file that Deployit Server reads from the classpath. The `targetDirectory` controls where the template is copied to. Inside the template, properties like `jdbcUrl` on the `datasource` can be used to produce a proper configuration file.

## Discovery

The Generic Model plugin supports discovery in any subtype of [Container](#), [Nested Container](#) and [Deployed](#). Extenders of the plugin provide shell scripts that interact with the discovery mechanism, via the standard out, with specially formatted output representing the inspected property or discovered configuration item.

```
<!-- Sample of extending Generic Mode plugin -->
<type type="sample.TomcatServer" extends="generic.Container">
  ...
  <property name="inspectScript" default="inspect/inspect-server" hidden="true"/>
</type>

<type type="sample.VirtualHost" extends="sample.NestedContainer">
  <property name="server" kind="ci" as-containment="true" referenced-type="sample.TomcatServer"/>
  ...
  <property name="inspectScript" default="inspect/inspect-virtualhost" hidden="true"/>
</type>

<type type="sample.DataSource" extends="generic.ProcessedTemplate" deployable-type="sample.DataSourceSpec"
  container-type="sample.Server">
  <generate-deployable type="sample.DataSourceSpec" extends="generic.Resource"/>
  <property name="inspectScript" default="inspect/inspect-ds" hidden="true"/>
  ...
</type>
```

## Encoding

The discovery mechanism uses URL encoding as described in [RFC3986](#) to interpret the value of an inspected property. It is the responsibility of the plugin extender to perform said encoding in the inspect shell scripts.

Sample of encoding in a BASH shell script

```
function encode()
{
  local myresult=$(printf "%b" "$1" | perl -pe's/([^\-_A-Za-z0-9])/sprintf("%%02X", ord($1))/seg')
  echo "$myresult"
}

myString='This is a string spanning many lines and with funky characters like !@#%^&*() and \\'';<>.[|]{}'
myEncodedString = $(encode "$myString")
echo $myEncodedString
```

## Property Inspection

The discovery mechanism identifies an inspected property when output with the following format is sent to the standard out.

```
INSPECTED:propertyName=value
```

The output must be prefixed with *INSPECTED*: followed by the name of the inspected property, an = sign and then the encoded value of the property.

Sample :

```
echo INSPECTED:stringField=A,value,with,commas
echo INSPECTED:intField=1999
echo INSPECTED:boolField=true
```



## Inspecting Set properties

When an inspected property is a set of strings, the value must be comma separated

```
INSPECTED:propertyName=value1,value2,value3
```

Sample :

```
echo INSPECTED:stringSetField=$(encode 'Jac,q,ues'),de,Molay
# will result in the following output
# INSPECTED:stringSetField=Jac%2Cq%2Cues,de,Molay
```

## Inspecting Map properties

When an inspected property is a map of strings, entries must be comma separated and key values must be colon separated

```
INSPECTED:propertyName=key1:value1,key2:value2,key3:value3
```

Sample :

```
echo INSPECTED:mapField=first:(encode 'Jac,q,ues:'),second:2
# will result in the following output
# INSPECTED:mapField=first:Jac%2Cq%2Cues,second:2
```

## Configuration Item Discovery

The discovery mechanism identifies a discovered configuration item when output with the following format is sent to the standard out.

```
DISCOVERED:configurationItemId=type
```

The output must be prefixed with DISCOVERED:\_, followed by the id of the configuration item as stored in the Deployit repository, an = sign and the type of the configuration item

Sample :

```
echo DISCOVERED:Infrastructure/tomcat/defaultContext=sample.VirtualHost
```

## Mail server setup

The generic plugin also adds support to Deployit for mail servers. A Mail server is a configuration item `mail.SmtServer` defined under the `/Configuration` root node. A `udm.Environment` environment configuration item can have a reference to a mail server. If it does not have a reference, a default mail server named 'defaultSmtServer' will be used to send configured mails. Using the mail server, configuration items such as the `generic.ManualProcess` can send mails notifying you of manual actions that need to be taken.

Here's a CLI snippet showing you how to create a mail server CI:

```
mailServer = factory.configurationItem("Configuration/MailServer", "mail.SmtServer")
mailServer.host = "smtp.mycompany.com"
mailServer.username = "mymailuser"
mailServer.password = "secret"
mailServer.fromAddress = "noreply@mycompany.com"
repository.create(mailServer)
```

The `mail.SmtServer` uses Java Mail to send emails. You can specify additional Java Mail properties in the `smtpProperties` attribute. Refer to the [Javadoc for JavaMail](#) for a list of all properties.

## Configuring TLS

To configure the mail server to send emails using TLS, set the following property in the SMTP properties:

```
mailServer.smtpProperties = {}
mailServer.smtpProperties["mail.smtp.starttls.enable"] = "true"
repository.update(mailServer)
```

## CI Reference

### Configuration Item Overview

#### Deployables

CI	Description
<a href="#">generic.Archive</a>	A generic, compressed binary artifact
<a href="#">generic.File</a>	A generic binary artifact
<a href="#">generic.Folder</a>	A generic folder artifact
<a href="#">generic.Resource</a>	A generic resource specification

#### Deployeds

CI	Description
<a href="#">generic.AbstractDeployed</a>	Abstract deployed that can target any deployable to a generic container
<a href="#">generic.AbstractDeployedArtifact</a>	Abstract deployed that can target any artifact to a generic container
<a href="#">generic.CopiedArtifact</a>	An artifact deployed on a generic container
<a href="#">generic.ExecutedFolder</a>	Scripts in the folder are executed against a Container based on a naming convention
<a href="#">generic.ExecutedScript</a>	A script executed on a generic container
<a href="#">generic.ExecutedScriptWithDerivedArtifact</a>	A script executed on a generic container whose deployable artifact supports placeholder replacement
<a href="#">generic.ManualProcess</a>	A manual process that needs to be performed on a generic container
<a href="#">generic.ProcessedTemplate</a>	A template deployed to a generic container

#### Containers

CI	Description
<a href="#">generic.BaseGenericContainer</a>	The supertype of all Containers
<a href="#">generic.Container</a>	A container to which generic CIs can be deployed
<a href="#">generic.NestedContainer</a>	A container that is nested with another container

#### Other Configuration Items

CI	Description
<a href="#">generic.AbstractDeployed</a>	Abstract deployed that can target any deployable to a generic container
<a href="#">generic.AbstractDeployedArtifact</a>	Abstract deployed that can target any artifact to a generic container
<a href="#">generic.Archive</a>	A generic, compressed binary artifact
<a href="#">generic.BaseGenericContainer</a>	The supertype of all Containers
<a href="#">generic.Container</a>	A container to which generic CIs can be deployed
<a href="#">generic.CopiedArtifact</a>	An artifact deployed on a generic container
<a href="#">generic.ExecutedFolder</a>	Scripts in the folder are executed against a Container based on a naming convention
<a href="#">generic.ExecutedScript</a>	A script executed on a generic container
<a href="#">generic.ExecutedScriptWithDerivedArtifact</a>	A script executed on a generic container whose deployable artifact supports placeholder replacement
<a href="#">generic.File</a>	A generic binary artifact
<a href="#">generic.Folder</a>	A generic folder artifact
<a href="#">generic.ManualProcess</a>	A manual process that needs to be performed on a generic container
<a href="#">generic.NestedContainer</a>	A container that is nested with another container
<a href="#">generic.ProcessedTemplate</a>	A template deployed to a generic container
<a href="#">generic.Resource</a>	A generic resource specification
<a href="#">mail.SmtServer</a>	SMTP Mail Server Configuration

### Configuration Item Details

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

Abstract deployed that can target any deployable to a generic container

Parent	
<b>* container</b> : <code>CI&lt;udm.Container&gt;</code>	The container on which this deployed runs.
Public Properties	
<b>deployable</b> : <code>CI&lt;udm.Deployable&gt;</code>	The deployable that this deployed is derived from.
Hidden Properties	
<b>* createOptions</b> : <code>SET_OF_STRING</code> = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <code>INTEGER</code> = 50	The order of the step in the step list for the create operation.
<b>* createVerb</b> : <code>STRING</code> = Create	Create Verb
<b>* destroyOptions</b> : <code>SET_OF_STRING</code> = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <code>INTEGER</code> = 40	The order of the step in the step list for the destroy operation.
<b>* destroyVerb</b> : <code>STRING</code> = Destroy	Destroy Verb
<b>* modifyOptions</b> : <code>SET_OF_STRING</code> = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <code>INTEGER</code> = 50	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <code>STRING</code> = Modify	Modify Verb
<b>* noopOptions</b> : <code>SET_OF_STRING</code> = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <code>INTEGER</code> = 50	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <code>STRING</code> = Modify	Noop Verb
<b>inspectClasspathResources</b> : <code>SET_OF_STRING</code>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <code>STRING</code>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <code>SET_OF_STRING</code>	Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.
<b>restartRequired</b> : <code>BOOLEAN</code> = false	The generic container requires a restart for the action performed by this deployed.
<b>restartRequiredForNoop</b> : <code>BOOLEAN</code> = false	The generic container requires a restart for the NOOP action performed by this deployed.

**generic.AbstractDeployedArtifact**

**Virtual Type**

**Type Hierarchy** [generic.AbstractDeployed](#) >> udm.BaseDeployed >>  
udm.BaseConfigurationItem

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

Abstract deployed that can target any artifact to a generic container

Parent
<div><div><div>*</div><div>container</div></div><div>: CI&lt;udm.Container&gt;</div><div>The container on which this deployed runs.</div></div>

Hidden Properties	
<b>* createOptions</b> : SET_OF_STRING = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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 = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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, uploadClasspathResources, uploadTemplateClasspathResources]	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 = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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>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.

## generic.Archive

**Virtual Type**

**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

A generic, compressed binary artifact

Public Properties	
<b>checksum</b> :	STRING The checksum used to detect differences on the artifact. If not provided, it will be calculated by Deployit.
<b>excludeFileNamesRegex</b> :	STRING Regular expression that matches file names that must be excluded from scanning
<b>placeholders</b> :	SET_OF_STRING Placeholders detected in this artifact
<b>scanPlaceholders</b> :	BOOLEAN = false Whether to scan this artifact for placeholders when it is imported
<b>tags</b> :	SET_OF_STRING If set, this deployable will only be mapped automatically to containers with the same tag.
Hidden Properties	
* <b>textFileNamesRegex</b> :	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

## generic.BaseGenericContainer

**Virtual Type**

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

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

The supertype of all Containers.

Public Properties	
<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>* restartOrder : INTEGER = 90</b>	The order of the restart container step in the step list.
<b>* startOrder : INTEGER = 90</b>	The order of the start container step in the step list.
<b>* startWaitTime : INTEGER = 0</b>	The time to wait in seconds for a container start action.
<b>* stopOrder : INTEGER = 10</b>	The order of the stop container step in the step list.
<b>* stopWaitTime : INTEGER = 0</b>	The time to wait in seconds for a container stop action.
<b>inspectClasspathResources : SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript : STRING</b>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources : 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>restartClasspathResources : SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the restart script.
<b>restartScript : STRING</b>	Classpath to the script used to restart the generic container.
<b>restartTemplateClasspathResources : SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the restart 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>restartWaitTime : INTEGER = 0</b>	The time to wait in seconds for a container restart action.
<b>startClasspathResources : SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the start script.
<b>startScript : STRING</b>	Classpath to the script used to start the generic container.
<b>startTemplateClasspathResources : SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the start 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>stopClasspathResources : SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the stop script.
<b>stopScript : STRING</b>	Classpath to the script used to stop the generic container.
<b>stopTemplateClasspathResources : SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the stop 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.

## generic.Container

### Virtual Type

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

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

A container to which generic CIs can be deployed. Start, stop and restart behavior of this container can be controlled using the corresponding script properties.

Parent
<p><b>* host</b> : <code>CI&lt;overthere&gt;.Host</code> Host upon which the container resides</p>
Public Properties
<p><b>envVars</b> : <code>MAP_STRING_STRING</code> Environment variables for container</p>
<p><b>tags</b> : <code>SET_OF_STRING</code> If set, only deployables with the same tag will be automatically mapped to this container.</p>
Hidden Properties
<p><b>* restartOrder</b> : <code>INTEGER = 90</code> The order of the restart container step in the step list.</p>
<p><b>* startOrder</b> : <code>INTEGER = 90</code> The order of the start container step in the step list.</p>
<p><b>* startWaitTime</b> : <code>INTEGER = 0</code> The time to wait in seconds for a container start action.</p>
<p><b>* stopOrder</b> : <code>INTEGER = 10</code> The order of the stop container step in the step list.</p>
<p><b>* stopWaitTime</b> : <code>INTEGER = 0</code> The time to wait in seconds for a container stop action.</p>
<p><b>inspectClasspathResources</b> : <code>SET_OF_STRING</code> Additional classpath resources that should be uploaded to the working directory before executing the inspect script.</p>
<p><b>inspectScript</b> : <code>STRING</code> Classpath to the script used to inspect the generic container.</p>
<p><b>inspectTemplateClasspathResources</b> : <code>SET_OF_STRING</code> Additional template classpath resources that should be uploaded to the working directory before executing the inspect script. The template is first rendered and the rendered content copied to a file, with the same name as the template, in the working directory.</p>
<p><b>restartClasspathResources</b> : <code>SET_OF_STRING</code> Additional classpath resources that should be uploaded to the working directory before executing the restart script.</p>
<p><b>restartScript</b> : <code>STRING</code> Classpath to the script used to restart the generic container.</p>
<p><b>restartTemplateClasspathResources</b> : <code>SET_OF_STRING</code> Additional template classpath resources that should be uploaded to the working directory before executing the restart 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.</p>
<p><b>restartWaitTime</b> : <code>INTEGER = 0</code> The time to wait in seconds for a container restart action.</p>
<p><b>startClasspathResources</b> : <code>SET_OF_STRING</code> Additional classpath resources that should be uploaded to the working directory before executing the start script.</p>
<p><b>startScript</b> : <code>STRING</code> Classpath to the script used to start the generic container.</p>
<p><b>startTemplateClasspathResources</b> : <code>SET_OF_STRING</code> Additional template classpath resources that should be uploaded to the working directory before executing the start 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.</p>
<p><b>stopClasspathResources</b> : <code>SET_OF_STRING</code> Additional classpath resources that should be uploaded to the working directory before executing the stop script.</p>
<p><b>stopScript</b> : <code>STRING</code> Classpath to the script used to stop the generic container.</p>
<p><b>stopTemplateClasspathResources</b> : <code>SET_OF_STRING</code> Additional template classpath resources that should be uploaded to the working directory before executing the stop 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.</p>

## generic.CopiedArtifact




**Virtual Type**

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

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

An artifact deployed on a generic container

Parent	
	<b>container</b> : <a href="#">CI&lt;udm.Container&gt;</a> The container on which this deployed runs.
Public Properties	
	<b>deployable</b> : <a href="#">CI&lt;udm.Deployable&gt;</a> The deployable that this deployed is derived from.
	<b>placeholders</b> : <a href="#">MAP_STRING_STRING</a> A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>
	<b>targetFile</b> : <a href="#">STRING</a> Name of the artifact on the generic server.

Hidden Properties	
<b>* createOptions</b> : SET_OF_STRING = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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 = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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, uploadClasspathResources, uploadTemplateClasspathResources]	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 = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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>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>preserveExistingFiles</b> : BOOLEAN = false	if true, preserve the existing files on the remote host (do not delete the copied files during the destroy operation).
<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.
<b>targetPathSharedSubDirectories</b> : 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.


## generic.ExecutedFolder

**Virtual Type**

**Type Hierarchy** [generic.AbstractDeployed](#) >> udm.BaseDeployed >>  
udm.BaseConfigurationItem

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

Scripts in the folder are executed against a Container based on a naming convention


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

Hidden Properties	
<b>* commonScriptFolderName</b> : <b>STRING</b> = <b>common</b>	Common folder that should be uploaded to the working directory.
<b>* createOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	Options for the create step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* createOrder</b> : <b>INTEGER</b> = <b>50</b>	The order of the step in the step list for the create operation.
<b>* createVerb</b> : <b>STRING</b> = <b>Create</b>	Create Verb
<b>* destroyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	Options for the destroy step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* destroyOrder</b> : <b>INTEGER</b> = <b>40</b>	The order of the step in the step list for the destroy operation.
<b>* destroyVerb</b> : <b>STRING</b> = <b>Destroy</b>	Destroy Verb
<b>* modifyOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	Options for the modify step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* modifyOrder</b> : <b>INTEGER</b> = <b>50</b>	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <b>STRING</b> = <b>Modify</b>	Modify Verb
<b>* noopOptions</b> : <b>SET_OF_STRING</b> = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	Options for the noop step (1 or more of: none,uploadArtifactData,uploadClasspathResources,uploadTemplateClasspathResources).
<b>* noopOrder</b> : <b>INTEGER</b> = <b>50</b>	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <b>STRING</b> = <b>Modify</b>	Noop Verb
<b>classpathResources</b> : <b>SET_OF_STRING</b>	Additional classpath resources that should be uploaded to the working directory before executing the script.
<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> = <b>false</b>	The generic container requires a restart for the action performed by this deployed.
<b>restartRequiredForNoop</b> : <b>BOOLEAN</b> = <b>false</b>	The generic container requires a restart for the NOOP action performed by this deployed.
<b>templateClasspathResources</b> : <b>SET_OF_STRING</b>	Additional template classpath resources that should be uploaded to the working directory before executing the 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.

## generic.ExecutedScript

**Virtual Type**  
**Type Hierarchy** [generic.AbstractDeployed](#) >> udm.BaseDeployed >>  
udm.BaseConfigurationItem  
**Interfaces** udm.EmbeddedDeployedContainer, udm.Deployed,  
udm.ConfigurationItem

A script executed on a generic container

Parent	
	<b>container</b> : <a href="#">CI&lt;udm.Container&gt;</a> The container on which this deployed runs.
Public Properties	
	<b>deployable</b> : <a href="#">CI&lt;udm.Deployable&gt;</a> The deployable that this deployed is derived from.

Hidden Properties	
<b>* createOptions</b> : SET_OF_STRING = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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>* createScript</b> : STRING	Classpath to the script that is uploaded and executed on the generic container for the create operation.
<b>* createVerb</b> : STRING = Create	Create Verb
<b>* destroyOptions</b> : SET_OF_STRING = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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, uploadClasspathResources, uploadTemplateClasspathResources]	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 = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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>classpathResources</b> : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the script.
<b>destroyScript</b> : STRING	Classpath to the script that is uploaded and executed on the generic container for the 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> : 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>modifyScript</b> : STRING	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
<b>noopScript</b> : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
<b>remoteWorkingDirectoryPath</b> : STRING	Name of working directory on target host. Default is to create a temporary directory which is deleted when connection is closed.
<b>restartRequired</b> : 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.

**retainRemoteWorkingDirectory** : **BOOLEAN** = false

Retain the specified working directory on target host after completion.

**templateClasspathResources** : **SET\_OF\_STRING**

Additional template classpath resources that should be uploaded to the working directory before executing the 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.

## generic.ExecutedScriptWithDerivedArtifact

### Virtual Type

**Type Hierarchy** [generic.ExecutedScript](#) >> [generic.AbstractDeployed](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

A script executed on a generic container whose deployable artifact supports placeholder replacement

### Parent

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

### Public Properties

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

**placeholders** : [MAP\\_STRING\\_STRING](#)  
A key/value pair mapping of placeholders in the deployed artifact to their values. Special values are and

Hidden Properties	
<b>* createOptions</b> : SET_OF_STRING = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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>* createScript</b> : STRING	Classpath to the script that is uploaded and executed on the generic container for the create operation.
<b>* createVerb</b> : STRING = Create	Create Verb
<b>* destroyOptions</b> : SET_OF_STRING = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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, uploadClasspathResources, uploadTemplateClasspathResources]	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 = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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>classpathResources</b> : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the script.
<b>destroyScript</b> : STRING	Classpath to the script that is uploaded and executed on the generic container for the 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> : 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>modifyScript</b> : STRING	Classpath to the script that is uploaded and executed on the generic container for the modify operation.
<b>noopScript</b> : STRING	Classpath to the script that is uploaded and executed on the generic container for the noop operation.
<b>remoteWorkingDirectoryPath</b> : STRING	Name of working directory on target host. Default is to create a temporary directory which is deleted when connection is closed.
<b>restartRequired</b> : 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.

**retainRemoteWorkingDirectory** : **BOOLEAN** = false

Retain the specified working directory on target host after completion.

**templateClasspathResources** : **SET\_OF\_STRING**

Additional template classpath resources that should be uploaded to the working directory before executing the 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.

**generic.File****Virtual Type**

**Type Hierarchy** udm.BaseDeployableFileArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

A generic binary 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** = true

Whether to scan this artifact for placeholders when it is imported

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

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

**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

**generic.Folder****Virtual Type**

**Type Hierarchy** udm.BaseDeployableFolderArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem

**Interfaces** udm.Taggable, udm.Deployable, udm.SourceArtifact, udm.Artifact, udm.DeployableArtifact, udm.ConfigurationItem, udm.FolderArtifact

A generic folder 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** = true

Whether to scan this artifact for placeholders when it is imported

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

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

**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


**generic.ManualProcess**

**Virtual Type**

**Type Hierarchy** [generic.AbstractDeployed](#) >> udm.BaseDeployed >>  
udm.BaseConfigurationItem

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

A manual process that needs to be performed on a generic container

Parent	
	<b>container</b> : <a href="#">CI&lt;udm.Container&gt;</a> The container on which this deployed runs.
Public Properties	
	<b>deployable</b> : <a href="#">CI&lt;udm.Deployable&gt;</a> The deployable that this deployed is derived from.
	<b>fromAddress</b> : <a href="#">STRING</a> From mail address. Defaults to SMTPServer fromAddress.
	<b>subject</b> : <a href="#">STRING</a> Mail subject
	<b>toAddresses</b> : <a href="#">LIST_OF_STRING</a> Mail addresses of recipients.

Hidden Properties	
<b>* createOptions</b> : SET_OF_STRING = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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>* createScript</b> : STRING	Classpath to the script that generates the instructions for the create operation.
<b>* createVerb</b> : STRING = Create	Create Verb
<b>* destroyOptions</b> : SET_OF_STRING = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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, uploadClasspathResources, uploadTemplateClasspathResources]	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 = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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>destroyScript</b> : STRING	Classpath to the script that generates the instructions for the 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> : 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>modifyScript</b> : STRING	Classpath to the script that generates the instructions for the modify operation.
<b>noopScript</b> : STRING	Classpath to the script that generates the instructions for the noop operation.
<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.

## generic.NestedContainer

**Virtual Type**

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

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

A container that is nested with another container

Public Properties	
<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> : <a href="#">INTEGER</a> = 90	The order of the restart container step in the step list.
* <b>startOrder</b> : <a href="#">INTEGER</a> = 90	The order of the start container step in the step list.
* <b>startWaitTime</b> : <a href="#">INTEGER</a> = 0	The time to wait in seconds for a container start action.
* <b>stopOrder</b> : <a href="#">INTEGER</a> = 10	The order of the stop container step in the step list.
* <b>stopWaitTime</b> : <a href="#">INTEGER</a> = 0	The time to wait in seconds for a container stop action.
<b>inspectClasspathResources</b> : <a href="#">SET_OF_STRING</a>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
<b>inspectScript</b> : <a href="#">STRING</a>	Classpath to the script used to inspect the generic container.
<b>inspectTemplateClasspathResources</b> : <a href="#">SET_OF_STRING</a>	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>restartClasspathResources</b> : <a href="#">SET_OF_STRING</a>	Additional classpath resources that should be uploaded to the working directory before executing the restart script.
<b>restartScript</b> : <a href="#">STRING</a>	Classpath to the script used to restart the generic container.
<b>restartTemplateClasspathResources</b> : <a href="#">SET_OF_STRING</a>	Additional template classpath resources that should be uploaded to the working directory before executing the restart 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>restartWaitTime</b> : <a href="#">INTEGER</a> = 0	The time to wait in seconds for a container restart action.
<b>startClasspathResources</b> : <a href="#">SET_OF_STRING</a>	Additional classpath resources that should be uploaded to the working directory before executing the start script.
<b>startScript</b> : <a href="#">STRING</a>	Classpath to the script used to start the generic container.
<b>startTemplateClasspathResources</b> : <a href="#">SET_OF_STRING</a>	Additional template classpath resources that should be uploaded to the working directory before executing the start 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>stopClasspathResources</b> : <a href="#">SET_OF_STRING</a>	Additional classpath resources that should be uploaded to the working directory before executing the stop script.
<b>stopScript</b> : <a href="#">STRING</a>	Classpath to the script used to stop the generic container.
<b>stopTemplateClasspathResources</b> : <a href="#">SET_OF_STRING</a>	Additional template classpath resources that should be uploaded to the working directory before executing the stop 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.

## generic.ProcessedTemplate

### Virtual Type

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

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

A template deployed to a generic container

Parent
<p><b>* container :</b> <code>Cl&lt;udm.Container&gt;</code> The container on which this deployed runs.</p>

Hidden Properties	
* <b>createOptions</b> : SET_OF_STRING = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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 = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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, uploadClasspathResources, uploadTemplateClasspathResources]	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 = [uploadArtifactData, uploadClasspathResources, uploadTemplateClasspathResources]	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.

## generic.Resource

**Virtual Type****Type Hierarchy** udm.BaseDeployable >> udm.BaseConfigurationItem**Interfaces** udm.Taggable, udm.Deployable, udm.ConfigurationItem

A generic resource specification

**Public Properties****tags** : SET\_OF\_STRING

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

**mail.SmtpServer****Interfaces** udm.ConfigurationItem

SMTP Mail Server Configuration

**Public Properties****\* fromAddress** : STRING

Default from address to use for messages sent with this server.

**\* host** : STRING

SMTP host

**\* port** : INTEGER = 25

SMTP port

**password** : STRING

Password to authenticate with host

**smtpProperties** : MAP\_STRING\_STRINGRefer to <http://javamail.kenai.com/nonav/javadocs/com/sun/mail/smtp/package-summary.html> for all properties that can be used.**testAddress** : STRING

The address to which a test mail is sent when using the 'Send Test Mail' control task.

**username** : STRING

Username to authenticate with host

Control task	Parameter CI	Attributes	Description
<b>sendTestMail</b>			no description