

# Deployit Webserver Plugin Manual

Version 3.8.1

## Table of Contents

Table of Contents	2
Preface	3
Overview	3
Features	3
Requirements	3
Usage in Deployment Packages	3
Using the deployables and deployed	3
Deployable vs. Container table	3
Deployed Actions Table	4
CI Reference	4
Configuration Item Overview	4
Deployables	4
Deployeds	4
Containers	4
Other Configuration Items	4
Configuration Item Details	4
www.ApacheConfFragment	4
www.ApacheConfFragmentSpec	5
www.ApacheHttpdServer	6
www.ApacheVirtualHost	7
www.ApacheVirtualHostSpec	8
www.PublishedWebContent	9
www.WebContent	10

## Preface

This document describes the functionality provided by the Webserver plugin.

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

## Overview

The webserver-plugin is a Deployit plugin that supports deployment of web content and webserver configuration to a webserver.

## Features

- Runs on Deployit 3.6 and up.
- Supports deployment to Apache and IHS web servers.
- Deploys and undeploys webserver artifacts:
  - Web content (HTML pages, images, etc.)
  - Virtual host configuration
  - Any configuration fragment
- Supports starting, stopping and restarting of web servers as control tasks.

## Requirements

- **Deployit requirements**
  - **Deployit:** version 3.6+
  - **Other Deployit Plugins:** None
- **Infrastructural requirements**
  - **User credentials** for accessing the webserver executables on the host running the webserver.

## Usage in Deployment Packages

The following is a manifest snippet that shows how web content and a virtual host can be included in a deployment package. The web content CI refers to a folder, *html*, included in the deployment package.

```
Manifest-Version: 1.0
Deployit-Package-Format-Version: 1.3
CI-Application: PetClinic-ear
CI-Version: 2.0

Name: PetClinic-2.0.ear
CI-Type: jee.Ear
CI-Name: PetClinic

Name: html
CI-Type: www.WebContent
CI-Name: PetClinic-html

Name: PetClinic-vh
CI-Type: www.ApacheVirtualHostSpec
CI-host: *
CI-port: 8080
```

## Using the deployables and deployed

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

### Deployable vs. Container table

Deployable	Container	Generated deployed
www.WebContent	www.ApacheHttpdServer	www.PublishedWebContent
www.ApacheVirtualHostSpec	www.ApacheHttpdServer	www.ApacheVirtualHost
www.ApacheConfFragmentSpec	www.ApacheHttpdServer	www.ApacheConfFragment

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

## Deployed Actions Table

Deployed	Actions performed for operations		
	Create	Destroy	Modify
<a href="#">www.PublishedWebContent</a>	<ul style="list-style-type: none"> <li>Create target path on webserver, if needed</li> <li>Copy web content to target path on webserver</li> </ul>	<ul style="list-style-type: none"> <li>Delete web content from host</li> </ul>	<ul style="list-style-type: none"> <li>Delete existing web content from webserver</li> <li>Copy modified web content to target path on webserver</li> </ul>
<a href="#">www.ApacheVirtualHost</a>	<ul style="list-style-type: none"> <li>Copy virtual host configuration to webserver</li> </ul>	<ul style="list-style-type: none"> <li>Delete virtual host configuration from webserver</li> </ul>	<ul style="list-style-type: none"> <li>Delete virtual host configuration from webserver</li> <li>Copy new virtual host configuration to webserver</li> </ul>
<a href="#">www.ApacheConfFragment</a>	<ul style="list-style-type: none"> <li>Copy configuration fragment to webserver</li> </ul>	<ul style="list-style-type: none"> <li>Delete configuration fragment from webserver</li> </ul>	<ul style="list-style-type: none"> <li>Delete configuration fragment from webserver</li> <li>Copy new configuration fragment to webserver</li> </ul>

## CI Reference

### Configuration Item Overview

#### Deployables

CI	Description
<a href="#">www.ApacheConfFragmentSpec</a>	Specification for an Apache configuration fragment
<a href="#">www.ApacheVirtualHostSpec</a>	Specification for an Apache virtual host
<a href="#">www.WebContent</a>	Folder containing web content (HTML pages, images, etc)

#### Deployeds

CI	Description
<a href="#">www.ApacheConfFragment</a>	Apache configuration fragment
<a href="#">www.ApacheVirtualHost</a>	Apache virtual host
<a href="#">www.PublishedWebContent</a>	Web content on the webserver

#### Containers

CI	Description
<a href="#">www.ApacheHttpdServer</a>	Apache HTTPD web server

#### Other Configuration Items

CI	Description
<a href="#">www.ApacheConfFragment</a>	Apache configuration fragment
<a href="#">www.ApacheConfFragmentSpec</a>	Specification for an Apache configuration fragment
<a href="#">www.ApacheHttpdServer</a>	Apache HTTPD web server
<a href="#">www.ApacheVirtualHost</a>	Apache virtual host
<a href="#">www.ApacheVirtualHostSpec</a>	Specification for an Apache virtual host
<a href="#">www.PublishedWebContent</a>	Web content on the webserver
<a href="#">www.WebContent</a>	Folder containing web content (HTML pages, images, etc)

### Configuration Item Details

#### [www.ApacheConfFragment](#)

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

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

Apache configuration fragment

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>* createOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the create operation.
<b>* createVerb</b> : <code>STRING = Create</code>	Create Verb
<b>* destroyOrder</b> : <code>INTEGER = 40</code>	The order of the step in the step list for the destroy operation.
<b>* destroyVerb</b> : <code>STRING = Destroy</code>	Destroy Verb
<b>* modifyOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the modify operation.
<b>* modifyVerb</b> : <code>STRING = Modify</code>	Modify Verb
<b>* noopOrder</b> : <code>INTEGER = 50</code>	The order of the step in the step list for the noop operation.
<b>* noopVerb</b> : <code>STRING = Modify</code>	Noop Verb
<b>* targetDirectory</b> : <code>STRING = \${deployed.container.configurationFragmentDirectory}</code>	Target directory name
<b>* targetFile</b> : <code>STRING = \${deployed.name}.conf</code>	Target file name
<b>* template</b> : <code>STRING = www/apache/\${deployed.type}.conf.ftl</code>	Configuration fragment template file name.
<b>createTargetDirectory</b> : <code>BOOLEAN = false</code>	Create the target directory on the generic server if it does not exist.
<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 = true</code>	Restart Required
<b>restartRequiredForNoop</b> : <code>BOOLEAN = false</code>	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <code>BOOLEAN = true</code>	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.

## www.ApacheConfFragmentSpec

**Type Hierarchy** generic.Resource >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification for an Apache configuration fragment.

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

**www.ApacheHttpdServer**

**Type Hierarchy** generic.Container >> udm.BaseContainer >> udm.BaseConfigurationItem  
**Interfaces** udm.Taggable, udm.ConfigurationItem, generic.GenericContainer, udm.Container, overthere.HostContainer

Apache HTTPD web server

Parent	
* <b>host</b> : <code>CI&lt;overthere.Host&gt;</code>	Host upon which the container resides
Public Properties	
* <b>configurationFragmentDirectory</b> : <code>STRING</code>	The directory into which Apache configuration files (for virtual host definitions) should be written
* <b>defaultDocumentRoot</b> : <code>STRING</code>	The directory in which document roots for virtual hosts should be created. Web content deployed without a virtual host will also be copied into this directory
* <b>restartWaitTime</b> : <code>INTEGER = 10</code>	Duration (in secs) to wait after the restart server step has been executed
* <b>startCommand</b> : <code>STRING</code>	The command to start Apache. Arguments containing spaces are not supported
* <b>startWaitTime</b> : <code>INTEGER = 10</code>	Duration (in secs) to wait after the start server step has been executed
* <b>stopCommand</b> : <code>STRING</code>	The command to stop Apache. Arguments containing spaces are not supported
* <b>stopWaitTime</b> : <code>INTEGER = 10</code>	Duration (in secs) to wait after the stop server step has been executed
<b>envVars</b> : <code>MAP_STRING_STRING</code>	Environment variables for container
<b>restartCommand</b> : <code>STRING</code>	The command to restart Apache. If left blank, reload command is used. If reload command is also blank than the sequence 'stop, start' is used. Arguments containing spaces are not supported
<b>tags</b> : <code>SET_OF_STRING</code>	If set, only deployables with the same tag will be automatically mapped to this container.
Hidden Properties	
* <b>restartOrder</b> : <code>INTEGER = 90</code>	The order of the restart container step in the step list.
* <b>restartScript</b> : <code>STRING = &lt;#assign restart=(container.restartCommand)!"&gt;&lt;#if restart != "&gt;www/apache/restart&lt;#else&gt;www/apache/stopstart</code>	The command to restart or reload Apache. If left blank, the sequence 'stop, start' is used. Arguments containing spaces are not supported
* <b>startOrder</b> : <code>INTEGER = 90</code>	The order of the start container step in the step list.
* <b>startScript</b> : <code>STRING = www/apache/start</code>	The command to start Apache. Arguments containing spaces are not supported
* <b>stopOrder</b> : <code>INTEGER = 10</code>	The order of the stop container step in the step list.
* <b>stopScript</b> : <code>STRING = www/apache/stop</code>	The command to stop Apache. Arguments containing spaces are not supported
<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.

Control Tasks		
	<b>restart</b>	Restart Apache
	<b>start</b>	Start Apache
	<b>stop</b>	Stop Apache

## www.ApacheVirtualHost

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

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

Apache virtual host

Parent	
* <b>container</b> :	CI<udm.Container> The container on which this deployed runs.
Public Properties	
* <b>host</b> :	STRING The virtual host name. Use "*" to match all hosts
* <b>port</b> :	STRING The virtual host port. Use "*" to match all ports
<b>deployable</b> :	CI<udm.Deployable> The deployable that this deployed is derived from.
<b>documentRoot</b> :	STRING The document root for web content using this virtual host. If left blank, a directory is created in the web server's document root

Hidden Properties	
<b>* createOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the create operation.
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	The order of the step in the step list for the destroy operation.
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<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>* 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> = \${deployed.container.configurationFragmentDirectory}	Target directory name
<b>* targetFile</b> : <b>STRING</b> = \${deployed.name}.conf	Target file name
<b>* template</b> : <b>STRING</b> = www/apache/\${deployed.type}.conf.ftl	Configuration fragment template file name.
<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
<b>restartRequiredForNoop</b> : <b>BOOLEAN</b> = false	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <b>BOOLEAN</b> = 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.

## www.ApacheVirtualHostSpec

**Type Hierarchy** [www.ApacheConfFragmentSpec](#) >> generic.Resource >> udm.BaseDeployable >> udm.BaseConfigurationItem

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

Specification for an Apache virtual host

Public Properties	
<b>documentRoot</b> : <b>STRING</b>	The document root for web content using this virtual host. If left blank, a directory is created in the web server's document root (string)
<b>host</b> : <b>STRING</b>	The virtual host name. Use '*' to match all hosts (string)
<b>port</b> : <b>STRING</b>	The virtual host port. Use '*' to match all ports (string)
<b>tags</b> : <b>SET_OF_STRING</b>	If set, this deployable will only be mapped automatically to containers with the same tag.



**www.PublishedWebContent**

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

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

### Web content on the webserver

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

Hidden Properties	
<b>* createOrder</b> : <b>INTEGER</b> = 50	The order of the step in the step list for the create operation.
<b>* createVerb</b> : <b>STRING</b> = Create	Create Verb
<b>* destroyOrder</b> : <b>INTEGER</b> = 40	The order of the step in the step list for the destroy operation.
<b>* destroyVerb</b> : <b>STRING</b> = Destroy	Destroy Verb
<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>* 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> = <#assign target=(deployed.virtualHost.documentRoot)!"/><#if target != ">\${target}<#else>\${deployed.container.defaultDocumentRoot}</#if>	Target directory name
<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>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).
<b>restartRequired</b> : <b>BOOLEAN</b> = false	Restart Required
<b>restartRequiredForNoop</b> : <b>BOOLEAN</b> = false	The generic container requires a restart for the NOOP action performed by this deployed.
<b>targetDirectoryShared</b> : <b>BOOLEAN</b> = 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>targetFile</b> : <b>STRING</b>	Target File

## www.WebContent

<b>Type Hierarchy</b>	generic.Folder >> udm.BaseDeployableFolderArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem
<b>Interfaces</b>	udm.Taggable, udm.Deployable, udm.SourceArtifact, udm.Artifact, udm.DeployableArtifact, udm.ConfigurationItem, udm.FolderArtifact

Folder containing web content (HTML pages, images, etc)

Public Properties	
<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> = true	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> = .+\. (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