

Deployit Webserver Plugin Manual

Version 3.8.5

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.ApacheConfFile	4
www.ApacheConfFragment	5
www.ApacheConfFragmentSpec	6
www.ApacheHttpdServer	6
www.ApacheVirtualHost	7
www.ApacheVirtualHostSpec	9
www.DeployedApacheConfFile	9
www.PublishedWebContent	10
www.WebContent	11

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 web server executables on the host running the web server.

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
www.PublishedWebContent	<ul style="list-style-type: none"> Create target path on webserver, if needed Copy web content to target path on webserver 	<ul style="list-style-type: none"> Delete web content from host 	<ul style="list-style-type: none"> Delete existing web content from webserver Copy modified web content to target path on webserver
www.ApacheVirtualHost	<ul style="list-style-type: none"> Copy virtual host configuration to webserver 	<ul style="list-style-type: none"> Delete virtual host configuration from webserver 	<ul style="list-style-type: none"> Delete virtual host configuration from webserver Copy new virtual host configuration to webserver
www.ApacheConfFragment	<ul style="list-style-type: none"> Copy configuration fragment to webserver 	<ul style="list-style-type: none"> Delete configuration fragment from webserver 	<ul style="list-style-type: none"> Delete configuration fragment from webserver Copy new configuration fragment to webserver

CI Reference

Configuration Item Overview

Deployables

CI	Description
www.ApacheConfFile	Apache configuration file
www.ApacheConfFragmentSpec	Specification for an Apache configuration fragment
www.ApacheVirtualHostSpec	Specification for an Apache virtual host
www.WebContent	Folder containing web content (HTML pages, images, etc)

Deployeds

CI	Description
www.ApacheConfFragment	Deployed generated Apache configuration file
www.ApacheVirtualHost	Apache virtual host
www.DeployedApacheConfFile	Deployed Apache configuration file
www.PublishedWebContent	Web content on the webserver

Containers

CI	Description
www.ApacheHttpdServer	Apache HTTPD web server

Other Configuration Items

CI	Description
www.ApacheConfFile	Apache configuration file
www.ApacheConfFragment	Deployed generated Apache configuration file
www.ApacheConfFragmentSpec	Specification for an Apache configuration fragment
www.ApacheHttpdServer	Apache HTTPD web server
www.ApacheVirtualHost	Apache virtual host
www.ApacheVirtualHostSpec	Specification for an Apache virtual host
www.DeployedApacheConfFile	Deployed Apache configuration file
www.PublishedWebContent	Web content on the webserver
www.WebContent	Folder containing web content (HTML pages, images, etc)

Configuration Item Details

[www.ApacheConfFile](#)

Type Hierarchy	generic.File >> udm.BaseDeployableFileArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem
Interfaces	udm.Taggable, udm.Deployable, udm.SourceArtifact, udm.Artifact, udm.DeployableArtifact, udm.ConfigurationItem, udm.FileArtifact

Apache configuration file

Public Properties	
excludeFileNamesRegex :	STRING Regular expression that matches file names that must be excluded from scanning
placeholders :	SET_OF_STRING Placeholders detected in this artifact
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

www.ApacheConfFragment

Type Hierarchy	generic.ProcessedTemplate >> generic.AbstractDeployedArtifact >> generic.AbstractDeployed >> udm.BaseDeployed >> udm.BaseConfigurationItem
Interfaces	udm.Deployed, udm.ConfigurationItem

Deployed generated Apache configuration file

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

Hidden Properties	
* createOrder : INTEGER = 50	The order of the step in the step list for the create operation.
* createVerb : STRING = Deploy	Create Verb
* destroyOrder : INTEGER = 40	The order of the step in the step list for the destroy operation.
* destroyVerb : STRING = Remove	Destroy Verb
* modifyOrder : INTEGER = 50	The order of the step in the step list for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* targetDirectory : STRING = \${deployed.container.configurationFragmentDirectory}	Target directory name
* targetFile : STRING = \${deployed.name}.conf	Target file name
* template : STRING = www/apache/\${deployed.type}.conf.ftl	Configuration fragment template file name.
createTargetDirectory : BOOLEAN = false	Create the target directory on the generic server if it does not exist.
inspectClasspathResources : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : 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.
restartRequired : BOOLEAN = true	Restart Required
restartRequiredForNoop : BOOLEAN = false	The generic container requires a restart for the NOOP action performed by this deployed.
targetDirectoryShared : BOOLEAN = true	Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

www.ApacheConfFragmentSpec

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

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

Specification for an Apache configuration fragment.

Public Properties	
tags : SET_OF_STRING	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		
* host : <code>CI<overthere>.Host</code>		Host upon which the container resides
Public Properties		
* configurationFragmentDirectory : <code>STRING</code>		The directory into which Apache configuration files (for virtual host definitions) should be written
* defaultDocumentRoot : <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
* restartWaitTime : <code>INTEGER = 10</code>		Duration (in secs) to wait after the restart server step has been executed
* startCommand : <code>STRING</code>		The command to start Apache. Arguments containing spaces are not supported
* startWaitTime : <code>INTEGER = 10</code>		Duration (in secs) to wait after the start server step has been executed
* stopCommand : <code>STRING</code>		The command to stop Apache. Arguments containing spaces are not supported
* stopWaitTime : <code>INTEGER = 10</code>		Duration (in secs) to wait after the stop server step has been executed
envVars : <code>MAP_STRING_STRING</code>		Environment variables for container
restartCommand : <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
tags : <code>SET_OF_STRING</code>		If set, only deployables with the same tag will be automatically mapped to this container.
Hidden Properties		
* restartOrder : <code>INTEGER = 90</code>		The order of the restart container step in the step list.
* restartScript : <code>STRING = <#assign restart=(container.restartCommand)!"><#if restart != ">www/apache/restart<#else>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
* startOrder : <code>INTEGER = 90</code>		The order of the start container step in the step list.
* startScript : <code>STRING = www/apache/start</code>		The command to start Apache. Arguments containing spaces are not supported
* stopOrder : <code>INTEGER = 10</code>		The order of the stop container step in the step list.
* stopScript : <code>STRING = www/apache/stop</code>		The command to stop Apache. Arguments containing spaces are not supported
inspectClasspathResources : <code>SET_OF_STRING</code>		Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : <code>STRING</code>		Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : <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		
	restart	Restart Apache
	start	Start Apache
	stop	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	
* container : CI<udm.Container>	The container on which this deployed runs.
Public Properties	
* host : STRING	The virtual host name. Use "*" to match all hosts
* port : STRING	The virtual host port. Use "*" to match all ports
deployable : CI<udm.Deployable>	The deployable that this deployed is derived from.
documentRoot : 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	
* createOrder : INTEGER = 50	The order of the step in the step list for the create operation.
* createVerb : STRING = Deploy	Create Verb
* destroyOrder : INTEGER = 40	The order of the step in the step list for the destroy operation.
* destroyVerb : STRING = Remove	Destroy Verb
* modifyOrder : INTEGER = 50	The order of the step in the step list for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* targetDirectory : STRING = \${deployed.container.configurationFragmentDirectory}	Target directory name
* targetFile : STRING = \${deployed.name}.conf	Target file name
* template : STRING = www/apache/\${deployed.type}.conf.ftl	Configuration fragment template file name.
createTargetDirectory : BOOLEAN = false	Create the target directory on the generic server if it does not exist.
inspectClasspathResources : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : 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.
restartRequired : BOOLEAN = true	Restart Required
restartRequiredForNoop : BOOLEAN = false	The generic container requires a restart for the NOOP action performed by this deployed.
targetDirectoryShared : BOOLEAN = true	Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

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	
documentRoot :	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 (string)
host :	STRING The virtual host name. Use "*" to match all hosts (string)
port :	STRING The virtual host port. Use "*" to match all ports (string)
tags :	SET_OF_STRING If set, this deployable will only be mapped automatically to containers with the same tag.

www.DeployedApacheConfFile

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

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

Deployed Apache configuration file

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

Hidden Properties	
* createOrder : INTEGER = 50	The order of the step in the step list for the create operation.
* createVerb : STRING = Deploy	Create Verb
* destroyOrder : INTEGER = 40	The order of the step in the step list for the destroy operation.
* destroyVerb : STRING = Remove	Destroy Verb
* modifyOrder : INTEGER = 50	The order of the step in the step list for the modify operation.
* modifyVerb : STRING = Modify	Modify Verb
* noopOrder : INTEGER = 50	The order of the step in the step list for the noop operation.
* noopVerb : STRING = Modify	Noop Verb
* targetDirectory : STRING = \${deployed.container.configurationFragmentDirectory}	Target directory name
* targetFile : STRING = \${deployed.name}	Target file name
createTargetDirectory : BOOLEAN = false	Create the target directory on the generic server if it does not exist.
inspectClasspathResources : SET_OF_STRING	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : STRING	Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : 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.
preserveExistingFiles : BOOLEAN = false	if true, preserve the existing files on the remote host (do not delete the copied files during the destroy operation).
restartRequired : BOOLEAN = true	Restart Required
restartRequiredForNoop : BOOLEAN = false	The generic container requires a restart for the NOOP action performed by this deployed.
targetDirectoryShared : BOOLEAN = true	Is the target directory shared by others on the generic server. When true, the target directory is not deleted during a destroy operation; only the artifacts copied to it.

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

Public Properties	
deployable : <code>CI<udm.Deployable></code>	The deployable that this deployed is derived from.
placeholders : <code>MAP_STRING_STRING</code>	A Map containing all the placeholders mapped to their values. Special values are <code><ignore></code> or <code><empty></code>
virtualHost : <code>CI<www.ApacheVirtualHost></code>	Virtual host which this web content maps to.
Hidden Properties	
* createOrder : <code>INTEGER = 50</code>	The order of the step in the step list for the create operation.
* createVerb : <code>STRING = Deploy</code>	Create Verb
* destroyOrder : <code>INTEGER = 40</code>	The order of the step in the step list for the destroy operation.
* destroyVerb : <code>STRING = Remove</code>	Destroy Verb
* modifyOrder : <code>INTEGER = 50</code>	The order of the step in the step list for the modify operation.
* modifyVerb : <code>STRING = Modify</code>	Modify Verb
* noopOrder : <code>INTEGER = 50</code>	The order of the step in the step list for the noop operation.
* noopVerb : <code>STRING = Modify</code>	Noop Verb
* targetDirectory : <code>STRING = <#assign target=(deployed.virtualHost.documentRoot)!"/><#if target != ">\${target}<#else>\${deployed.container.defaultDocumentRoot}</if></code>	Target directory name
createTargetDirectory : <code>BOOLEAN = false</code>	Create the target directory on the generic server if it does not exist.
inspectClasspathResources : <code>SET_OF_STRING</code>	Additional classpath resources that should be uploaded to the working directory before executing the inspect script.
inspectScript : <code>STRING</code>	Classpath to the script used to inspect the generic container.
inspectTemplateClasspathResources : <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.
preserveExistingFiles : <code>BOOLEAN = false</code>	if true, preserve the existing files on the remote host (do not delete the copied files during the destroy operation).
restartRequired : <code>BOOLEAN = false</code>	Restart Required
restartRequiredForNoop : <code>BOOLEAN = false</code>	The generic container requires a restart for the NOOP action performed by this deployed.
targetDirectoryShared : <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.
targetFile : <code>STRING</code>	Target File

www.WebContent

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

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

Public Properties	
excludeFileNamesRegex : STRING	Regular expression that matches file names that must be excluded from scanning
placeholders : SET_OF_STRING	Placeholders detected in this artifact
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