

# Deployit File Plugin Manual

Version 3.5.1.2

# Table of Content

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
Deployable Configuration Items	4
Deployed Configuration Items	4
Topology Configuration Items	4
Virtual Deployed Configuration Items	5
Virtual Topology Configuration Items	5
Configuration Item Details	5
file.Archive	5
file.DeployedArchive	5
file.DeployedArtifactOnHost	6
file.DeployedFile	7
file.DeployedFolder	7
file.File	8
file.Folder	8
overthere.CifsHost	9
overthere.Host	10
overthere.LocalHost	11
overthere.SshHost	11

# Preface

This document describes the functionality provided by the File Plugin.

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

## Overview

In many cases, an application depends on external resources for its configuration. The application accesses these resources from a predefined location or using a predefined mechanism. In the simplest of forms, a resource can be described as a file, an archive or a folder (collection of files). The File Plugin enables the definition of such resources in a deployment package and subsequently managing them on a target host. The resources can contain placeholders that the plugin will replace when targeting to the specific host, thus allowing resources to be defined independent of their environment.

## Features

- Deploy, upgrade, and undeploy of a file based resource on a [Host](#).
  - [File](#)
  - [Folder](#)
  - [Archive](#)

## Requirements

This plugin requires:

- **Deployit:** version 3.5+

## Usage in Deployment Packages

Please refer to *Packaging Manual* for more details about the DAR packaging format.

Sample DAR MANIFEST.MF entries defining a file, folder and archive resource

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

Name: sampleFile.txt
CI-Name: sampleFile
CI-Type: file.File

Name: sampleArchive.zip
CI-Name: sampleArchive
CI-Type: file.Archive

Name: sampleFolder
CI-Name: sampleFolder
CI-Type: file.Folder
```

## Using the deployables and deployed

### Deployable vs. Container Table

The following table describes which deployable / container combinations are possible. Note that the CIs can only be targeted to containers derived from [Host](#).

Deployables	Containers	Generated Deployed
file.File file.Archive	overthere.Host	file.DeployedFile file.DeployedArchive
file.Folder	overthere.Host	file.DeployedFolder

## Deployed Actions Table

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

Deployed	Create	Destroy	Modify
file.DeployedFile file.DeployedArchive	<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</li> </ul>	<ul style="list-style-type: none"> <li>Delete old file from host</li> <li>Copy modified file to target path on host</li> </ul>
file.DeployedFolder	<ul style="list-style-type: none"> <li>Create target folder on host, if needed</li> <li>Copy folder content to target folder on host</li> </ul>	<ul style="list-style-type: none"> <li>Delete folder content from host</li> <li>If folder is not a shared folder, the folder itself is deleted from host</li> </ul>	<ul style="list-style-type: none"> <li>Perform actions as described by <i>Destroy</i> for old folder</li> <li>Perform actions as described by <i>Create</i> for modified folder</li> </ul>

## CI Reference

### Configuration Item Overview

#### Deployable Configuration Items

CI	Description
<a href="#">file.Archive</a>	An archive (zip, jar, tar, etc) artifact that can be packaged in a Deployment Package
<a href="#">file.File</a>	A single file artifact that can be packaged in a Deployment Package
<a href="#">file.Folder</a>	A folder artifact that can be packaged in a Deployment Package

#### Deployed Configuration Items

CI	Description
<a href="#">file.DeployedArchive</a>	The archive as deployed on the Host
<a href="#">file.DeployedFile</a>	The file as deployed on the Host
<a href="#">file.DeployedFolder</a>	The folder as deployed on the Host

#### Topology Configuration Items

CI	Description
<a href="#">overthere.CifsHost</a>	A machine that can be connected to using either WinRM or Telnet and can perform file manipulation via the CIFS protocol
<a href="#">overthere.LocalHost</a>	The machine on which the Deployit Server is running on
<a href="#">overthere.SshHost</a>	A machine that can be connected to using ssh

## Virtual Deployed Configuration Items

CI	Description
<a href="#">file.DeployedArtifactOnHost</a>	Abstract deployed that can target any DeployableArtifact to a Host

## Virtual Topology Configuration Items

CI	Description
<a href="#">overthere.Host</a>	A machine that runs middleware, on which scripts can be executed, etc

## Configuration Item Details

### file.Archive

**Hierarchy** `udm.BaseDeployableArchiveArtifact >> udm.BaseDeployableFileArtifact >> udm.BaseDeployableArtifact >> udm.BaseDeployable >> udm.BaseConfigurationItem`

**Interfaces** `udm.Deployable, udm.SourceArtifact, udm.ArchiveArtifact, udm.Artifact, udm.DeployableArtifact, udm.ConfigurationItem, udm.FileArtifact`

An archive (zip, jar, tar, etc) artifact that can be packaged in a Deployment Package

Public Properties
<b>placeholders</b> : <code>SET_OF_STRING</code>
Placeholders detected in this artifact

Hidden Properties
<b>textFileNamesRegex</b> * : <code>STRING = .+\. (cfg   conf   config   ini   properties   props   txt   asp   aspx   htm   html   jsf   jsp   xht   xhtml   sql   xml   xsd   xsl   xslt)</code>
Regular expression that matches file names of text files

### file.DeployedArchive

**Hierarchy** `file.DeployedArtifactOnHost >> udm.BaseDeployedArtifact >> udm.BaseDeployed >> udm.BaseConfigurationItem`

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

The archive as deployed on the Host.

## Public Properties

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

The container on which this deployed runs.

**targetPath** : `STRING`

Path to which artifact must be copied to on the host.

**createTargetPath** : `BOOLEAN = false`

Create the targetPath on the host if it does not exist.

**deployable** : `CI<udm.Deployable>`

The deployable that this deployed is derived from.

**placeholders** : `MAP_STRING_STRING`

A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>

**targetFileName** : `STRING`

Name of the artifact on the host.

**targetPathShared** : `BOOLEAN = true`

Is the targetPath shared by others on the host. When true, the targetPath is not deleted during undeployment; only the artifacts copied to it.

## file.DeployedArtifactOnHost

**Hierarchy**     `udm.BaseDeployedArtifact >> udm.BaseDeployed >> udm.BaseConfigurationItem`

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

Abstract deployed that can target any DeployableArtifact to a Host.

## Public Properties

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

The container on which this deployed runs.

**targetPath** : `STRING`

Path to which artifact must be copied to on the host.

**createTargetPath** : `BOOLEAN = false`

Create the targetPath on the host if it does not exist.

**deployable** : `CI<udm.Deployable>`

The deployable that this deployed is derived from.

**placeholders** : `MAP_STRING_STRING`

A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>

**targetFileName** : `STRING`

Name of the artifact on the host.

**targetPathShared** : `BOOLEAN = true`

Is the targetPath shared by others on the host. When true, the targetPath is not deleted during undeployment; only the artifacts copied to it.

## file.DeployedFile

**Hierarchy** [file.DeployedArtifactOnHost](#) >> [udm.BaseDeployedArtifact](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

The file as deployed on the Host.

### Public Properties

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

The container on which this deployed runs.

**targetPath\*** : [STRING](#)

Path to which artifact must be copied to on the host.

**createTargetPath** : [BOOLEAN](#) = *false*

Create the targetPath on the host if it does not exist.

**deployable** : [CI](#)<[udm.Deployable](#)>

The deployable that this deployed is derived from.

**placeholders** : [MAP\\_STRING\\_STRING](#)

A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>

**targetFileName** : [STRING](#)

Name of the artifact on the host.

**targetPathShared** : [BOOLEAN](#) = *true*

Is the targetPath shared by others on the host. When true, the targetPath is not deleted during undeployment; only the artifacts copied to it.

## file.DeployedFolder

**Hierarchy** [file.DeployedArtifactOnHost](#) >> [udm.BaseDeployedArtifact](#) >> [udm.BaseDeployed](#) >> [udm.BaseConfigurationItem](#)

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

The folder as deployed on the Host.

## Public Properties

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

The container on which this deployed runs.

**targetPath**\* : `STRING`

Path to which artifact must be copied to on the host.

**createTargetPath** : `BOOLEAN = false`

Create the targetPath on the host if it does not exist.

**deployable** : `CI<udm.Deployable>`

The deployable that this deployed is derived from.

**placeholders** : `MAP_STRING_STRING`

A Map containing all the placeholders mapped to their values. Special values are <ignore> or <empty>

**targetPathShared** : `BOOLEAN = true`

Is the targetPath shared by others on the host. When true, the targetPath is not deleted during undeployment; only the artifacts copied to it.

## Hidden Properties

**targetFileName** : `STRING`

Not applicable for this type.

## file.File

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

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

A single file artifact that can be packaged in a Deployment Package

## Public Properties

**placeholders** : `SET_OF_STRING`

Placeholders detected in this artifact

## Hidden Properties

**textFileNamesRegex**\* : `STRING = .+\. (cfg | conf | config | ini | properties | props | txt | asp | aspx | htm | html | jsf | jsp | xht | xhtml | sql | xml | xsd | xsl | xslt)`

Regular expression that matches file names of text files

## file.Folder

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

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



A folder artifact that can be packaged in a Deployment Package

### Public Properties

**placeholders** : SET\_OF\_STRING

Placeholders detected in this artifact

### Hidden Properties

**textFileNamesRegex**\* : STRING = *.+\. (cfg | conf | config | ini | properties | props | txt | asp | aspx | htm | html | jsf | jsp | xht | xhtml | sql | xml | xsd | xsl | xslt)*

Regular expression that matches file names of text files

## overthere.CifsHost

**Hierarchy** `overthere.Host` >> `udm.BaseConfigurationItem`

**Interfaces** `udm.ConfigurationItem`, `udm.Container`

A machine that can be connected to using either WinRM or Telnet and can perform file manipulation via the CIFS protocol.

### Public Properties

**address**\* : STRING

Address of the host

**connectionType**\* : ENUM [TELNET, WINRM\_HTTP, WINRM\_HTTPS] = *TELNET*

Connection Type

**os**\* : ENUM [WINDOWS, UNIX]

Operating system

**password**\* : STRING

Password to use for authentication

**username**\* : STRING

Username to connect with

**cifsPort** : INTEGER = *445*

Port on which the CIFS server runs

**port** : INTEGER

Port on which the Telnet or WinRM server runs

**temporaryDirectoryPath** : STRING

Directory into which temporary files are stored. Will be cleaned up when the connection is closed.

## Hidden Properties

**connectionTimeoutMillis**\* : INTEGER = 1200000

Connection Timeout Millis

**protocol**\* : STRING = cifs

Protocol

**tmpFileCreationRetries**\* : INTEGER = 1000

Tmp File Creation Retries

**winrmContext**\* : STRING = /wsman

Winrm Context

**winrmEnvelopSize**\* : INTEGER = 153600

Winrm Envelop Size

**winrmLocale**\* : STRING = en-US

Winrm Locale

**winrmTimeout**\* : STRING = PT60.000S

Winrm Timeout

**tmpDeleteOnDisconnect** : BOOLEAN = true

Whether to delete the temporary connection directory when the connection is closed

## overthere.Host

**Hierarchy** udm.BaseConfigurationItem

**Interfaces** udm.ConfigurationItem, udm.Container

A machine that runs middleware, on which scripts can be executed, etc.

## Public Properties

**os**\* : ENUM [WINDOWS, UNIX]

Operating system

**temporaryDirectoryPath** : STRING

Directory into which temporary files are stored. Will be cleaned up when the connection is closed.

## Hidden Properties

**connectionTimeoutMillis**\* : INTEGER = 1200000

Connection Timeout Millis

**protocol**\* : STRING

Protocol to use when connecting to this host

**tmpFileCreationRetries**\* : INTEGER = 1000

Tmp File Creation Retries

**tmpDeleteOnDisconnect** : BOOLEAN = true

Whether to delete the temporary connection directory when the connection is closed

## overthere.LocalHost

**Hierarchy** `overthere.Host` >> `udm.BaseConfigurationItem`

**Interfaces** `udm.ConfigurationItem`, `udm.Container`

The machine on which the Deployit Server is running on.

### Public Properties

**os**\* : `ENUM [WINDOWS, UNIX]`

Operating system

**temporaryDirectoryPath** : `STRING`

Directory into which temporary files are stored. Will be cleaned up when the connection is closed.

### Hidden Properties

**connectionTimeoutMillis**\* : `INTEGER = 1200000`

Connection Timeout Millis

**protocol**\* : `STRING = local`

Protocol

**tmpFileCreationRetries**\* : `INTEGER = 1000`

Tmp File Creation Retries

**tmpDeleteOnDisconnect** : `BOOLEAN = true`

Whether to delete the temporary connection directory when the connection is closed

## overthere.SshHost

**Hierarchy** `overthere.Host` >> `udm.BaseConfigurationItem`

**Interfaces** `udm.ConfigurationItem`, `udm.Container`

A machine that can be connected to using ssh.

## Public Properties

**address**\* : *STRING*

Address of the host

**connectionType**\* : *ENUM [SFTP, SCP, SUDO, INTERACTIVE\_SUDO] = SFTP*

Type of SSH connection to create

**os**\* : *ENUM [WINDOWS, UNIX]*

Operating system

**port**\* : *INTEGER = 22*

Port on which the SSH server runs

**username**\* : *STRING*

Username to connect with

**passphrase** : *STRING*

Optional passphrase for the private key in the private key file

**password** : *STRING*

Password to use for authentication

**privateKeyFile** : *STRING*

Private key file to use for authentication

**sudoUsername** : *STRING*

Username to sudo to when accessing files or executing commands

**temporaryDirectoryPath** : *STRING*

Directory into which temporary files are stored. Will be cleaned up when the connection is closed.

## Hidden Properties

**connectionTimeoutMillis**\* : *INTEGER = 1200000*

Connection Timeout Millis

**protocol**\* : *STRING = ssh*

Protocol

**sudoPasswordPromptRegex**\* : *STRING = .\*[Pp]assword.\*:*

Sudo Password Prompt Regex

**tmpFileCreationRetries**\* : *INTEGER = 1000*

Tmp File Creation Retries

**allocateDefaultPty** : *BOOLEAN = true*

Allocate Default Pty

**tmpDeleteOnDisconnect** : *BOOLEAN = true*

Whether to delete the temporary connection directory when the connection is closed