

Deployit Frequently Asked Questions

Version 3.9

Table of Contents

Table of Contents	2
Frequently Asked Questions	3
Deployit support	3
How do I contact XebiaLabs technical support?	3
What level of support does XebiaLabs offer?	3
Server Installation	3
Can I perform an unattended install of Deployit, for instance using Puppet?	3
Server configuration and startup	3
How do I prevent Deployit from writing temporary files for imported packages?	3
How do I enable additional logging for the Deployit server?	4
How do I configure Deployit to use specific file encoding?	4
Middleware server configuration	4
Where can I find more information about configuring middleware for use with Deployit?	4
Do I always need a CIFS connection to my Windows middleware hosts?	4
CLI configuration and startup	4
I get the message: 'Error loading terminal, using fallback. Your terminal will have reduced functionality.'	4
CLI usage	4
How do I create the most common CIs in the CLI?	4
Packaging	5
How do I refer to another CI in my manifest file?	5
WAS Plugin	5
Why does Deployit hang when it starts up wsadmin for the first time?	5
Customization and extension	5
Can I add a synthetic task to an existing type (e.g. SshHost) without modifying it?	5
How do I turn off placeholder scanning?	6

Frequently Asked Questions

Deployit support

How do I contact XebiaLabs technical support?

To contact Deployit support, use our [support desk](#). For urgent problems, the XebiaLabs support team can also be reached by phone at **+31610930435** (standard European business hours and workdays only).

What level of support does XebiaLabs offer?

We classify incoming support issues as follows:

- **Urgent:** Application or major part of the application freezes, crashes or fails to start or data is corrupted.
- **High:** Key feature does not work, can not be used or returns incorrect results. No workaround is available.
- **Normal:** Key feature is difficult to use or looks terrible. A secondary feature does not work, cannot be used or returns incorrect results. A High issue for which there is a workaround.
- **Low:** Secondary feature has a cosmetic issue. Minor feature is difficult to use or looks terrible. Minor glitches in images, spell mistakes, etc.

Depending on the priority of the issue, XebiaLabs will assign a specialist to work on the issue:

- **Urgent:** XebiaLabs will assign a specialist to provide a workaround or work on correcting the issue within 4 hours on a working day.
- **High, Normal:** XebiaLabs will assign a specialist to provide a workaround or work on correcting the issue within 2 working days.
- **Low:** XebiaLabs may include an update in the next maintenance release.

Please note that these are indications of when we will start working on the issue. We can not guarantee a solution within a particular timeframe as this depends too much on the complexities of the issue.

Server Installation

Can I perform an unattended install of Deployit, for instance using Puppet?

It is possible to do an unattended install. Both the server and command line interface (CLI) are distributed as separate ZIP archives. Installation for the CLI is as simple as extracting the ZIP file.

The server needs some more configuration. When the server is started, it looks for a file called *deployit.conf* in the *conf* directory in it's home directory. If this does not exist, it enters an interactive setup wizard to create it.

An unattended install can be performed by including a *deployit.conf* file in the package and copy it to the *conf* directory once the installation ZIP file is extracted. You could do the installation manually once to obtain a *deployit.conf* file. After this installation, Deployit server can be started without entering the setup wizard. If you are using a Deployit repository on disk, you do have to create the empty repository directory manually as this is normally done by the setup wizard.

Another option is to run through the setup wizard automatically, accepting all the defaults. This Unix command will do that:

```
yes yes | bin/server.sh -setup
```

Server configuration and startup

How do I prevent Deployit from writing temporary files for imported packages?

When uploading a package using the CLI, Deployit stores a temporary file on the server. This file is deleted only if you shut down the JVM. An alternative is to make Deployit read the archive in memory. To do this, use the following setting when starting the Deployit server:

```
-Dorg.apache.james.mime4j.defaultStorageProvider=org.apache.james.mime4j.storage.MemoryStorageProvider
```

How do I enable additional logging for the Deployit server?

Logging is configured in the file `SERVER_HOME/conf/logback.xml`. To enable debug mode, change the following in the logback file:

```
<root level="debug">
  ...
</root>
```

If this results in too much logging, you can tailor logging for specific packages by adding loglevel definitions for them. For example:

```
<logger name="com.xebialabs" level="info" />
```

Note that the server needs to be restarted to activate the new log settings.

See the [logback site](#) for more information.

How do I configure Deployit to use specific file encoding?

Deployit uses file.encoding system property. To change file encoding other than system default set the following system property when starting the Deployit server:

```
For Oracle JDK:
-Dfile.encoding=<FileEncodingType>
e.g: -Dfile.encoding=UTF-8

For IBM JDK:
-Dclient.encoding.override=<FileEncodingType>
e.g: -Dclient.encoding.override=UTF-8
```

Middleware server configuration

Where can I find more information about configuring middleware for use with Deployit?

See the [documentation provided with the Overthere framework](#).

Do I always need a CIFS connection to my Windows middleware hosts?

Yes. Deployit can use Telnet or WinRM to execute commands on the middleware hosts, but needs CIFS to transfer files to the middleware host.

CLI configuration and startup

I get the message: 'Error loading terminal, using fallback. Your terminal will have reduced functionality.'

The Deployit CLI was unable to load the correct Terminal. If you're running the CLI on Windows 2003 or Windows 2008, the solution for this is to install the [Microsoft Visual C++ 2008 Redistributable Package](#).

CLI usage

How do I create the most common CIs in the CLI?

The following snippet shows examples of creating common UDM CIs.

```
# Create a host
host = factory.configurationItem('Infrastructure/sampleHost', 'overthere.SshHost', { 'os': 'UNIX', 'address':
'localhost', 'username': 'scott' })
repository.create(host)
deployit.print(host)

# Create a dictionary
dict = factory.configurationItem('Environments/myDict', 'udm.Dictionary')
dict.entries = { 'a': '1', 'b': '2' }
repository.create(dict)
deployit.print(dict)

# Create an environment
env = factory.configurationItem('Environments/sampleEnv', 'udm.Environment')
env.dictionaries = [ dict.id ]
env.members = [ host.id ]
repository.create(env)
deployit.print(env)
```

Packaging

How do I refer to another CI in my manifest file?

You must use the value of the *Name* attribute of a CI to refer to another CI in the manifest. This is true, even if you use the *CI-Name* property to provide a CI name that is different from the artifact name.

For example, in the following manifest, the *_wsdl_andmapping* CI has a reference to the *Strawberry-wsdl*s and *Strawberry-wsdl*s-mapping CIs:

Name: wsdl_and_mapping
CI-Name: Strawberry-wsdl
CI-Type: org.WsdlSource
CI-Wsdl: wsdl
CI-Mapping: wsdl/mapping/mapping.csv
Name: wsdl
CI-Name: Strawberry-wsdl
CI-Type: org.Wsdl
Name: wsdl/mapping/mapping.csv
CI-Name: Strawberry-wsdl-mapping
CI-Type: org.WsdlMapping

WAS Plugin

Why does Deployit hang when it starts up wsadmin for the first time?

When Deployit starts up *wsadmin* for the first time on a machine, the user has to interactively accept the dmgr certificate. Deployit cannot do that so it will hang.

This is the output shown in the step log:

SSL SIGNER EXCHANGE PROMPT ***
SSL signer from target host null is not found in trust store
/opt/ws/6.1/appserver/profiles/AppSrv01/etc/trust.pl2.
Here is the signer information (verify the digest value matches what is displayed at the server):
Subject DN: CN=was-61-sa, O=IBM, C=US
Issuer DN: CN=was-61-sa, O=IBM, C=US
Serial number: 1306835778
Expires: Wed May 30 11:56:18 CEST 2012
SHA-1 Digest: C9:A3:48:43:BD:20:96:67:AF:51:E5:9A:EE:46:60:EC:6F:0E:F6:51
MD5 Digest: 15:43:57:AD:03:74:A0:DB:158:BE:4A:68:A4:57:6C
Add signer to the trust store now? (y/n)

Customization and extension

Can I add a synthetic task to an existing type (e.g. SshHost) without modifying it?

Note that synthetic types are present from 3.6 onwards

At the moment it is not possible to synthetically () add control tasks to an existing type such as *Host*. However you could achieve the same functionality by using the Generic Model Plugin, which does support synthetic control tasks.

Example :

1. Define your custom container, that extends the generic container, which defines the control task and its associated script to run for the task. The scripts are freemarker templates that get render, copied to the target host and executed.

<type type="mycompany.ConnectionTest" extends="generic.Container">
<!-- inherited hidden -->
<property name="startProcessScript" default="mycompany/connectiontest/start" hidden="true"/>
<property name="stopProcessScript" default="mycompany/connectiontest/stop" hidden="true"/>
<!-- control tasks -->
<method name="start" description="Start some process"/>
<method name="stop" description="Stop some process"/>
</type>

2. Create the container under the host you wish to test in the repository editor.

3. Execute the control task.

How do I turn off placeholder scanning?

When importing a package, Deployit by default scans the artifacts it contains for placeholders that need to be resolved during a deployment. If you want to turn off placeholder scanning, there are various ways to do this.

Disabling placeholder scanning for one file extension on a particular artifact type

Deployit looks for files to scan in artifact CIs based on the file extension. It is possible to exclude certain extensions from this process. To do this, edit the `deployit-defaults.properties` file and set the `excludeFileNamesRegex` property on the artifact CI type you want to exclude. For example:

```
file.Archive.excludeFileNamesRegex=.*\\.js
```

Restart the Deployit server for the change to take effect.

Disabling placeholder scanning for one file extension on all artifacts

Deployit looks for files to scan in artifact CIs based on the file extension. It is possible to exclude certain extensions from this process. To do this, edit the `deployit-defaults.properties` file and set the `excludeFileNamesRegex` property on the artifact CI type you want to exclude. For example:

```
udm.BaseDeployableArchiveArtifact.excludeFileNamesRegex=.*\\.js
```

Restart the Deployit server for the change to take effect.

Disabling placeholder scanning for one CI instance

Edit the deployment package manifest and change the `scanPlaceholders` property of the particular artifact:

```
Name: sampleArchive.zip
CI-Name: sampleArchive
CI-Type: file.Archive
CI-scanPlaceholders: false
```

Disabling placeholder scanning for one CI type

Edit the `deployit-defaults.properties` file and set the `scanPlaceholders` property for the CI type you want to exclude. For example:

```
file.Archive.scanPlaceholders=false
```

Restart the Deployit server for the change to take effect.

Disabling placeholder scanning completely

Edit the `deployit-defaults.properties` file and set the following property:

```
udm.BaseDeployableArtifact.scanPlaceholders=false
```

Restart the Deployit server for the change to take effect.

For more information, see the [packaging manual](#).