

# Deployit Tomcat Plugin Manual

January, 2011

## Contents

<b>Preface</b>	<b>1</b>
<b>Introduction</b>	<b>1</b>
<b>Tomcat Plugin Requirements</b>	<b>1</b>
<b>Supported Tomcat Versions</b>	<b>1</b>
<b>Supported Tomcat Features</b>	<b>2</b>
<b>Tomcat Runbook</b>	<b>2</b>
<b>Tomcat Configuration Items (CIs)</b>	<b>2</b>
ContextFiles . . . . .	3
ContextFilesMapping . . . . .	3
TomcatActiveMQConnectionFactory . . . . .	3
TomcatConfigurationFilesMapping . . . . .	3
TomcatDataSource . . . . .	4
TomcatJNDIResources . . . . .	5
TomcatJNDIResourcesMapping . . . . .	6
TomcatLibrariesMapping . . . . .	6
TomcatManagedServer . . . . .	6
TomcatResourceMapping . . . . .	7
TomcatUnmanagedServer . . . . .	7
TomcatWarMapping . . . . .	8

## Preface

This manual describes the Deployit Tomcat Plugin.

## Introduction

The Tomcat Plugin supports the deployment, re-deployment and undeployment of a deployment package to a Tomcat servlet container.

## Tomcat Plugin Requirements

In addition to the requirements for Deployit, the Tomcat Plugin has the following additional requirements:

- the user account used to access the Tomcat server must have permission to perform the following actions:
  - write to the Tomcat context file.
  - run the Tomcat start / stop script (e.g. `catalina.sh`)

## Supported Tomcat Versions

The Tomcat plugin supports the following versions of Tomcat:

- **5.5.x**
- **6.0.x**

## Supported Tomcat Features

The Tomcat Plugin supports the following features:

Concept	Remarks
WAR files	Deploy and undeploy WAR archives to Tomcat.
Context files	Deploy and undeploy folders containing context files.
ActiveMQ connection factory	Deploy and undeploy ActiveMQ connection factories.
Datasources	Deploy and undeploy datasources to Tomcat.
JNDI Resources	Deploy and undeploy resources in the Tomcat JNDI tree.
Libraries	Deploy and undeploy Java libraries to Tomcat.
Tomcat Managed Server	Deploy and undeploy to Tomcat servers using the Tomcat manager app.
Tomcat Unmanaged Server	Deploy and undeploy to Tomcat servers without the Tomcat manager app.

## Tomcat Runbook

When the Tomcat runbook is triggered, the plugin populates the steplist with steps based on the executed task. First, the Tomcat runbook determines which servers are affected by the pending task. These are all the Tomcat servers that are a target of one of the deployed items in the deployment or the Tomcat server that a deployed application is running on in case of an undeploy.

The Tomcat runbook adds steps in the following order:

- Stop all servers affected by the deployment.
- Undeploy removed datasources from the Tomcat server.
- Undeploy removed wars from the Tomcat server.
- Undeploy removed libraries from the host.
- Undeploy removed configuration files from the host.
- Deploy SQL folders and scripts to the host.
- Deploy added libraries on the host.
- Deploy added configuration files on the host.
- Deploy added datasources to the Tomcat server.
- Deploy added wars to the Tomcat server.
- Start all affected servers.
- Deploy wars to managed Tomcat servers.

## Tomcat Configuration Items (CIs)

The Tomcat Plugin defines configuration items (CIs) needed to deploy to Tomcat middleware. To get more information about these CIs, use Deployit's command line interface (CLI). See the **Deployit Command Line Interface (CLI) Manual** for more information.

### ContextFiles

Folder in the Deployment Package containing Tomcat context files

*Type:* com.xebialabs.deployit.plugin.tomcat.ci.ContextFiles

*Properties:*

- **location**(**STRING**): Location of the artifact.
- **name**(**STRING**): The technical name of the artifact as it will be used within application servers.

## ContextFilesMapping

A mapping of a ContextFiles CI to a TomcatServer

*Type:* com.xebialabs.deployit.plugin.tomcat.ci.ContextFilesMapping

*Properties:*

- **source(java.io.Serializable):** Source
- **target(java.io.Serializable):** Target
- **keyValuePairs(List<com.xebialabs.deployit.ci.mapping.KeyValuePair>):** Key Value Pairs
- **placeholderFormat(ENUM):** Placeholder Format
  - Values: [SPRING, WINDOWS\_SHELL, STARS, NONE]

## TomcatActiveMQConnectionFactory

Description unavailable

*Type:* com.xebialabs.deployit.plugin.tomcat.ci.TomcatActiveMQConnectionFactory

*Properties:*

- **brokerName(STRING):** Broker Name
- **brokerURL(STRING):** Broker URL
- **factory(STRING):** Factory
- **jndiName(STRING):** Jndi Name

## TomcatConfigurationFilesMapping

A mapping of Configuration Files CI to a TomcatServer

*Type:* com.xebialabs.deployit.plugin.tomcat.ci.TomcatConfigurationFilesMapping

*Properties:*

- **source(java.io.Serializable):** Source
- **target(java.io.Serializable):** Target
- **targetDirectory(STRING):** The directory on the target host to which the configuration is to be copied. Example: /data/petclinic/1.0/conf. It can be overridden by using Key TARGET\_DIRECTORY
- **createIfNotExist(BOOLEAN):** create the target directory and sub directory if they are not exist
- **keyValuePairs(List<com.xebialabs.deployit.ci.mapping.KeyValuePair>):** Key Value Pairs
- **placeholderFormat(ENUM):** Placeholder Format
  - Values: [SPRING, WINDOWS\_SHELL, STARS, NONE]
- **shared(BOOLEAN):** The target directory is shared when another resources using it as well, default true.

## TomcatDataSource

Description unavailable

Type: com.xebialabs.deployit.plugin.tomcat.ci.TomcatDataSource

Properties:

- *accessToUnderlyingConnectionAllowed(BOOLEAN)*: Controls if the PoolGuard allows access to the underlying connection, default value is false
- *connectionInitSqls(STRING)*: A Collection of SQL statements that will be used to initialize physical connections when they are first created, default value is null
- *connectionProperties(STRING)*: The connection properties that will be sent to our JDBC driver when establishing new connections
- *connectionUrl(STRING)*: The connection URL to be passed to our JDBC driver to establish a connection
- *defaultAutoCommit(BOOLEAN)*: The default auto-commit state of connections created by this pool, default value is true
- *defaultCatalog(STRING)*: The default catalog of connections created by this pool
- *defaultReadOnly(STRING)*: The default read-only state of connections created by this pool, default value is driver default
- *driverClass(STRING)*: The fully qualified Java class name of the JDBC driver to be used
- *initialSize(INTEGER)*: The initial number of connections that are created when the pool is started, default value is 0
- *jndiName(STRING)*: Jndi Name
- *logAbandoned(BOOLEAN)*: Flag to log stack traces for application code which abandoned a Statement or Connection, default value is false
- *maxActive(INTEGER)*: The maximum number of active connections that can be allocated from this pool at the same time, or negative for no limit, default value is 8
- *maxIdle(INTEGER)*: The maximum number of connections that can remain idle in the pool, without extra ones being released, or negative for no limit, default value is 8
- *maxOpenPreparedStatements(INTEGER)*: The maximum number of open statements that can be allocated from the statement pool at the same time, or zero for no limit, default value is unlimited
- *maxWait(INTEGER)*: The maximum number of milliseconds that the pool will wait (when there are no available connections) for a connection to be returned before throwing an exception, or -1 to wait indefinitely, default value is indefinitely
- *minEvictableIdleTimeMillis(INTEGER)*: The minimum amount of time an object may sit idle in the pool before it is eligible for eviction by the idle object evictor (if any), default value is 1000 \* 60 \* 30
- *minIdle(INTEGER)*: The minimum number of connections that can remain idle in the pool, without extra ones being created, or zero to create none, default value is 0
- *numTestsPerEvictionRun(INTEGER)*: The number of objects to examine during each run of the idle object evictor thread (if any), default value is 3

- *password*(*STRING*): The connection password to be passed to our JDBC driver to establish a connection
- *poolPreparedStatements*(*BOOLEAN*): Enable prepared statement pooling for this pool, default value is false
- *removeAbandoned*(*BOOLEAN*): Flag to remove abandoned connections if they exceed the *removeAbandonedTimeout*, default value is false
- *removeAbandonedTimeout*(*INTEGER*): Timeout in seconds before an abandoned connection can be removed, default value is 300
- *testOnBorrow*(*BOOLEAN*): The indication of whether objects will be validated before being borrowed from the pool, default value is true
- *testOnReturn*(*BOOLEAN*): The indication of whether objects will be validated before being returned to the pool, default value is false
- *testWhileIdle*(*BOOLEAN*): The indication of whether objects will be validated by the idle object evictor (if any), default value is false
- *timeBetweenEvictionRunsMillis*(*INTEGER*): The number of milliseconds to sleep between runs of the idle object evictor thread, default value is -1
- *username*(*STRING*): The connection username to be passed to our JDBC driver to establish a connection
- *validationQuery*(*STRING*): The SQL query that will be used to validate connections from this pool before returning them to the caller

## TomcatJNDIResources

Description unavailable

*Type:* com.xebialabs.deployit.plugin.tomcat.ci.TomcatJNDIResources

*Properties:*

- *resourceLinks*(*List*<*com.xebialabs.deployit.plugin.tomcat.ci.ResourceLink*>): Resources links that are to stored in the Tomcat JNDI tree

## TomcatJNDIResourcesMapping

Description unavailable

*Type:* com.xebialabs.deployit.plugin.tomcat.ci.TomcatJNDIResourcesMapping

*Properties:*

- **source**(**java.io.Serializable**): Source
- **target**(**java.io.Serializable**): Target
- *resourceLinks*(*List*<*com.xebialabs.deployit.plugin.tomcat.ci.ResourceLink*>): Resources links that are to stored in the Tomcat JNDI tree

## TomcatLibrariesMapping

A mapping of Libraries Files CI to a TomcatServer

*Type:* com.xebialabs.deployit.plugin.tomcat.ci.TomcatLibrariesMapping

*Properties:*

- **source(java.io.Serializable):** Source
- **target(java.io.Serializable):** Target
- **targetDirectory(STRING):** The directory on the target host to which the configuration is to be copied. Example: /data/petclinic/1.0/conf. It can be overridden by using Key TARGET\_DIRECTORY
- **createIfNotExist(BOOLEAN):** create the target directory and sub directory if they are not exist
- **keyValuePairs(List<com.xebialabs.deployit.ci.mapping.KeyValuePair>):** Key Value Pairs
- **placeholderFormat(ENUM):** Placeholder Format
  - Values: [SPRING, WINDOWS\_SHELL, STARS, NONE]
- **shared(BOOLEAN):** The target directory is shared when another resources using it as well, default true.

## TomcatManagedServer

Tomcat Server instance

*Type:* com.xebialabs.deployit.plugin.tomcat.ci.TomcatManagedServer

*Properties:*

- **host(com.xebialabs.deployit.ci.Host):** Host on which the Tomcat server is running.
- **managerAppUrl(STRING):** URL of the manager app URL, e.g. http://tomcat-6:8080/manager
- **managerPassword(STRING):** Password to be used for the manager application deployed on tomcat
- **managerUsername(STRING):** Username to be used for the manager application deployed on tomcat
- **tomcatHome(STRING):** Place where Tomcat is installed such as /opt/apache-tomcat-6.0.24.
- **ajpPort(INTEGER):** AJP Port for the Tomcat Server, default is 8009
- **appBase(STRING):** Tomcat appBase, ex webapps
- **service(STRING):** Tomcat service, ex Catalina

## TomcatResourceMapping

Description unavailable

*Type:* com.xebialabs.deployit.plugin.tomcat.ci.TomcatResourceMapping

*Properties:*

- **source(java.io.Serializable):** Source
- **target(java.io.Serializable):** Target
- **sourcePropertyOverrides(List<com.xebialabs.deployit.ci.mapping.KeyValuePair>):** Overrides for properties of the mapping's source. The key is the property name (consult the documentation or run 'describe' in the CLI), the value is the value to set. Only string, integer and enumerable properties can be overridden. Example: Key: redeliveryLimits, Value: 2

## TomcatUnmanagedServer

Tomcat Server instance

*Type:* com.xebialabs.deployit.plugin.tomcat.ci.TomcatUnmanagedServer

*Properties:*

- **baseUrl(String):** The base URL which will be used to check whether server is running, e.g. http://tomcat-6:8080
- **host(com.xebialabs.deployit.ci.Host):** Host on which the Tomcat server is running.
- **startCommand(String):** Command that should be executed to start the Tomcat server, e.g. /opt/tomcat/bin/catalina start.
- **stopCommand(String):** Command that should be executed to stop the Tomcat server, e.g. /opt/tomcat/bin/catalina stop
- **tomcatHome(String):** Place where Tomcat is installed such as /opt/apache-tomcat-6.0.24.
- **ajpPort(Integer):** AJP Port for the Tomcat Server, default is 8009
- **appBase(String):** Tomcat appBase, ex webapps
- **service(String):** Tomcat service, ex Catalina

## TomcatWarMapping

A mapping of a WAR to a Tomcat server

*Type:* com.xebialabs.deployit.plugin.tomcat.ci.TomcatWarMapping

*Properties:*

- **source(java.io.Serializable):** Source
- **target(java.io.Serializable):** Target



- *appBase*(*STRING*): absolute path of the directory where wars are deployed. Only needed when context path specified in the mapping is different from the war name
- *contextRoot*(*STRING*): Context root to deploy to
- *ejbReferences*(*List*<*com.xebialabs.deployit.ci.mapping.EjbReference*>): Specifies the mapping from ejb reference jndi names and locals used in the web.xml to bean jndi names available in middleware
- *exploded*(*BOOLEAN*): Explode the war file on tomcat server (does not work when using the Tomcat manager).
- *keyValuePairs*(*List*<*com.xebialabs.deployit.ci.mapping.KeyValuePair*>): Key Value Pairs
- *placeholderFormat*(*ENUM*): Placeholder Format
  - Values: [SPRING, WINDOWS\_SHELL, STARS, NONE]
- *resourceEnvironmentEntryReferences*(*List*<*com.xebialabs.deployit.ci.mapping.ResourceReference*>): Specifies the mapping from resource environment references jndi names and types used in the web.xml to resource references jndi names available in middleware
- *resourceReferences*(*List*<*com.xebialabs.deployit.ci.mapping.ResourceReference*>): Specifies the mapping from resource references jndi names and types used in the web.xml to resource references jndi names available in middleware
- *virtualHost*(*STRING*): Virtual host to deploy to