

WebLogic Deployer Plugin

Plugin Information

View Deploy WebLogic on the [plugin site](#) for more information.

This plugin deploys any artifacts built on Jenkins to a weblogic target (managed server, cluster ...) as an application or a library component.

- [Pre-requisites](#)
- [Configuration](#)
 - [System Configuration](#)
 - [Authentication mode](#)
 - [Job Configuration](#)
 - [If the plugin is not configured](#)
 - [If the plugin is already configured](#)
 - [Customization of parameters](#)
 - [Job Advanced configuration](#)
 - [Stage mode](#)
 - [Override deployment execution](#)
- [User Guide](#)
 - [Build status](#)
 - [Check a build deployment status](#)
 - [Check the last successful deployment on main job dashboard](#)
 - [View deployment Logs](#)
 - [Disabling deployment task at runtime](#)
 - [Master/slave architecture support](#)
- [Release Notes](#)
 - [Version 4.1 \(September 03, 2017\)](#)
 - [Version 4.0 \(August 21, 2017\)](#)
 - [Version 3.7 \(June 21, 2017\)](#)
 - [Version 3.6 \(Oct 25, 2016\)](#)
 - [Version 3.5 \(Sep 04, 2016\)](#)
 - [Version 3.4 \(May 26, 2016\)](#)
 - [Version 3.3 \(August 31, 2015\)](#)
 - [Version 3.1 \(March 31, 2015\)](#)
 - [Version 3.0 \(March 28, 2015\)](#)
 - [Version 2.13 \(February 16, 2015\)](#)
 - [Version 2.12 \(December 17, 2014\)](#)
 - [Version 2.11 \(December 16, 2014\)](#)
 - [Version 2.10 \(December 6, 2014\)](#)
 - [Version 2.9.1 \(December 20, 2013\)](#)
 - [Version 2.9 \(November 30, 2013\)](#)
 - [Version 2.8 \(October 06, 2013\)](#)
 - [Version 2.7 \(October 04, 2013\)](#)
 - [Version 2.6 \(September 16, 2013\)](#)
 - [Version 2.5 \(July 27, 2013\)](#)
 - [Version 2.4 \(July 24, 2013\)](#)
 - [Version 2.3 \(June 26, 2013\)](#)
 - [Version 2.2 \(May 04, 2013\)](#)
 - [Version 2.1 \(May 03, 2013\)](#)
 - [Version 2.0 \(Mar 04, 2013\)](#)
 - [Version 1.4 \(Feb 11, 2013\)](#)
 - [Version 1.3 \(Aug 13, 2012\)](#)
 - [Version 1.2 \(Aug 2, 2012\)](#)
 - [Version 1.1 \(May 5, 2012\)](#)

Pre-requisites

*This plugin works with at least **1.485** jenkins version.
This plugin only deploys ear, war, jar artifacts.*

Configuration

System Configuration

Before use it, the plugin has to be configured. It consists of loading the weblogic target configuration, choosing the default JDK to run a deployment task and adding the weblogic library path :

- for WLS 8.X /pathtojar/weblogic.jar
- for WLS 10.3 /pathtojar/wlfullclient.jar
- for WLS 12.1 /pathtojar/wlthint3client.jar.

For building the wfullclient.jar look at here: http://docs.oracle.com/cd/E12840_01/wls/docs103/client/jarbuilder.html.

Further information at <https://docs.oracle.com/middleware/1213/wls/SACLT/basics.htm#SACLT124>



Workaround for WebLogic 12.2

For an unknown reason, deployment fails with libraries generated from a weblogic 12.2 product. You can generate and use a **wfullclient-10.3.6.jar**, with that I can deploy without any problem onto our WebLogic Server Version: 12.2.1.1.0.

WebLogic Deployment Plugin	
Jdk	system (defined in java.home)
Java Options to use	-Xms256M -Xmx256M
Additional classpath	P:\Documents\perso\wls-deploy-plugin\weblogic.jar
Configuration File	P:\Documents\perso\wls-deploy-plugin\default.xml
Excluding artifact name list	<input type="text"/>
Disable	<input type="checkbox"/>

- *Jdk*: the jdk used to run a deployment task.
- *Additional classpath*: weblogic.jar file path loaded for deployment. If no value is set but the WL_HOME environment variable is found, the plugin will try to load %WL_HOME%/server/lib/weblogic.jar.
Depending on the weblogic version the weblogic.jar should be replaced by :

- for WLS 10.3 /path/to/jar/wfullclient.jar
- for WLS 12.1 /path/to/jar/wlthint3client.jar.



Useful Information

If you use authentication mode, you should replace in WLS 12 wfullclient.jar by wlthint3client.jar. If you have to add many dependencies you can append them like that /path/to/wfullclient.jar:/path/to/wlthint3client.jar. The separator ':' is platform dependent.

see <https://docs.oracle.com/middleware/1213/wls/SACLT/basics.htm#SACLT125>



Oracle Information

"The WebLogic full client, wfullclient.jar, is deprecated as of WebLogic Server 12.1.3 and may be removed in a future release. Oracle recommends using the WebLogic Thin T3 client or other appropriate client depending on your environment. For more information on WebLogic client types, see WebLogic Server Client Types and Features."

- *Java Options to use*: the java options used by the java process when invoking the weblogic library. By default it will be -Xms256M -Xmx256M.
- *Excluding artifact name list*: the built artifacts name pattern excluded from post build deployment action.
- *Disable*: If checked, the plugin execution is disabled.
- *Configuration File*: xml file path containing deployments targets. It has to be [http://host:port/plugin/weblogic-deployer-plugin/defaultConfig/plugin-configuration.xsd] schema compliant. No validation currently performed. The file can be added through a local path (/usr/users/.../plugin-configuration.xml) or a remote URL (<http://xxxxxxx/plugin-configuration.xml>);

configuration.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="http://org.jenkinsci.plugins/WeblogicDeploymentPlugin"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://org.jenkinsci.plugins/WeblogicDeploymentPlugin plugin-configuration.xsd">
<weblogic-targets>
  <weblogic-target>
    <name>my_managed_server1</name>
    <host>_weblogic.admin.server.host_</host>
    <port>_weblogic.admin.server.port_</port>
    <login>_weblogic.admin.server.login_</login>
    <password>_weblogic.admin.server.password_</password>
    <authMode>BY_KEY</authMode>
    <userconfigfile>P:\Documents\perso\dev\wls-deploy-plugin\wls_userConfig.properties<
/userconfigfile>
    <userkeyfile>P:\Documents\perso\dev\wls-deploy-plugin\wls_userConfig.key</userkeyfile>
  </weblogic-target>
  <weblogic-target>
    <name>my_managed_server2</name>
    <host>_weblogic.admin2.server.host_</host>
    <port>_weblogic.admin2.server.port_</port>
    <login>_weblogic.admin2.server.login_</login>
    <password>_weblogic.admin2.server.password_</password>
    <authMode>BY_LOGIN</authMode>
  </weblogic-target>
  <weblogic-target>
    <name>my_managed_server3</name>
    <host>_weblogic.admin.server.host_</host>
    <port>_weblogic.admin.server.port_</port>
    <login>_weblogic.admin.server.login_</login>
    <password>_weblogic.admin.server.password_</password>
    <!-- used to transfer by FTP the component flagged as library to the WL machine -->
    <ftpHost>_weblogic.remote.host_</ftpHost>
    <ftpUser>_weblogic.remote.user_</ftpUser>
    <ftpPasswrdr>_weblogic.remote.password_</ftpPasswrdr>
    <remoteDir>/remote/path/used/to/transfer/libraries</remoteDir>
  </weblogic-target>
</weblogic-targets>
</config>
```



Library Deployment Tip

The ftp parameters are used to deploy libraries. When you checked your artifact deployment as a library deployment, the plugin connects to the ftpHost and transfers the artifact into the remoteDir directory. Basically, the ftpHost value should be the same as the host value.

Authentication mode

There are 2 kinds of authentication :

- BY_KEY : uses a keystore file and a config file
- BY_LOGIN : uses a login/password (this is the default authentication mode).

Since 2.3, the plugin also supports authentication by keystore. To use it, set authMode with BY_KEY value then set userconfigfile/userkeyfile elements. If you forget to specify authMode with BY_KEY value, the plugin uses the default authentication mode BY_LOGIN.

Job Configuration

If the plugin is not configured

If no configuration has been set, the plugin will display an error message

Analyse des dépendances
 Déployer les artefacts dans le repository Maven
 Additional test report features
 Record fingerprints of Maven 3 artifacts
 Archive Maven 3 artifacts
 Déployer l'artefact sur une cible Weblogic
 Sonar
 Emotional Hudson

**Avant toute utilisation, le plugin doit être configuré
Veuillez le faire dans la configuration système.**

If the plugin is already configured

If the plugin is already configured, Jenkins will display the job plugin configuration form.

Post-build Actions

Deploy the artifact to any Weblogic environments
Deployment tasks list

Deploy the artifact to any Weblogic environments
Task name: [text field]
environment: MACHINE_1 (localhost:7005) [dropdown]
name: toto1 [text field]
Base directory where the resource to deploy can be found: P:\Documents\workspace-sta\artefact-ear-app\target\ [text field]
Built resource to deploy: {.*}myEarName(.* [text field]
targets: AdminServer [text field]
weblogic libraries:
Delete the deployment task [button]

Deploy the artifact to any Weblogic environments
Task name: tache n°2 [text field]
environment: DEV96 (lhdsvdev:51200) [dropdown]
name: toto2 [text field]
Base directory where the resource to deploy can be found: [text field]
Built resource to deploy: *nothing*.war [text field]
targets: AdminServer [text field]
weblogic libraries:
Delete the deployment task [button]

Deploy the artifact to any Weblogic environments
Task name: [text field]
environment: DEVA2 (lhdsvdev:32100) [dropdown]
name: [text field]
Base directory where the resource to deploy can be found: P:\Documents\workspace-sta\artefact-ear-app\ [text field]
Built resource to deploy: {.*}myEarName(.* [text field]
targets: AdminServer [text field]
weblogic libraries:
Delete the deployment task [button]

Add a deployment task [button]

Build failed when deployment failed:
Force deployment tasks stop on the first failure:
Deployment policy: Non specified
Legacy code started this job. No cause information is available
Built by user
Built by remote host
Deploying only when sources changes:
Projects deployment Dependencies: [text field]
Multiple projects can be specified like abc.def. If blank, control is disabled.

- **Deployment task list** : The list of deployments to run. For each task, some of parameters has to be set :
 - **Task name (optional)** : The name of the task which will be used to identify it.
 - **environment** : the targeted weblogic environment. Can be customized.
 - **name** : the name used by weblogic to display component.
 - **Base directory where the resource can be found**. If this field is left blank, the plugin will lookup under workspace directory.
 - **Built resource to deploy** : a regular expression to filter resource which will be deployed.
 - **targets** : the weblogic target (can be a managed server instance, a cluster either the AdminServer). By default the plugin will use the AdminServer as the target. Many targets can be defined and will be separated by un comma symbol (e.g AdminServer, ManagedServer1, Cluster2). Can be customized.
 - **weblogic libraries** : If checked, the generated artifact will be deployed as a library component.
 - **advanced configuration** : See the chapter below Job Advanced configuration.
- **Build failed when deployment failed** : If checked, The current build will be flagged as a failed one if the deployment task failed.
- **Force deployment tasks stop on first failure** : Useful if you have any task configured and you want to stop the next deployments tasks where an error occurs.
- **Deployment policy** : Configure which cause(s) will trigger the deployment action. If 'None specified' is selected, the deployment occurs each time the job builds. If another ones are selected, the deployment action only occurs when the build was triggered by a chosen cause.
- **Deploying only when sources changes** : If checked, the deployment action only occurs when the build trigger by a parametrized cause AND the SCM step detects any changes.

- *Projects deployment Dependencies* : (experimental) specifies a dependency with others job deploying some artifacts.

Customization of parameters

Some of parameters can also be tokenized with environment variables or even build parameters. Here are these parameters :

- *environment*: Instead of setting a fixed value into the configuration file (e.g localhost:7000) you can set a variable like `${WL_HOST}:${WL_PORT}`
- *targets*: Instead of setting a fixed value (e.g AdminServer) you can set a variable like `$(ENV_TARGET)` or `$TARGET`.
- *Base directory where the resource can be found*: you can set a variable like `$(WORKSPACE)`.
- *name*: The name parameter can be tokenized (e.g my-war-`$(MY_VERSION)`)

Ressource générée à déployer	<code>(.*)\.war</code>
Targets	<code>\$(ENV_TARGET)</code>
Plans de déploiement WebLogic	

Use the build parameter plugin or add some environment variable to set these values

Ce build a des paramètres

Text Parameter

Name

Default Value

Job Advanced configuration

It's possible to set advanced options like stage mode and command line. It helps user with special needs to override the default behaviour of the plugin.

JDK Nom	<input type="text" value="system"/>
JAVA_HOME	<input type="text" value="P:\Outils\Java\jdk1.6.0_43\jre"/>
Configuration WebLogic Stage Mode	<input type="text" value="bydefault"/>
Commande WebLogic Ligne de commande	<pre>-debug -remote -verbose -name {wl.deployment_name} -source {wl.source} - targets AdminServer -adminurl http://{wl.host}:{wl.port} -user {wl.login} - password {wl.password} -deploy; -debug -remote -verbose -name prevalo-webapp -targets AdminServer -adminurl t3://localhost:33100 -userconfigfile P:\Documents\perso\dev\wls-deploy-</pre>

- *Stage mode* : stage, nostage or external stage
- *protocol* : t3, t3s, http or https
- *Command line* : The command which will be run by the deployer instead of the default ones (undeploy/deploy)

Stage mode

Stage mode define how the artifact will be deployed. By default the plugin use the stage mode 'stage'. Please refer to the weblogic documentation for further information.

Override deployment execution

For advanced users, it can be useful to define the commands that the deployer has to run instead of the default ones, To enable this feature, the user will set different command lines into the 'Command line' field of Job advanced configuration.

These commands has to be ended by the ';' separator. The following plugin token can be added to the command line :

- `{wl.deployment_name}` : the 'Name' parameter value
- `{wl.source}` : the resource to deploy (figured out by the plugin)

- {wl.targets} : the 'Targets' parameter value
- {wl.host} : the host part of the 'Environment' parameter
- {wl.port} : the port part of the 'Environment' parameter
- {wl.deploymentPlan} : the path of the deployment plan of the 'Environment' parameter
- {wl.login} : the login to authenticate to the server (provided by the configuration file)
- {wl.password} : the password to authenticate to the server (provided by the configuration file)
- {wl.user_configfile} : the Deployer configuration file (defined in the configuration file)
- {wl.user_keyfile} : the private key to authenticate (provided by the configuration file)

Note : only the arguments and parameters can be specified. The java executable used to run the command will be figure out by the plugin with the same algorithm than a default execution.

For instance : the user who prefers run deployment with 'distribute' instead of deploy could be set the following command lines:

command line parameter

```
-adminurl http://{wl.host}:{wl.port} -user {wl.login} -password {wl.password} -debug -remote -verbose -name {wl.deployment_name} -source {wl.source} -targets {wl.targets} -cancel;
-adminurl http://{wl.host}:{wl.port} -user {wl.login} -password {wl.password} -name {wl.deployment_name} -source {wl.source} -targets {wl.targets} -distribute
```



Be Careful

Sometimes source path contains special chars like \$.To ensure that the plugin well handle the path you can surround the token {wl.source} with quotes like this :

command line parameter

```
-adminurl http://{wl.host}:{wl.port} -user {wl.login} -password {wl.password} -debug -remote -verbose -name {wl.deployment_name} -source "{wl.source}" -targets {wl.targets} -cancel;
```

NB : Environment variables and build parameters can also be used.

User Guide

Build status

The build status will be modified according the following rules :

- the status is set to SUCCESS when :
 - the deployment task is successfully achieved
 - the checkbox "Mark build unstable if deployment was not attempted" is not checked whatever the result of pre-requisite step.
- If the checkbox "Mark build unstable if deployment was not attempted" is checked, the status is set to UNSTABLE when :
 - at least one of the verification of the deployment task failed (PLUGIN_DISABLED, OTHER_TRIGGER_CAUSE, NO_CHANGES, UNSATISFIED_DEPENDENCIES, PREVIOUS_STATUS_BUILD_FAILED).
 - the deployment task failed and the *Build failed when deployment failed* parameter is not checked.
- status is set to FAILED when the deployment task failed and the *Build failed when deployment failed* parameter is checked.

Check a build deployment status

On the build dashboard a new icon will appear. It displays the deployment status

Jenkins > free-style-project > #38

[Back to Project](#)

[Status](#)

[Changes](#)

[Console Output](#)

[Edit Build Information](#)

[Delete Build](#)

[Weblogic deployments](#)

[Previous Build](#)

Logger Console

 **Build #38 (Mar 4, 2013 6:04:45 AM)**

 No changes.

 Started by anonymous user

 [Deployments Statuses](#)

-  myEarName.ear#MACHINE_1
-  tache n°2#DEV96
-  myEarName.ear#DEVA2

[Help us localize this page](#)

Check the last successful deployment on main job dashboard

When you click on your job, a dashboard show a new icon with the date of the last successful deployment. If no deployment occurred or always failed. The date will be replaced by 'N/A' label.

Jenkins > free-style-project

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

Build History [\(trend\)](#)

	#38	Mar 4, 2013 6:04:45 AM
	#37	Mar 4, 2013 5:11:51 AM
	#36	Mar 4, 2013 3:30:26 AM
	#35	Mar 4, 2013 3:05:40 AM

Project free-style-project

 [Workspace](#)

 [Recent Changes](#)

  Last successful deployment(s) : Mon Mar 04 03:30:26 CET 2013 (build number #36)

Permalinks

- [Last build \(#38\), 4 hr 8 min ago](#)

View deployment Logs

You can view the deployment logs of each task. Simply click on the 'Deployments Statuses' link or 'Weblogic deployments' link. They will appear one by one in the deployment page.

Jenkins > free-style-project > #38 > Weblogic deployments

[Back to Project](#)
[Status](#)
[Changes](#)
[Console Output](#)
[Edit Build Information](#)
[Delete Build](#)
Weblogic deployments
[Previous Build](#)

Weblogic deployments logs

Time : Mon Mar 04 06:04:45 CET 2013
Resource : myEarName.ear
Machine : MACHINE_1
Targets : AdminServer

Status : 

```
----- ARTIFACT UNDEPLOYMENT -----
weblogic.Deployer invoked with options: -debug -remote -verbose -noexit -name totol -targets AdminServer -adminurl t3://localho
[WebLogicDeploymentManagerImpl.():103] : Constructing DeploymentManager for J2EE version V1_4 deployments
[WebLogicDeploymentManagerImpl.getNewConnection():146] : Connecting to admin server at localhost:7006, as user weblogic
[ServerConnectionImpl.getEnvironment():28] : setting environment
[ServerConnectionImpl.getEnvironment():29] : getting context using t3://localhost:7006
[ServerConnectionImpl.close():334] : Closing DM connection
[ServerConnectionImpl.close():354] : Unregistered all listeners
weblogic.deploy.api.tools.deployer.DeployerException: Unable to connect to 't3://localhost:7006': Destination unreachable; nested
java.net.ConnectException: Connection refused: connect; No available router to destination. Ensure the url represents a
protocol, tunneling must be enabled on the admin server.
    at weblogic.deploy.api.tools.deployer.Jsr88Operation.connect(Jsr88Operation.java:314)
    at weblogic.deploy.api.tools.deployer.Deployer.perform(Deployer.java:137)
    at weblogic.deploy.api.tools.deployer.Deployer.runBody(Deployer.java:88)
    at weblogic.utils.compiler.Tool.run(Tool.java:169)
    at weblogic.utils.compiler.Tool.run(Tool.java:118)
    at weblogic.Deployer.run(Deployer.java:70)
    at weblogic.Deployer.main(Deployer.java:54)
Caused by: javax.enterprise.deploy.spi.exceptions.DeploymentManagerCreationException
    at weblogic.deploy.api.spi.deploy.WebLogicDeploymentManagerImpl.(WebLogicDeploymentManagerImpl.java:121)
    at weblogic.deploy.api.spi.factories.internal.DeploymentFactoryImpl.getDeploymentManager(DeploymentFactoryImpl.java:84)
    at weblogic.deploy.api.tools.SessionHelper.getDeploymentManager(SessionHelper.java:432)
    at weblogic.deploy.api.tools.deployer.Jsr88Operation.connect(Jsr88Operation.java:302)
    ... 6 more
Caused by: weblogic.deploy.api.spi.exceptions.ServerConnectionException
    at weblogic.deploy.api.spi.deploy.internal.ServerConnectionImpl.init(ServerConnectionImpl.java:143)
    at weblogic.deploy.api.spi.deploy.WebLogicDeploymentManagerImpl.getNewConnection(WebLogicDeploymentManagerImpl.java:148)
    at weblogic.deploy.api.spi.deploy.WebLogicDeploymentManagerImpl.(WebLogicDeploymentManagerImpl.java:118)
```

Logger Console

Disabling deployment task at runtime

A deployment task can be skipped during a job build by setting an env variable or a build parameter following this naming rule (DEPLOY_<task_name>_SKIP) to true.

For instance, if your task names 'task1' you have the possibility to skip its execution by setting a variable named DEPLOY_TASK1_SKIP to true (It could be a boolean build parameter).

Master/slave architecture support

The latest version of the plugin (4.0) supports master/slave architecture. That is to say, the plugin automatically copy weblogic libraries and any kind of configuration to the remote node. Be sure that the right JDK used by the plugin is correctly installed and declared in the node configuration.

Note master/slave architecture

The plugin doesn't copy userkeyfile nor userconfigfile to the remote node. This enhancement will be done in the next version.

Release Notes

Version 4.1 (September 03, 2017)

[JENKINS-45044](#) - Getting issue details...

Version 4.0 (August 21, 2017)

Master-Slave architecture support ([JENKINS-44599](#) - Getting issue details...

[JENKINS-36215](#) - Getting issue details...

Note : the copy of userkeyfile and userconfigfile on the remote node is not currently supported.

Merge pull request [#8](#) from mustafau/simplify-global-configuration

[JENKINS-45097](#) - Getting issue details... [STATUS](#)

[JENKINS-36524](#) - Getting issue details... [STATUS](#)

[JENKINS-37077](#) - Getting issue details... [STATUS](#)

Version 3.7 (June 21, 2017)

[JENKINS-42802](#) - Getting issue details... [STATUS](#)

Upgrade to 1.580.1 jenkins version

Minor enhancements (simply invoke join instead of start + join, using absolute path of jdk home)

Version 3.6 (Oct 25, 2016)

[JENKINS-39029](#) Fixed NPE for job configured with the plugin version older than 3.5

[JENKINS-38218](#) Support multiple files validation for extraClasspath field.

Version 3.5 (Sep 04, 2016)

[JENKINS-32392](#) Added the ability to choose the protocol to use during the deployment.

Refactoring deployment policy. LegacyCodeCause and UserCause no more supported.(thanks to mustafa ulu)

Version 3.4 (May 26, 2016)

[JENKINS-19962](#) Added the ability to declare a job successful when pre-requisite step failed

[JENKINS-35161](#) variable can be used for the Deployment name value

UserCause is now flagged deprecated in the triggers list

Added TimerTrigger to the list of triggers

Version 3.3 (August 31, 2015)

[JENKINS-26992](#) baseDirectory not well replaced

Removed unused org.codehaus.plexus dependency

Version 3.1 (March 31, 2015)

Removed org.codehaus.plexus.FileUtils dependency

Version 3.0 (March 28, 2015)

[JENKINS-26992](#) Supported baseDirectory variable expansion

[JENKINS-26366](#) Implemented Promoted Builds plugin support

[JENKINS-23700](#) By default the plugin searches under workspace directory



Be Careful

This version is not backward compatible. If you used previous version with maven project don't forget to update the maven jobs configuration (fill up base directory which is no more computed).

Version 2.13 (February 16, 2015)

[JENKINS-20351](#) Implemented the recursive search when no base directory is set in freestyle job configuration.

Fixed [JENKINS-26901](#) Not possible to run freestyle job with non-set baseDirectory.

Version 2.12 (December 17, 2014)

[JENKINS-24789](#) Added the possibility to skip a task deployment during a build by setting a build parameter or an env variable.

Version 2.11 (December 16, 2014)

Fixed regression due to an unexpected commit 🚧

Version 2.10 (December 6, 2014)

Fixing bug [JENKINS-23701](#) : Added better control on invalid directory parameter set.
French translation missing
Fixing wrong javadoc
Fixing wrong icon URL.

Version 2.9.1 (December 20, 2013)

Fixing bug [JENKINS-20970](#) : deployment failed due to an unexpected stream closed exception.

Version 2.9 (November 30, 2013)

Handled exception when configuration file uploaded is incorrect.
Added error log on jenkins project selector bean.

Version 2.8 (October 06, 2013)

Fixing bug [JENKINS-16007](#)

Version 2.7 (October 04, 2013)

Implementing [JENKINS-18685](#) Added support for customization of deployment targets/environment through build parameters.

Version 2.6 (September 16, 2013)

Implementing [JENKINS-19237](#) Added deployment plan support
Evolution [JENKINS-19016](#) Modified behavior when the step of pre-requisite check failed.

Version 2.5 (July 27, 2013)

Implementing [JENKINS-18882](#) Replaced weblogic icon
Fixing bug [JENKINS-18940](#) token {wl.source} never replaced in command line

Version 2.4 (July 24, 2013)

Implementing [JENKINS-17300](#) and [JENKINS-18812](#) : Added new feature which allow the advanced user to define his own command to run
Fixing bug [JENKINS-18881](#) : NPE when the user viewing deployment result of a skipped execution
Fixing bug [JENKINS-18883](#) : cron value doesn't appear in 'Deploy periodically' field

Version 2.3 (June 26, 2013)

Implementing [JENKINS-18467](#) : stage mode attribute mishandled in job advanced configuration.
Implementing [JENKINS-17879](#) : Supports weblogic authentication by keystore.

Version 2.2 (May 04, 2013)

Implementing [JENKINS-17300](#) : Supports stage mode choice in job configuration.

Version 2.1 (May 03, 2013)

Implementing [JENKINS-17301](#) : Supports jdk configuration per job.

Version 2.0 (Mar 04, 2013)

Implementing [JENKINS-16808](#) : Supports multi-deployment feature.
Implementing [JENKINS-16699](#) : Searched resource pattern or name is traced in the job log



Be Careful

This version is not backward compatible.

Version 1.4 (Feb 11, 2013)

Implementing [JENKINS-15696](#) : Supports freestyle project.

Version 1.3 (Aug 13, 2012)

Implementing [JENKINS-14763](#) : English translation. The English language is the default plugin language.

Version 1.2 (Aug 2, 2012)

Implementing [JENKINS-14646](#) : The old java class or resources prefixed by HudsonXXXX has been refactored to XXXX.

Implementing [JENKINS-14626](#) : As of now, it's possible to filter the artifact name to deploy by regex defined in the job configuration. (Useful for maven multi-module project which generate many artifacts).

Version 1.1 (May 5, 2012)

Forked from Hudson version.

Added JDK selection on global configuration.

Removed task abortion if no compatibility detected between the JDK and the weblogic.Deployer API (As of now, It displays a warning).

Fixed translation and spelling errors.

Added WebLogic Server 10.3 support.