IBM WebSphere

> ⊘ **Warning**
>
> Running Jenkins on WebSphere 7.x requires WebSphere 7.0.0.7 or above. Jenkins will NOT work on WebSphere versions 7.0.0.0 - 7.0.0.5. See issue 3274 in the #Known Issues section for further information!

> ⚠   • The following installation was tested with WAS 6.1 ND, WAS 7.0 ND, WAS 8.0 Developer Edition and WAS 8.5 Developer Edition
>   • Some links in Jenkins are broken in WebSphere, but there is usually either another way to access them or it is intuitive to correct the URL in the address bar. It is not critical problem, thus no time to investigate it further...

Table of contents

Requirements

- The following **Hudson versions** run on WAS:
    - Hudson **1.238 to 1.126** (see hudson: issue 1680).
    - Hudson/Jenkins **1.291 or greater** (see hudson: issue 3069).
    - Other versions won't run on WAS.

- **WAS versions:**
    - Previously mentioned Jenkins versions should all run on WAS 6.1.
    - With WAS 7.0 (7.0.0.0, 7.0.0.1 and 7.0.0.3), a WebSphere class loader issue will prevent Hudson 1.291 or greater to start and use all its internal components (see hudson: issue 3274). The bug has been identified and fixed by IBM, but the fix is not yet publicly available. *This page will be updated when it's done.*

See also the #Known Issues section.

Installation procedure

There are two ways to install Jenkins on WAS:

- Manually through the Admin console;
- Or automated through the WAS scripting interface.

Admin console installation

> ⚠   The following works fine on WAS 6.1, WAS 7.0, WAS 8.0 and WAS 8.5.

The installation instructions are simple as for each WAR Module.

⚠️

- Start the installation wizard under Applications - Install new application.
- Select jenkins.war and enter the context root for the application - for example jenkins

After finishing the wizard click the installed application (jenkins_war for example) and perform the following steps to change the classloader mode:

- Click Manage Modules
- Click Hudson / Jenkins
- From the "Class loader order" drop-down select "Classes loaded with application class loader first" ("Classes loaded with local class loader first" V8.5)
- Save the changes
- Restart the server - this is necessary to apply the changed class loader mode. You need to restart only the server onto which jenkins is installed. (V8.5: no need to restart the server, just start the Jenkins app).

If you try to install Jenkins 1.450 on WAS 8, you may face some exceptions mentioned in JENKINS-12334. This may occur, if an SLF4J implementation is missing.
Since log4j is already on the Classpath, you might want to download the slf4j-log4j bridge and copy that JAR into the WEB-INF/lib folder of Jenkins.

Installing through wsadmin

ⓘ Likely needs to be edited to work with Jenkins

⚠️ If you plan to install Hudson on WAS 7, use the Admin console as the script below won't work (due to some changes in IBM codebase which requires some fixing).

A jacl script called hudson.jacl and configuration file config.properties are provided to install and configure hudson on WAS 6.1 ND.

Before running hudson.jacl make the necessary changes in the config.properties file - adjust the server envoronment, the context root, the virtual host mapping and optional the hudson home directory. After that run following command from the deployment manager bin directory:

```
wsadmin(.sh) -f <path_to_hudson.jacl>/hudson.jacl <path_to_config.properties>/config.properties <command>
<path_to_hudson.war>/hudson.war
```

 where the <command> is:

- deploy - use it for first time installation. it performs installation only
- redeploy - use it to upgrade hudson. It uninstalls previously installed hudson instance and installs the new version from the hudson.war
- undeploy - use to uninstall hudson. In this case the last parameter is not necessary.

The paths must be provided in UNIX style even on Windows - d:/test/hudson/hudson.war instead of d:\test\hudson\hudson.war

After successfully installation the server must be restarted manually - the script doesn't provide this functionality at the moment, but the stopServer and startServer commands can be used for this purpose.

If security is enabled on the server either provide -user and -password arguments to the command or adjust the soap.client.props file with the login information in order to be able to connect to the server's admin interface.

The script is not tested with the base server edition, but should work on it - the only change is to disable the node synchronization in the saveConfigAndSync procedure - the line to comment can be found in hudson.jacl - search for BASE or for saveConfigAndSync.

Installation using Jython Scripts

⚠️ These scripts make heavy use of 'wsadminlib.py', a large monolithic jython library for WebSphere Administration.
The original scripts can be found here Sample Scripts for WebSphere Application Server.
The 'wsadminlib.py' file that is included here has been modified to include some extra functions that were not present in the original.

There are four script files that have been provided. They have been tested under WAS 6.1 and AIX. Feedback is sought for WAS 7.0 and other platforms.

- createHudsonServer.py
- deleteHudsonServer.py
- installHudson.py
- wsadminlib.py

They are executed using the wsadmin.sh tool:

```
/was61/WebSphere/AppServer/bin/wsadmin.sh -lang jython -f <<<script file>>>
```

Attachments

- createHudsonServer.py - wsadmin script to create the Hudson Server.
- deleteHudsonServer.py - wsadmin script to delete the Hudson Server.
- installHudson.py - wsadmin script to install the Hudson Application into the Hudson Server.
- wsadminlib.py - The wsadmin Über library.

Additional Setup

Please remember that a 64-bit JVM takes about 60% more heap space than a 32-bit JVM does. Size your VM, if using 64-bit, accordingly. I'm using 1Gb (-Xmx1024M) heap, as 512Mb was causing regular heap dumps due to out of memory errors.

Notes

This is simply how I do it. It does require additional machine resources, as it takes a complete JVM, but I prefer to create a server per application which provides for better application isolation and tuning capabilities.

CVS SSH authentication

For CVS ssh authentification following environment entry is required in the application server: CVS_RSH=ssh. This can be set in the admin console in:

Application Servers / <server> / Java and Process Management / Process Definition / Environment Entries

Upgrading

Changing the WAR file to reference the newest dom4j library (see hudson: issue 1680):

- Unpack hudson.war and navigate to WEB-INF/lib directory.

- Download dom4j-1.6.1.jar and copy it to the WEB-INF/lib directory.

- Delete dom4j-1.5.jar

- Zip the war file again.

> ⚠️ Do not remove dom4j-1.6.1-hudson-1.jar!

Following steps should be done upgrading running hudson instance:
- Backup the current configuration

```
tar -cvf hudsonXX.tar <hudson_home_dir>
```

* Redeploy hudson with the provided scripts as follows:

Edit config.properties to the environment and run (see Installing through wsadmin above for further information):

```
wsadmin.sh -f hudson.jacl config.properties redeploy hudson.war
```

The above command assumes that wsadmin.sh is in PATH and hudson.jacl, config.properties and hudson.war are in the current dir.

- Restart the application server.

Setting up WAS for the update center to work

As of version 1.23x? hudson provides update center. In order to be able to use it with the WebSphere's JDK a trusted certificate must be imported in the server's keystore. To acomplish this following steps are necessary:

1. Download the RapidSSL root certificate from here .
2. Install the certificate using #keytool (command-line mode) or IBM Key Management (GUI mode).
3. After importing the certificate the application server must be restarted. After that it should be possible to upgrade your plugins from the Update Center in hudson (don't forget to set the proxy if you have one).

Importing the certificate using the keytool command (command-line mode)

```
keytool -import -trustcacerts -file rapidssl_01.cer -keystore <server_trust.p12> -storepass WebAS -storetype
PKCS12 -alias  rapidssl
```

where as:

- <server_trust.p12> points to the trust.p12 file of the server on which hudson is running. It is found in `<WAS_Install_Dir>/profiles` `/<profile>/config/cells/<cellname>/node/<nodename>`.
- The command uses the default WebSphere password for the keystore file (*WebAS*) - if it has been changed change the command accordingly.

Issuing the command brings a message that the certificate is found in the system wide keystore - confirm with yes that it should be updated.

Eg:

```
$ cd /tmp
$ wget http://www.rapidssl.com/cps/rapidssl_01.cer
<snip>
$ /was61/WebSphere/AppServer/java/jre/bin/keytool -import -trustcacerts -file rapidssl_01.cer -keystore /was61
/WebSphere/AppServer/profiles/AppSrv01/config/cells/localhostCell01/nodes/localhostNode01/trust.p12 -storepass
WebAS -storetype PKCS12 -alias  rapidssl
Certificate already exists in system-wide CA keystore under alias <equifaxsecureca>
Do you still want to add it to your own keystore? [no]:  yes
Certificate was added to keystore
$
```

Importing the certificate using IBM Key Management (GUI mode)

- Start `ikeyman.bat/ikeyman.sh` located in `<WAS_Install_Dir>/bin`.
- Go to *Key Database File > Open*.
- In the *Open* window, set the *Key database type* to *PKCS12 _, then use the _Browse* button to locate the `trust.p12` file (it's in `<WAS_Install_ Dir>/profiles/<profile>/config/cells/<cellname>/node/<nodename>`).
- Enter the password for this keystore (the default value is *WebAS*).
- In the *Key database content*, select the *Signer Certificates item*.
- Click on the *Add...* button, locate the certificate you've previously downloaded, and click on *OK*.
- When asked to *Enter a Label*, set the value to *rapidssl* and click on *OK*.
- Go to *Key Database File > Save As...* and overwrite the existing `trust.p12` file (it's a good idea to make a copy of it before...).
- When asked for a password, set it to the previous value (again, *WebAS* is the default one).
- Exit from the tool.

Known Issues

Following table tries to summarize the known issues. If there is a fix from IBM its PK Id will be referenced.

Issue #	Hudson - Affects	Hudson - Fixed In	WAS - Affects	WAS - Fixed In	PK #
JENKINS-1680	1.213	1.237	All		
JENKINS-3069	1.271	1.291	All		
JENKINS-3274	> 1.285		7.0.0.x	7.0.0.7	PK86109

JIRA Plugin

I'm experiencing VerifyError on WAS 7.0 trying to use the provided JIRA plugin due to dependencies on the Stax API included in the stax-api-1.0.1.jar. I've simply removed the jar file from hudson's lib directory and it works ok for me. For further details see this thread.