

# **WebSphere 5.0 Installation HOWTO**

---

\$Revision: 1.4 \$

**Blaine Simpson. [blaine.simpson@admc.com](mailto:blaine.simpson@admc.com)**

---

## Table of Contents

Assumptions .....	2
Introduction .....	3
Installing WAS Software and the Main Instance .....	4
Installing an Alternate Instance .....	7
Setting up Oracle Data Sources .....	10
Important Files .....	11

Author: Blaine Simpson. [blaine.simpson@admc.com](mailto:blaine.simpson@admc.com)

\$Revision: 1.4 \$

The software which is referred to in this document and which is hosted on the [admc.com](http://admc.com) web site, is distributed for free according to the GNU Public License.

**All-in-one HTML**

<http://admc.com/blaine/howtos/ws50install/ws50install-all.html>  
([ws50install-all.html](#))

**Split-up HTML**

<http://admc.com/blaine/howtos/ws50install/index.html> ([index.html](#))

**Info**

<http://admc.com/blaine/howtos/ws50install/ws50install.info> ([ws50install.info](#))  
(useful if you are on UNIX and want to view this in a terminal window instead of with a web browser, and without losing the linking features).

## Assumptions

We want to run with iPlanet web server. Since WebSphere on Linux does not support iPlanet web server, we install using the built-in http service on Linux, and with iPlanet web server support on Solaris. Therefore, we do not install the IHS Apache web server.

We use the message queue server embedded in WAS. Therefore, we do not install the standalone MQ server.

We use https to access the Administrative Console.

We set up the main installation to run as a non-root user, but use this instance only for testing (because if this instance gets messed up, it can be become permanently disabled). We will create an alternative instance, and it is this instance which we will use for our real development work. The procedure can be repeated to make as many alternative instances as desired. These instances can easily be re-created if anything gets messed up. WAS shares binaries and runtime images among instances wherever possible. Alternative instance(s) are run by non-root user(s). Each alternative instance can be run by a different user (this allows for multiple UNIX users to independently administer their own instance).

Example applications are installed for the main instance, but not for alternative instances.

## Introduction

**IMPORTANT: This document is largely superseded by the other installation HOWTO's.**

**This document is not currently being maintained. It is left here because it is the only place where I have documented the steps specifically needed for root setups (setting up OS IPC resources, message queue accounts, etc.).**

When I use variables like \$THIS or <THIS>, in general you have to type in the real values. (If you know how to use the shell, then you can figure out how to set and export variables so you don't have to type out the real values). Here are the variables that I use in this document.

'\$WAS\_HOME'

Main directory where you installed the WAS software.

'\$USER\_INSTALL\_ROOT'

Root directory of an alternative WAS instance.

'<MAIN\_OWNER>'

Unix user which will "own" the files under \$WAS\_HOME. (The default instance will run as this user).

'<INST\_OWNER>'

Unix user who will "own" the files under \$USER\_INSTALL\_ROOT. This alternative instance will run as this user.

'<INST>' Name (identifier) of the alternative instance.

'<NODE>' WAS's identifier for this host (but don't confuse this with "hostname", like the WAS docs often do).

'<IWS\_INST\_HOME>'

Home directory for your iPlanet Web Server instance. (If you use default settings, this will be something like /opt/iplanet/server4/https-SOMENAME.nextel.com).

Where I just give a URL, that means to use your browser to go to the given URL.

I suggest that you run iPlanet Web Server as a non-root user, such as "netscape" or "iplanet".

## Installing WAS Software and the Main Instance

For Solaris, I assume that your iPlanet web server is already installed and you have verified that it works. If you want my tips for installing and configuring IHS (IBM's Apache server) with WAS, email me.

Download the following files from the IBM web site or wherever.

1. ae\_offline\_infocenter\_en.zip
2. ibmwas5\_trial\_for\_\*.zip

(The former is available at

[ftp://ftp.software.ibm.com/software/websphere/info/appserv/v50/ae/ae\\_offline\\_infocenter\\_en.zip](ftp://ftp.software.ibm.com/software/websphere/info/appserv/v50/ae/ae_offline_infocenter_en.zip).

Make sure you get it because it is extremely difficult to install it afterwards).

Extract both of those files into an empty directory.

Become root.

**SOLARIS ONLY** Set the following kernel parameters in `/etc/system` and reboot.

```
set shmsys:shminfo_shmmax = 4294967295
set shmsys:shminfo_shmseg = 1024
set shmsys:shminfo_shmmni = 1024
set semsys:seminfo_semaem = 16384
set semsys:seminfo_semmni = 1024
set semsys:seminfo_semmmap = 1026
set semsys:seminfo_semmns = 16384
set semsys:seminfo_semmsl = 100
set semsys:seminfo_semopm = 100
set semsys:seminfo_semmnu = 2048
set semsys:seminfo_semume = 256
set msgsys:msginfo_msgmap = 1026
set msgsys:msginfo_msgmax = 4096
set rlim_fd_cur=1024
```

**Remove all core dump files from the `/var/sadm/pkg` directory.**

Create the group mqm. (Use different integer if this one already in use). (Skip if in Linux, since Linux will do this automatically when you run the first useradd command below).

```
groupadd -g 2001 mqm
```

Create the group mqbrkrs. (Use different integer if this one already in use).

```
groupadd -g 2002 mqbrkrs
```

Create the user mqm

```
useradd [-M] -d /dev/null -u 2001 -g 2001 -c Mqm -s /bin/false mqm
# Give -M only in Linux
```

If <MAIN\_OWNER> doesn't exist yet, then create that account, adding the user to the mqm and mqbrkrs shared groups.

```
groupadd -u 1601 <MAIN_OWNER>    # Not needed for Linux
useradd -d $WAS_HOME -c WebSphere -u 1601 -m -G mqm,mqbrkrs <MAIN_OWNER>
```

Create your <INST\_OWNER> account if it doesn't already exist. Now edit the /etc/group file and add root, <MAIN\_OWNER> and <INST\_OWNER> to the following groups. (This makes it work. IBM's instructions do not).

```
<MAIN_OWNER>
mqm
mqbrkrs
```

Log out and log back in as root (in order to pick up the new groups that you just added).

Cd to the directory into which you unzipped the files above and run ./install. You must run install as root since OS packages will be installed and because some mqm programs (at least) are made set-user.

1. Accept all defaults except where noted here.
2. Ignore warning about unsupported OS (if RH8.0)
3. Change paths. I like /usr/local/ws-5.0.
4. I recommend that you name your node "node" + your hostname with initial cap, like "nodeSaturn". This is to avoid confusing hostname with node name. This document hereafter uses <NODE> to refer to the node name.
5. For host name, you need to use the name of an address that will always be available locally. I suggest that you allocate a static IP addr just for this purpose. (Everything listens to IPADDR\_ANY by default. The problem is just that the queue client programs connect using this host name, so if it's not available the queue commands will fail).
6. At the end, "install" will start the "First Steps" gui. This is just a pretty wrapper to view release notes, run the verifier, etc. Just close it (because any Admin needs to know how to run the important stuff without this Gui, and the other stuff is useless).

Edit the file \$WAS\_HOME/bin/setupCmdLine.sh. Search for the "ulimit" command and append "2>&-" to it (this command fails for normal non-super users).

#### LINUX ONLY

Add "/usr/local/ibm/gsk5/lib" to /etc/ld.so.conf and run "ldconfig -X". If /usr/lib/libdb.so.3 doesn't exist, then make a sym link

```
ln -s libdb1.so.2 /usr/lib/libdb.so.3
```

(assuming that the former exists).

Chown the WAS software to <MAIN\_OWNER>.

```
chown -R <MAIN_OWNER>:<MAIN_OWNER> $WAS_HOME
```

Log in as <MAIN\_OWNER> and do the following actions as <MAIN\_OWNER>.

Run \$WAS\_HOME/bin/ivt.sh to verify the installation. (It should "fail to connect" and then start the app server).

Verify that the WAS admin console works.

```
http://localhost:9090/admin
```

(Log in with any user name. No password required.)

Verify that a sample application works.

```
http://localhost/snoop
```

Shut down the instance.

```
$WAS_HOME/bin/stopServer.sh server1
```

You are finished installing the software and the main instance. (There's no need to configure this instance any more since we won't be using it for anything but testing).

## Installing an Alternate Instance

You need to make up a name for the new instance. This document hereafter uses <INST> to refer to this name. Note that this is the "unqualified" instance name. The full instance name is <INST>\_<NODE>. You will need a range of 14 available consecutive (for sanity's sake) IP ports. (Since you're running as non-root, they must be > 1023). You also need to pick a \$USER\_INSTALL\_ROOT, the directory in which to install this instance (commonly under your personal home directory somewhere). This document calls the UNIX user who will "own" and "run" the new instance <INST\_OWNER>.

Chown the \$USER\_INSTALL\_ROOT to <MAIN\_OWNER>. (It won't stay this way. This is only needed to do the install). (N.b: This is <MAIN\_OWNER> here, not <INST\_OWNER>!)

You have to be <MAIN\_OWNER> to perform the following actions.

1. Put \$WAS\_HOME/bin in your search path (wsinstance.sh depends on this).
2. cd \$WAS\_HOME/bin/wsinstance.
3. cp portdef.props <NODE>\_<INST>\_portdef.props.

**N.b.**

1. **This filename uses an inverted instance name. The full instance name is <INST>\_<NODE>, but the props file uses <NODE>\_<INST>.**
2. **The WS docs incorrectly state the the props file uses the former, but it actually uses the latter.**
3. **(In general, always be careful of what order is required in different circumstances, because they do vary).**

Now edit the new file and change the port numbers to your 14 available ports. The order of the values in the .props file is arbitrary, but the order of the ports may not be, so, to be safe, assign port numbers in the following sequence (this sample uses the port sequence beginning at 4002). (I recommend that you edit a copy of portdef.props instead of what I supply here, in case IBM has changed anything in your version).

```
HTTPS_TRANSPORT_ADMIN=4002
HTTP_TRANSPORT_ADMIN=4003
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=4004
HTTP_TRANSPORT=4005      # Can connect directly to this
HTTPS_TRANSPORT=4006
INTERNAL_JMS_SERVER=4007
BOOTSTRAP_ADDRESS=4008
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=4009
SOAP_CONNECTOR_ADDRESS=4010
DRS_CLIENT_ADDRESS=4011
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=4012
JMSSERVER_QUEUED_ADDRESS=4013
JMSSERVER_DIRECT_ADDRESS=4014
ORB_LISTENER_ADDRESS=4015
```

Many of these ports are very important. The bootstrap port (BOOTSTRAP\_ADDRESS) must be supplied to the dumpNameSpace.sh command (with the

-port switch). HTTP\*\_TRANSPORT\_ADMIN are the Administrative console ports. (in this example, you'd use `https://localhost:4002/admin` for the SSL Admin server). HTTP\*\_TRANSPORT are the direct HTTP ports to use the App Server (as opposed to going through a standalone web server). The default ports for the default instance are listed at the bottom of `http://publib7b.boulder.ibm.com/wasinfo1/en/info/aes/ae/rins_portnumber.html`.

Run the command

```
./wsinstance.sh -name <INST> -path $USER_INSTALL_ROOT -host <NODE> -create
```

(N.b. you specify the node name, not the host name. The switch name and the WAS docs are wrong). Watch for error messages. (If you mess up, you must run `./wsinstance.sh -name <INST> -path $USER_INSTALL_ROOT -host <NODE> -delete` to remove the bad instance before recreating).

Become root.

Fix perms so that <INST\_OWNER> can access what he needs. \$WAS\_HOME and the following subdirs of \$WAS\_HOME must be rx-able to <INST\_OWNER>:

```
bin, java, properties, deploytool, config, lib, classes, null, samples, web.
```

And the same for `bin/setupCmdLine.sh`. (Most of these will probably be ok since the instance user is in the group <MAIN\_OWNER>, but that depends on the umask and env. when install was run). Chown \$USER\_INSTALL\_ROOT to <INST\_OWNER>.

```
chown -R <INST_OWNER>:<INST_OWNER> $USER_INSTALL_ROOT
```

Make sure that \$USER\_INSTALL\_ROOT/config and all parent directories rx-able to all (<MAIN\_OWNER> and the IWS user need access\*). Make sure that the UNIX user who IWS runs as\* has read access to the file \$USER\_INSTALL\_ROOT/config/cells/plugin-cfg.xml.

\* Ignore the IWS user stuff if you are on Linux or aren't setting up iPlanet web server.

Skip the next two paragraphs if you are on Linux or aren't setting up iPlanet web server.

Back up and edit <IWS\_INST\_HOME>/config/obj.conf. Search for the attribute "bootstrap.properties". Change the value of this attribute to \$USER\_INSTALL\_ROOT/config/cells/plugin-cfg.xml (If you have multiple iWS instances, or multiple virtual sites within one obj.conf file, you can just copy the 3 lines containing ("libs41\_http.so", ".../plugin-cfg.xml", and the following Service line) to the target obj.conf files or obj.conf virtual site blocks.

Change the WebSpherePluginConfig value to \$USER\_INSTALL\_ROOT/config/cells/plugin-cfg.xml. (If you have virtual sites, you can have different virtual sites using plugin-cfg.xml files from different WAS instances). Restart the web server

```
<IWS_INST_HOME>/restart
```

Become <INST\_OWNER> and test things out.

Export the variable \$WAS\_HOME

```
export WASROOT=$WAS_HOME # If you're running korn or bash shell
```

Source \$USER\_INSTALL\_ROOT/bin/setupCmdLine.sh

```
. $USER_INSTALL_ROOT/bin/setupCmdLine.sh
```

and run some commands, like

```
$WAS_HOME/bin/startServer.sh server1
http://localhost:4003 (where the number is the HTTP_TRANSPORT_ADMIN above).
dumpNameSpace.sh -port 4008 (where the number is the BOOTSTRAP_ADDRESS above).
```

(Note that the sample apps will not be available unless you install them. The `wsinstance.sh` command copies only the Admin Console application).

Our security strategy is to permit Admin Console access only via https on the localhost. If you are on another host, you can use VNC, X forwarding, or SSH X forwarding to run your Admin Console browser on the target server. You can always enter any username to access the Console. We do not turn on global security because that requires the instance to be run as root, or access to an external LDAP server. It would be nice if IBM let you set your own source IP list for access (like you can with Tomcat and pretty much every web server), but they don't. If you have iptables or other firewall software installed, it is easy and more flexible to use that than to run the Admin service on localhost as explained below.

Now disable services that we don't use (non-HTTPS Admin Console and HTTPS web access), and tighten up access. Stop WAS

```
$WAS_HOME/bin/stopServer.sh server1
```

Back up, then edit the following file.

```
$USER_INSTALL_ROOT/config/cells/<NODE>/nodes/<NODE>_<INST>/servers/server1/server.xml
```

Remove the `<transport... </transports>` stanzas which contain the following strings:

```
HTTPTransport_1, HTTPTransport_2, HTTPTransport_3
```

if you are using an external web server; and

```
HTTPTransport_2, HTTPTransport_3
```

if you are using only the built-in HTTP service. In the `HTTPTransport_4` stanza, change the value of `host` from `"` to `"localhost"`. Back up, then edit the following file.

```
$USER_INSTALL_ROOT/config/cells/<NODE>/virtualhosts.xml
```

Remove the lines containing the following strings.

```
HostAlias_1, HostAlias_3, HostAlias_4
```

if you are using an external web server; and

```
HostAlias_2, HostAlias_3, HostAlias_4).
```

if you are using only the built-in HTTP service. Start WAS back up.

```
$WAS_HOME/bin/startServer.sh server1
```

From here on, you must access the Admin Console at `https://localhost:4002/admin` from a local login. (Use the port number specified for `HTTPTransport_4` in the `server.xml` file, which you edited above.

You access the end-user interface through your external web server (if you have set one up), or at the `HTTP_TRANSPORT` port otherwise (see the file `$IWS_HOME/bin/wsinstance/<NODE>_<INST>_portdef.props`).

## Setting up Oracle Data Sources

(Just like all other operations in the Admin Console, after you make a set of changes, make sure you commit the changes by clicking a "Save" link, then the "Save" button.)

The instructions in the WebSphere docs for setting up shared libraries for DB drivers are wrong. They work for the first-level shared `libcijdkb9.so`, but not for the others. So, forget setting `$LIBPATH` or any other variables. Don't even waste your time setting the Native Library Path for the Provider. Just export `LD_LIBRARY_PATH=$ORACLE_HOME/lib` before starting WAS. (Alternatively, you could copy the 3 needed `.so`'s to a system lib location or add `$ORACLE_HOME/lib` as a system lib location). Use Admin Console to perform the following actions. Environment/WebSphere-Variables. Set

`ORACLE_JDBC_DRIVER_PATH` to the value of `$ORACLE_HOME/jdbc/lib`.

Resources/JDBC Providers

1. Add the Oracle JDBC oci8 Driver. (Use whichever scope you prefer, but be aware that this determines the scope of all DataSources you create using this Provider).
2. Make sure that Classpath is set to `${ORACLE_JDBC_DRIVER_PATH}/classes12.zip`.
3. Make sure that Implementation Classname is set to `oracle.jdbc.pool.OracleConnectionPoolDataSource`.

Restart WAS.

```
$WAS_HOME/bin/stopServer.sh server1; $WAS_HOME/bin/startServer.sh server1
```

Admin Console again.

Resources/JDBC Providers/Oracle JDBC oci8 Driver/DataSources

1. New. Set any name. JNDI name should begin with "jdbc/". If "Statement Cache Size" is empty, set it to "10". Leave everything else at default settings and hit "Apply".
2. Then hit "J2C Authentication Data Entries" (at bottom of the data source edit page) and create an alias for a name/password pair.
3. Edit the DataSource to add this alias for Component-managed or Container-managed Authentication Alias. (If you use the former, then your apps have to set "Res. Auth." to "Application"; if you use the latter, then your apps have to set "Res. Auth." to "Container-managed".
4. Hit "Custom Properties" (at bottom of the data source edit page). Click on the link "URL". Enter an Oracle oci8 URL like `jdbc:oracle:oci8:@keeper`.

## Important Files

You may want to put sym-links to these in a main directory like \$USER\_INSTALL\_ROOT or your home directory.

```
$USER_INSTALL_ROOT/config/cells/<NODE>/nodes/<NODE>_<INST>/servers/server1/server.xml ■  
$USER_INSTALL_ROOT/config/cells/<NODE>/virtualhosts.xml.  
$USER_INSTALL_ROOT/logs/server1/SystemErr.log  
$USER_INSTALL_ROOT/logs/server1/SystemOut.log
```

(Contrary to common sense, SystemOut.log has most of WAS's "error" messages). (Note that many files in \$USER\_INSTALL\_ROOT/logs are in binary format).