



JBoss 4.0.5 GA Installation

(Also includes Section for setting up as
JBoss as a Service on Windows and UNIX)

Simple Manual Install For:

Application

Server: JBoss

Java: JDK

Windows Installation and Service Setup:	3
Download and Installation	3
How to Setup JBoss to start as a Service on Windows 2003.....	3
Option 1. Use JBoss Native for Windows	3
Download.....	3
Install.....	3
Start.....	4
Stop.....	4
Restart.....	4
Uninstall.....	4
Customization	4
Option 2. Use the JavaServiceWrapper by Tanuki.....	5
UNIX Installation and Daemon Setup.	7
Download and Installation	7
Starting JBoss on boot with Linux.....	7
Create a user called "jboss".....	8
Create the script in /etc/rc.d/init.d.....	8
Using the included init Scripts (JBoss 4.0.1 and higher).....	10
create links	10
Using Tanuki Software's Java Service Wrapper.....	11

Windows Installation and Service Setup:

Download and Installation

- 1) Download JBoss from the JBoss web site
 - a. [JBoss 4.0.5 GA Installs](#)
 - i. [JBoss 4.0.5 Zip File](#)
 - b. [JBoss Quick Start PDF Guide](#)
- 2) Download JDK 1.5.03 or higher From [Sun](#) and install
- 3) Unzip this file to the C:\ Drive. This will create a folder C:\jboss-4.0.5.GA
- 4) Go to C:\jboss-4.0.5.GA\bin
- 5) Double click on run.bat
- 6) Once this stops processing open a web browser and go to [Http://<yourmachinename>:8080](http://<yourmachinename>:8080)

Note: This should show you the JBoss maintenance screen.

Note: Detailed installation instructions and trouble shooting tips can be found at [JBoss Web site](#). Be sure to install JBoss in a directory pathname that contains no spaces. Make sure no other applications are running on port 8080.

How to Setup JBoss to start as a Service on Windows 2003

Option 1. Use JBoss Native for Windows

You can use [JBossNative](#) : <http://labs.jboss.com/jbossweb/downloads/>. Download the appropriate version based on your system and go through the README-service.txt which lists down the steps for running JBoss as a service.

See: [Available Builds](#)

Using [JBoss Native](#) to run JBoss Application Server as a Windows Service

JBoss AS comes with Windows service executable as part of [JBossNative](#) that can run JBoss Application Server as service. The service executable jbosssvc.exe transforms the run.bat and shutdown.bat batch scripts to services. This means that any change made to those scripts will be used both in service and command line mode.

Download

JBoss Native is available from the [JBoss download area](#).

Install

To install the JBoss Application Server as Windows service, use the provided service.bat batch file.

```
C:\> cd c:\jboss-4.0.5\bin
```

```
C:\> service.bat install
```

Start

To start the JBoss Application Server as Windows service use Control panel or net start command. When running in service mode the console output is redirected to the file run.log. You can inspect the file for any errors during service startup.

```
C:\> net start JBAS4SVC
```

The JBoss Application Server 4.0.5 service is starting.

The JBoss Application Server 4.0.5 service was started successfully.

Stop

To stop the JBoss Application Server as Windows service use Control panel or net stop command. When running in service mode the console output is redirected to the file shutdown.log. You can inspect the file for any errors during service shutdown.

```
C:\> net stop JBAS4SVC
```

The JBoss Application Server 4.0.5 service was stopped successfully.

Restart

To restart the JBoss Application Server as Windows service use Control panel.

You'll need to reboot windows for the service to disappear from the Services snap-in

Uninstall

To remove the JBoss Application Server as Windows service use the provided service.bat batch file.

```
C:\> cd c:\jboss-4.0.5\bin
```

```
C:\> service.bat uninstall
```

Customization

Service customization is done by editing the service.bat script. Each command has a separate section that you can customize. The most common customization task would be changing service names if more then one service instances per box are required.

See [JBossSVC.exe?](#) for options.

jbossSvc -- program for running batch files as services.

```
Usage: jbossSvc -i service service.bat
        -i[wdcl] service workingpath description comment
                service.bat
        -u service
        -t service
```

Options:

```
-d Service display name
-c Service description
-w Service working path
-l Turn info logging On
```

Option 2. Use the JavaServiceWrapper by Tanuki.

You can use Java Service Wrapper : [Java Service Wrapper Integration](#) and manage it by JMX : [JBoss Integration](#)

Unzip the wrapper zip file, and do the following:

```
copy WRAPPER_HOME\bin\Wrapper.exe %JBOSS_HOME%\bin\Wrapper.exe
copy WRAPPER_HOME\lib\Wrapper.DLL %JBOSS_HOME%\lib\Wrapper.DLL
copy WRAPPER_HOME\lib\wrapper.jar %JBOSS_HOME%\lib\wrapper.jar
mkdir %JBOSS_HOME%\server\YOURCONFIG\wrapper
```

Create wrapper.conf file inside %JBOSS_HOME%\server\YOURCONFIG\wrapper with the below contents:

```
wrapper.java.command=D:/Java/jdk1.5.0_14/bin/java
```

```
wrapper.java.mainclass=org.tanukisoftware.wrapper.WrapperSimpleApp
```

```
wrapper.java.classpath.1=%JBOSS_HOME%/lib/wrapper.jar
wrapper.java.classpath.2=%JAVA_HOME%/bin/java/lib/tools.jar
wrapper.java.classpath.3=./run.jar
```

```
wrapper.java.library.path.1=%JBOSS_HOME%/lib
```

```
# these are the JAVA_OPTS
```

```
wrapper.java.additional.1=-server
```

```
# enviroment variables - define the ones that match your desired environment
```

```
wrapper.java.additional.2=-Denviromnment.variable=value
```

```
# memory parameters - define the ones that match your desired environment
```

```
wrapper.java.additional.3=-Xms64m
```

```
wrapper.java.additional.4=-Xmx96m
```

```
# If you need serialization support
```

```
wrapper.java.additional.5=-Dsession.serialization.jboss=true
```

```
wrapper.app.parameter.1=org.jboss.Main
```

```
# Parameters to be passed to the application (Jboss)
```

```
# Define server name (configuration) - If you need a config that is different than the "default" or need to run multiple configs
```

```
wrapper.app.parameter.2=-c YOURCONFIG
```

```
# Define listening IP - If you have more than one IP or want to indicate to listen on a specific IP
```

```
wrapper.app.parameter.3=-b aaa.bbb.ccc.ddd
```

```
# wrapper log location
```

```
wrapper.logfile=%JBOSS_HOME%/server/YOURCONFIG/log/wrapper.log
```

```
# You must not change below parameters without first uninstall the service
# service name
wrapper.ntservice.name=Jboss YOURCONFIG
# service display name
wrapper.ntservice.displayname=JBoss Server YOURCONFIG
```

Test the service:

```
cd %JBOSS_HOME%\bin\
wrapper.exe -c %JBOSS_HOME%\server\YOURCONFIG\wrapper\wrapper.conf
Install the service:
cd %JBOSS_HOME%\bin\
wrapper.exe -i %JBOSS_HOME%\server\YOURCONFIG\wrapper\wrapper.conf
Uninstall the service:
cd %JBOSS_HOME%\bin\
wrapper.exe -r %JBOSS_HOME%\server\YOURCONFIG\wrapper\wrapper.conf
```

Warning: make sure your JBOSS_HOME environment var is set correctly (this is not the case if you haven't done that yourself): the run.bat does set this itself, but the wrapper doesn't !

java-service-wrapper-service.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE server>
<server>
  <mbean code="org.tanukisoftware.wrapper.jmx.WrapperManager"
    name="JavaServiceWrapper:service=WrapperManager"/>

  <mbean code="org.tanukisoftware.wrapper.jmx.WrapperManagerTesting"
    name="JavaServiceWrapper:service=WrapperManagerTesting"/>
</server>
```

UNIX Installation and Daemon Setup.

Download and Installation

- 7) Download JBoss from the JBoss web site
 - a. [JBoss 4.0.5 GA Installs](#)
 - i. [JBoss 4.0.5 Zip File](#)
 - b. [JBoss Quick Start PDF Guide](#)
- 8) Download JDK 1.5.03 or higher From [Sun](#) and install
- 9) Unzip this file to the /usr/. This will create a folder /usr/jboss-4.0.5.GA
- 10) Go to /usr/jboss-4.0.5.GA/bin
- 11) Run `sh run.sh`
- 12) Once this stops processing open a web browser and go to `Http://<yourmachinename>:8080`

Note: This should show you the JBoss maintenance screen.

Note: Detailed installation instructions and trouble shooting tips can be found at [JBoss Web site](#). Be sure to install JBoss in a directory pathname that contains no spaces. Make sure no other applications are running on port 8080.

Starting JBoss on boot with Linux

Overview

Linux uses System V init scripts. Although there can be some differences between distributions, generally it looks like this:

- /etc/rc.d/init.d/ - contains the start and stop scripts (other distributions: /etc/init.d/)
- /etc/rc.(x)/ - contains links to the start and stop scripts prefixed with S or K (start or kill respectively)

There are various run levels for various stages of system use.

1. rc1.d - Single User Mode
2. rc2.d - Single User Mode with Networking
3. rc3.d - Multi-User Mode - boot up in text mode
4. rc4.d - Undefined
5. rc5.d - Multi-User Mode - boot up in X Windows
6. rc6.d - Shutdown

Most Linux systems used as Application Servers boot into run level 3 (if not, your system admin needs to be questioned on why your server needs to boot into X-Windows and needlessly waste system resources).

Your task is to:

- create a user for JBoss (recommended) so that JBoss can be restricted to accessing only the files and system resources that it has permission to access via the "jboss" user.

- create a script called /etc/rc.d/init.d/jboss
- create a link called /etc/rc3.d/S84jboss
 - optionally /etc/rc5.d/S84jboss and /etc/rc4.d/S84jboss
- create a link called /etc/rc6.d/K15jboss
 - create the K15 link in /etc/rc1.d, /etc/rc2.d, /etc/rc0.d

Create a user called "jboss"

As root type "adduser --system jboss" and enter a password in case you later want to login as user jboss; Normally you will treat jboss like the users apache, www, postgres,... and disable direct login. Su to jboss and install the server in \$JBOSS_HOME. Make sure the \$JBOSS_HOME directory can be read by "jboss" and that the \$JBOSS_HOME/server/default/work directory can be read and written. In case some permissions are messed up check the [RecommendedUNIXFilesystemPermissionsForJBossApplicationServer](#).

Note that SELinux environment uses "runuser" instead of "su" (see below). "runuser" requires a valid shell like "bin/sh" for your JBOSS user.

Create the script in /etc/rc.d/init.d

As root (su - root) type vi /etc/rc.d/init.d/jboss and paste the below:

```
#!/bin/sh
start(){
    echo "Starting jboss.."

    # If using an SELinux system such as RHEL 4, use the command below
    # instead of the "su":
    # eval "runuser - jboss -c '/opt/jboss/current/bin/run.sh > /dev/null 2> /dev/null &'
    # if the 'su -l ...' command fails (the -l flag is not recognized by my su cmd) try:
    # sudo -u jboss /opt/jboss/bin/run.sh > /dev/null 2> /dev/null &
    su -l jboss -c '/opt/jboss/current/bin/run.sh > /dev/null 2> /dev/null &'
}

stop(){
    echo "Stopping jboss.."

    # If using an SELinux system such as RHEL 4, use the command below
    # instead of the "su":
    # eval "runuser - jboss -c '/opt/jboss/current/bin/shutdown.sh -S &'
    # if the 'su -l ...' command fails try:
    # sudo -u jboss /opt/jboss/bin/shutdown.sh -S &
    su -l jboss -c '/opt/jboss/current/bin/shutdown.sh -S &'
}

restart(){
    stop
    # give stuff some time to stop before we restart
```



```

        sleep 60
# protect against any services that can't stop before we restart (warning this kills all Java
instances running as 'jboss' user)
    su -l jboss -c 'killall java'
# if the 'su -l ...' command fails try:
    # sudo -u jboss killall java
    start
}

case "$1" in
start)
    start
    ;;
stop)
    stop
    ;;
restart)
    restart
    ;;
*)
    echo "Usage: jboss {start|stop|restart}"
    exit 1
esac

exit 0

```

Also as root type "chmod 755 /etc/rc.d/init.d/jboss" in order to make the script executable. You should be able to test it by typing "service jboss stop", "service jboss start" and so forth. Note that the script hard codes \$JBOSS_HOME so make sure that it reflects your JBOSS_HOME. Because the console output is also in the server.log we're just sending it to /dev/null.

The restart target in this script does not perform the same actions as a stop followed by a start; it contains a hard-coded 60-second sleep before killing Java the hard way. This has several negative effects, including:

- making a restart take at least 60 seconds (incurring at least 60 seconds service downtime)
- potentially aborting a JBoss instance that would have shut down of its own accord in due course (the corollary of this is that the stop target is not guaranteed to stop the running JBoss instance)

Using the included init Scripts (JBoss 4.0.1 and higher)

Alternatively, JBoss 4.0.1 (and higher) comes with prebaked init scripts in the bin directory, jboss_init_redhat.sh and jboss_init_suse.sh. You can copy one of these scripts to /etc/rc.d/init.d/jboss, then make the links below, or create a symbolic link from /etc/rc.d/init.d/jboss to one of them. These scripts don't pipe logging to /dev/null, but to a real file, so you'll have to add one additional step:

```
mkdir $JBOSS_HOME/log
chown jboss $JBOSS_HOME/log
```

You will also need to modify the start and stop commands to use "runuser" instead of "su" if using a SELinux distribution such as RHEL 4.

example modifications:

```
JBOSSUS=${JBOSSUS:-"jboss"} # put this before JBOSS_CONSOLE def
SUBIT="runuser - $JBOSSUS -c " # replace su by runuser due to SELinux
JBOSS_CONSOLE="$JBOSS_HOME/log/jboss.log" # instead of nothing
chown $JBOSSUS $JBOSS_CONSOLE # make log writable by JBOSS user
JBOSSSH=${JBOSSSH:-"$JBOSS_HOME/bin/run.sh -c default"} # change "all" for
"default"
```

create links

The links will be used to identify at which run levels JBoss should be started and stopped. In general this is probably what you want (do as root):

```
ln -s /etc/rc.d/init.d/jboss /etc/rc3.d/S84jboss
ln -s /etc/rc.d/init.d/jboss /etc/rc5.d/S84jboss
ln -s /etc/rc.d/init.d/jboss /etc/rc4.d/S84jboss
```

```
ln -s /etc/rc.d/init.d/jboss /etc/rc6.d/K15jboss
ln -s /etc/rc.d/init.d/jboss /etc/rc0.d/K15jboss
ln -s /etc/rc.d/init.d/jboss /etc/rc1.d/K15jboss
ln -s /etc/rc.d/init.d/jboss /etc/rc2.d/K15jboss
```

Linux will execute the equivalent of "service jboss start" for the "S" links and "service jboss stop" for the K links.

[RedHat?](#) has a chkconfig command to manage these links, which may or may not work (it uses comments in the top of the script to determine which run-levels it should be started/stopped in). For Debian or Ubuntu, use update-rc.d(8). For [SuSe?](#), use the insserv command to properly register the new jboss script--skip the symlinking you see above, this will be ignored and will not run JBoss at boot time).

Using Tanuki Software's Java Service Wrapper

Have a look at <http://wrapper.tanukisoftware.org/> - the Java Service Wrapper is a very powerful service wrapper for any Java application and works fine with JBoss.

You can also have a look at the [Windows Java Service page](#), which has more instructions and configuration options for the Java Service Wrapper.

A basic wrapper.conf would look like this:

```
wrapper.java.command=/usr/java/bin/java
wrapper.java.mainclass=org.tanukisoftware.wrapper.WrapperSimpleApp
```

```
wrapper.java.classpath.1=../lib/wrapper.jar
wrapper.java.classpath.2=run.jar
wrapper.java.classpath.3=/usr/java/lib/tools.jar
```

```
wrapper.java.library.path.1=../lib
```

```
wrapper.java.additional.1=-server
wrapper.java.additional.2=-Dprogram.name=run.sh
```

```
wrapper.java.initmemory=256
wrapper.java.maxmemory=256
```

```
wrapper.app.parameter.1=org.jboss.Main
wrapper.app.parameter.2=-c
wrapper.app.parameter.3=default
```

```
wrapper.console.format=PM
wrapper.console.loglevel=INFO
```

```
wrapper.logfile=wrapper.log
wrapper.logfile.format=LPTM
wrapper.logfile.loglevel=INFO
wrapper.logfile.maxsize=1m
wrapper.logfile.maxfiles=1
```

```
wrapper.syslog.loglevel=NONE
```

Original version of the UNIX portion of this doc can be found at:

<http://wiki.jboss.org/wiki/StartJBossOnBootWithLinux?action=e&windowstate=normal&mode=view>