**JBoss Document**

**Pre-requirement**

JBoss version: 5.1

Jdk Version: 1.5 (0r) above

Operating System: UNIX (or) Windows

RAM size: 512 MB Min

**NOTE**: If we need to start the JBoss server in the machine, we need **JDK**. Install JDK 1.5 or 1.6 versions.

**JBoss**

1. Download the JBoss5.1 version from the **Red hat** site. Extract the zip file.
2. We’ll get one folder JBoss 5.1. In that we’ll get bin, Client, Common, docs, lib, and server folders.

Please refer the below Screen short.

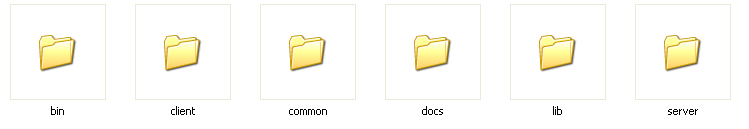


Fig: JBoss 5.1 version folders

**Common, and lib:**

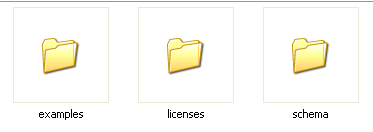
In the above folder all lib file will be available.

**Note:**

1. If your JBoss is not enterprise version we need to copy some (dependent) Jar file from here to specific lactation.
2. To find out the dependent jar files please read the Readme.html (Path: JBoss folder)

**Docs**

In docs folder we are having example, licenses and schema folders.



**Note:**

1. In the examples (\Jboss\jboss-5.1.0.GA\docs\examples\jca) folder we have **jca** folder. Here we are having all data sources related XML file

Ex: mysql-ds.xml, oracle-ds.xml, db2-ds.xml, sybase-ds etc.

1. We need to configure the data source in the server. So that we copy the data base related xml file from here to deploy directory. There we’ll configure the data source.
2. **Bin**

In the bin directory we have run.conf and run.sh or ron.bat

run.bat 🡪 it is for Windows

run.sh 🡪 it is for UNIX

In the run.conf file we need to pass the java related arguments. Refer the below details.

**Run.conf:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

rem ### -\*- batch file -\*- ######################################################

rem # ##

rem # JBoss Bootstrap Script Configuration ##

rem # ##

rem #############################################################################

rem # $Id: run.conf.bat 88820 2009-05-13 15:25:44Z dimitris@jboss.org $

rem #

rem # This batch file is executed by run.bat to initialize the environment

rem # variables that run.bat uses. It is recommended to use this file to

rem # configure these variables, rather than modifying run.bat itself.

rem #

if not "x%JAVA\_OPTS%" == "x" goto JAVA\_OPTS\_SET

rem #

rem # Specify the JBoss Profiler configuration file to load.

rem #

rem # Default is to not load a JBoss Profiler configuration file.

rem #

rem set "PROFILER=%JBOSS\_HOME%\bin\jboss-profiler.properties"

rem #

rem # Specify the location of the Java home directory (it is recommended that

rem # this always be set). If set, then "%JAVA\_HOME%\bin\java" will be used as

rem # the Java VM executable; otherwise, "%JAVA%" will be used (see below).

rem #

**set "JAVA\_HOME=C:\Java\jdk1.6.0"** 

rem #

rem # Specify the exact Java VM executable to use - only used if JAVA\_HOME is

rem # not set. Default is "java".

rem #

**set "JAVA=C:\Java\jdk1.6.0\bin\java"** 

rem #

rem # Specify options to pass to the Java VM. Note, there are some additional

rem # options that are always passed by run.bat.

rem #

rem # JVM memory allocation pool parameters - modify as appropriate.

**set "JAVA\_OPTS=-Xms128M -Xmx512M -XX:MaxPermSize=256M" **

rem # Reduce the RMI GCs to once per hour for Sun JVMs.

set "JAVA\_OPTS=%JAVA\_OPTS% -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000"

rem # Warn when resolving remote XML DTDs or schemas.

set "JAVA\_OPTS=%JAVA\_OPTS% -Dorg.jboss.resolver.warning=true"

rem # Sample JPDA settings for remote socket debugging

rem set "JAVA\_OPTS=%JAVA\_OPTS% -Xrunjdwp:transport=dt\_socket,address=8787,server=y,suspend=n"

rem # Sample JPDA settings for shared memory debugging

rem set "JAVA\_OPTS=%JAVA\_OPTS% -Xrunjdwp:transport=dt\_shmem,address=jboss,server=y,suspend=n"

:JAVA\_OPTS\_SET

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Note**: In this run.conf file we can mention JVM Heap size, Perm size, Garbage collector size. Class path also we can provide here.

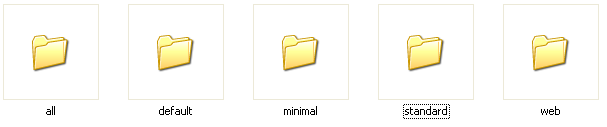
**run.bat (or) run.sh**

No need to change anything in this script. Here also we can pass the class path if it is required.

**Note:** It will call the run.con

**Server**

In the server directory we have 5 types of profiles. Please refer the below figure.



**Note**:

In the enter pries version we have one more server profile that is **Production.** It will increase the server performances.

**All**: We have all support future in the JBoss like Cluster and JMS.

**Default**: As a JBoss Instance starting purpose requirement we have in the default profile.

**Minimal**: we have very less options. If we need to configure a instance by using this, we need to define each every values in the xml files.

**Configure the two or more instances in JBoss in a single machine.**

Since 5.x version we can do two ways of configuration. That is

**I) First case:**

If we are using the cluster option in our environment just copy the server profile “all” or “Production” and past it, rename it as a production0, production1, production2.

**Note**

If you are using enterprise version in the we have server profile **Production.** Just copy that one past and rename it.

Now we are having three different instances.

Instances **starting and cluster** we can do in the instance starting time only.

**Production1 instances start command**

run.bat -c production1 –b IP Address -g productiocluster -u 239.230.010.020 -Djboss.messaging.ServerPeerID=10 -Djboss.service.binding.set=production0 (port)

**Note: UNIX**

1. run.sh -c production1 -g productiocluster -u 239.230.010.020 -Djboss.messaging.ServerPeerID=10 -Djboss.service.binding.set=production0 (port)
2. –b IP address is bind IP address

**Production2 instances start command**

run.bat -c production2 -g productiocluster -u 239.230.010.020 -Djboss.messaging.ServerPeerID=11 -Djboss.service.binding.set=production1 (port)

**Production3 instances start command**

run.bat -c production3 -g productiocluster -u 239.230.010.020 -Djboss.messaging.ServerPeerID=12 -Djboss.service.binding.set=production2 (port)

**Note:** These arguments in command mode.

**2) Second case: (Back end process)**

If we are using the cluster option in our environment just copy the server profile “all” or “Production” and past it, rename it as a production0, production1, production2.

**Production0:**

**Step1:**

Open Production0 server profiles go to this path (\Jboss\jboss-5.1.0.GA\server\production0\conf\bindingservice.beans\META-INF

Open the file **bindings-jboss-beans.xml** and edit default server name and Port number.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<?xml version="1.0" encoding="UTF-8"?>

<deployment xmlns="urn:jboss:bean-deployer:2.0">

<!-- The actual SBM from which services obtain binding information -->

<bean name="ServiceBindingManager" class="org.jboss.services.binding.ServiceBindingManager">

<annotation>@org.jboss.aop.microcontainer.aspects.jmx.JMX(name="jboss.system:service=ServiceBindingManager", exposedInterface=org.jboss.services.binding.ServiceBindingManagerMBean.class, registerDirectly=true)</annotation>

<!-- Here we use the ServiceBindingManagementObject as a factory to create the SBM -->

<constructor factoryMethod="getServiceBindingManager">

<factory bean="ServiceBindingManagementObject"/>

</constructor>

</bean>

<!-- Provides management tools with a ProfileService ManagementView

interface to the SBM and its components -->

<bean name="ServiceBindingManagementObject"

class="org.jboss.services.binding.managed.ServiceBindingManagementObject">

<constructor>

<!-- The name of the set of bindings to use for this server -->

<parameter>${jboss.service.binding.set:**production0**}</parameter>

<!-- The binding sets -->

<parameter>

<set>

<inject bean="PortsDefaultBindings"/>

<inject bean="Ports01Bindings"/>

<inject bean="Ports02Bindings"/>

<inject bean="Ports03Bindings"/>

</set>

</parameter>

<!-- Base binding metadata that is used to create bindings for each set -->

<parameter><inject bean="StandardBindings"/></parameter>

</constructor>

</bean>

<!-- The ports-default bindings are obtained by taking the base bindings and adding 0 to each port value -->

<bean name="PortsDefaultBindings" class="org.jboss.services.binding.impl.ServiceBindingSet">

<constructor>

<!-- The name of the set -->

<parameter>**production0**</parameter>

<!-- Default host name -->

<parameter>${jboss.bind.address}</parameter>

<!-- The port offset -->

<parameter>**0**</parameter> (**Default 8080**)

<!-- Set of bindings to which the "offset by X" approach can't be applied -->

<parameter><null/></parameter>

</constructor>

</bean>

**Step2:**

**Cluster configuration:**

Go to Dir path: \Jboss\jboss-5.1.0.GA\server\production0\deploy\cluster

Edit file: **hapartition-jboss-beans.xml**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<bean name="HAPartition"

class="org.jboss.ha.framework.server.ClusterPartition">

<depends>jboss:service=Naming</depends>

<annotation>@org.jboss.aop.microcontainer.aspects.jmx.JMX(name="jboss:service=HAPartition,partition=${jboss.partition.name:**Cluster1**}", exposedInterface=org.jboss.ha.framework.server.ClusterPartitionMBean.class, registerDirectly=true)</annotation>

<!-- ClusterPartition requires a Cache for state management -->

<property name="cacheHandler"><inject bean="HAPartitionCacheHandler"/></property>

<!-- Name of the partition being built -->

<property name="partitionName">${jboss.partition.name:**Cluster1**}</property>

**Note**: by default in the all or production server profiles cluster name is DefaultPartition.

Change the DefaultPartition name as per convenient.

**Step3:**

**To create the Multicast clustering:**

Go to Dir path**:** \Jboss\jboss-5.1.0.GA\server\production0\deploy\cluster\jgroups-channelfactory.sar\META-INF

Edit file: **jgroups-channelfactory-stacks.xml**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<UDP

singleton\_name="shared-udp"

mcast\_port="${jboss.jgroups.udp.mcast\_port:45688}"

mcast\_addr="${jboss.partition.udpGroup:228.11.11.11}"

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

If we want we can change the IP address and port number. Not Mandatory

**Default IP Address: 228.11.11.11**

**Port Number: 45688**

**Step4:**

**To create the Msg peer Id:**

Go to Dir path: \Jboss\jboss-5.1.0.GA\server\production0\deploy\messaging

Edit file: **messaging-service.xml**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

ute name="ServerPeerID">${jboss.messaging.ServerPeerID:**0**}</attribute>



**Step5:**

**Start the production0 Server**

Go to Dir: \Jboss\jboss-5.1.0.GA\bin

Execute below command:

run.cmd –c production0 –b 127.0.0.01 &

**Note**: **&** symblo is to run the script back end evevn closing the command promt(In UNIX).

**Step6:**

**Create Data Sources**

To create a data source go to JBoss🡪Doc🡪Examples🡪Copy the related data base .xml file from here to deploy directory.

**Ex: oracle-ds.xml**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<?xml version="1.0" encoding="UTF-8"?>

<!-- ===================================================================== -->

<!-- JBoss Server Configuration -->

<!-- ===================================================================== -->

<!-- See http://www.jboss.org/community/wiki/Multiple1PC for information about local-tx-datasource -->

<!-- $Id: oracle-ds.xml 88948 2009-05-15 14:09:08Z jesper.pedersen $ -->

<!-- ==================================================================== -->

<!-- Datasource config for Oracle originally from Steven Coy -->

<!-- ==================================================================== -->

<datasources>

<local-tx-datasource>

<jndi-name>**OracleDS**</jndi-name> (Provide the related **JNDI Name** ex: POWSAPS)

<connection-url>jdbc:oracle:thin:@youroraclehost:1521:yoursid</connection-url>

<!--

Here are a couple of the possible OCI configurations.

For more information, see http://otn.oracle.com/docs/products/oracle9i/doc\_library/release2/java.920/a96654/toc.htm

<connection-url>jdbc:oracle:oci:@youroracle-tns-name</connection-url>

or

<connection-url>jdbc:oracle:oci:@(description=(address=(host=youroraclehost)(protocol=tcp)(port=1521))(connect\_data=(SERVICE\_NAME=yourservicename)))</connection-url>

Clearly, its better to have TNS set up properly.

-->

<driver-class>oracle.jdbc.driver.OracleDriver</driver-class>

<user-name>**x**</user-name> (**Provide the Data Base Name**)

<password>**y**</password> (**Provide the Data Base Password**)

<!-- Uses the pingDatabase method to check a connection is still valid before handing it out from the pool -->

<!--valid-connection-checker-class-name>org.jboss.resource.adapter.jdbc.vendor.OracleValidConnectionChecker</valid-connection-checker-class-name-->

<!-- Checks the Oracle error codes and messages for fatal errors -->

<exception-sorter-class-name>org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter</exception-sorter-class-name>

<!-- sql to call when connection is created

<new-connection-sql>some arbitrary sql</new-connection-sql>

-->

<!-- sql to call on an existing pooled connection when it is obtained from pool - the OracleValidConnectionChecker is prefered

<check-valid-connection-sql>some arbitrary sql</check-valid-connection-sql>

-->

<!-- corresponding type-mapping in the standardjbosscmp-jdbc.xml (optional) -->

<metadata>

<!-- The minimum connections in a pool/sub-pool. Pools are lazily constructed on first use -->

<min-pool-size>5</min-pool-size>

<!-- The maximum connections in a pool/sub-pool -->

<max-pool-size>20</max-pool-size>

<type-mapping>Oracle9i</type-mapping> (**Provide the version of Data Base**)

</metadata>

</local-tx-datasource>

</datasources>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Note: To Creating the connect pool configuration. Please add below two lines in the **oracle-ds.xml** file.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<!-- The minimum connections in a pool/sub-pool. Pools are lazily constructed on first use -->

<min-pool-size>5</min-pool-size> (Min number of connections)

<!-- The maximum connections in a pool/sub-pool -->

<max-pool-size>20</max-pool-size> (Max number of connections)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Production1 Configuration:**

Follow the production0 steps.

**Note:**

**Step1**:

In the **bindings-jboss-beans.xml** file change the below lines

<parameter>${jboss.service.binding.set:**production1**}</parameter>

<parameter>**production1**</parameter>

<parameter>**100**</parameter> (**It will take 8180**)

**Step2:**

Copy the **hapartition-jboss-beans.xml** file from production0 to prodction1 deploy directory.

**Note**: Then only production0 and prodution1 run in one cluster and communicate.

**Step3:**

Copy the **jgroups-channelfactory-stacks.xml** file from production0 to prodction1.

**Step4:**

Copy the **messaging-service.xml** file from production0 to prodction1 deploy directory.

**Note**: Change the Msg peer ID. (It should be unique )

EX: Ute name="ServerPeerID">${jboss.messaging.ServerPeerID:**1**}</attribute>

**Step5:**

Go to Dir: \Jboss\jboss-5.1.0.GA\bin

Execute below command:

run.cmd –c production1 –b 127.0.0.01 &

**Step6:** \Jboss\jboss-5.1.0.GA\server\production1\deploy\jbossweb.sar in this path edit server.xml file and add jvmRoute="production1” perameter in this file.

Ex:

<Engine name="jboss.web" defaultHost="localhost" **jvmRoute="production1"**>

**Deployment:**

War/ear file we need to deploy in the below path: **\Jboss\jboss-5.1.0.GA\server\production1\deploy**

**Production2 Configuration:**

Similarly follow the prduction1 steps

**Production3 Configaration:**

Similarly follow the prduction1 steps

**Integration with Apache web server**

Apache to JBoss we can configare multpile ways.

EX:

1. Mod\_jk.so
2. Mod\_cluster.so

With out s/w load balance also we can configure.

**Configaration:**

To configure this we need to edit the http.conf file (\Apache\Apache2\conf).

Just add few linesin http.conf file

Step1: uncomment below mouldes

*LoadModule proxy\_module modules/mod\_proxy.so*

*LoadModule proxy\_connect\_module modules/mod\_proxy\_connect.so*

*LoadModule proxy\_http\_module modules/mod\_proxy\_http.so*

Step2: add below lines

*# control client access*

*<Proxy* [*http://localhost:80*](http://localhost:80)*>* **server name:port number**

*Order Deny,Allow*

*Allow from all*

*</Proxy>*

*ProxyPass /richfaces-demo http://localhost:8180/richfaces-demo*

*ProxyPassReverse /richfaces-demo* [*http://localhost:8180/richfaces-demo*](http://localhost:8180/richfaces-demo)

**APP** **Server Name: Port Application Name**

1. **Mod\_jk.conf**

To configaring the JBoss with Apache we need mod\_jk.so. Please download the file from below.

We need to create 2 file.

1. workers.properties
2. Mod\_jk.conf

**workers.properties**

**Justification:** If we have two production JBoss server profiles we need to create s/w loadbalancer.

*# Define list of workers that will be used*

*# for mapping requests*

*worker.list=loadbalancer,jkstatus***,loadbalancerprod1****loadbalancer Name**

*# Define* ***production1***

*# modify the host as your host IP or DNS name.*

*worker.****production1****.port=8109*

*worker.production1.host=localhost*

*worker.production1.type=ajp13*

*worker.production1.ping\_mode=A*

*worker.production1.lbfactor=1*

*# Define* ***producton2***

*# modify the host as your host IP or DNS name.*

*worker.production2.port=8209*

*worker.production2.host=localhost*

*worker.production2.type=ajp13*

*worker.production2.ping\_mode=A*

*worker.production2.lbfactor=1*

*#*

*# Load-balancing behavior*

*worker.loadbalancerprod1.type=lb*

*worker.loadbalancerprod1.balance\_workers=production1,production2*

*worker.loadbalancerprod1.sticky\_session=1*

*# Status worker for managing load balancer*

*worker.jkstatus.type=status*

**Mod\_jk.conf**

*# Include mod-jk*

*# Include conf/mod-jk.conf*

*# Enable mod-jk*

*LoadModule jk\_module modules/mod\_jk.so* **This line calls mod\_jk file**

*JkWorkersFile conf/workers.properties* **This line callsworker.properties file**

*##JkShmFile logs/mod\_jk.shm*

*JkLogFile logs/mod\_jk.log*

*JkLogLevel info*

*JkOptions +ForwardURIProxy +RejectUnsafeURI*

*# Add the jkstatus mount point*

*# Enable the JK manager access from localhost only*

*<Location /jkmanager/>*

*JkMount jkstatus*

*Order deny,allow*

*Deny from all*

*Allow from 127.0.0.1*

*</Location>*

*<VirtualHost \*:80>*

*# Add the jkstatus mount point*

*JkMount /richfaces-demo/\* loadbalancerprod1*

*</VirtualHost>*