

Minimal JBoss 4.0.3 for JBoss Portal 2.0

Part 1. Very Minimal JBoss Web configuration

Externalize services from jboss-service.xml. During startup the configuration is parsed in the following order. First jboss-service.xml is processed. Next all services are deployed from deploy folder. The configuration files in deploy folder are processed in alphabetical order.

After removing all the services from jboss-service.xml and placing them in deploy I've got the following configuration:

jboss-service.xml

```
<server>
  <classpath codebase="${jboss.server.lib.url:lib}" archives="*" />
  <mbean code="org.jboss.deployment.scanner.URLDeploymentScanner"
    name="jboss.deployment:type=DeploymentScanner,flavor=URL">
    <depends optional-attribute-name="Deployer">jboss.system:service=MainDeployer</depends>
    <attribute name="URLComparator">org.jboss.deployment.DeploymentSorter</attribute>
    <attribute name="FilterInstance"
      attributeClass="org.jboss.deployment.scanner.DeploymentFilter"
      serialDataType="javaBean">
      <property name="prefixes">#,%,\,.,,_,$_</property>
      <property name="suffixes">#,$,%~, \, v, .BAK, .bak, .old, .orig, .tmp, .rej, .sh</property>
    </property>
    name="matches">.make.state,.nse_depinfo, CVS, CVS.admin, RCS, RCSLOG, SCCS, TAGS, core, tags</property>
    </attribute>
    <attribute name="ScanPeriod">5000</attribute>
    <attribute name="ScanEnabled">true</attribute>
    <attribute name="URLs">
      deploy/
    </attribute>
    <attribute name="RecursiveSearch">True</attribute>
  </mbean>
</server>
```

01-basic-threadpool-service.xml

```
<server>
  <!-- A Thread pool service -->
  <mbean code="org.jboss.util.threadpool.BasicThreadPool"
    name="jboss.system:service=ThreadPool">
    <attribute name="Name">JBoss System Threads</attribute>
    <attribute name="ThreadGroupName">System Threads</attribute>
    <attribute name="KeepAliveTime">60000</attribute>
    <attribute name="MaximumPoolSize">10</attribute>
    <attribute name="MaximumQueueSize">1000</attribute>
    <attribute name="BlockingMode">run</attribute>
  </mbean>
</server>
```

02-naming-service.xml

```
<server>
  <!-- ===== -->
  <!-- JNDI -->
  <!-- ===== -->
  <mbean code="org.jboss.naming.NamingService"
    name="jboss:service=Naming">
    <attribute name="CallByValue">false</attribute>
    <attribute name="Port">1099</attribute>
    <attribute name="BindAddress">${jboss.bind.address}</attribute>
```

```
<attribute name="RmiPort">1098</attribute>
<attribute name="RmiBindAddress">${jboss.bind.address}</attribute>
<depends optional-attribute-name="LookupPool"
    proxy-type="attribute">jboss.system:service=ThreadPool</depends>
</mbean>
</server>
```

03-logging-service.xml

```
<server>
<!-- ===== -->
<!-- Log4j Initialization -->
<!-- ===== -->

<mbean code="org.jboss.logging.Log4jService"
    name="jboss.system:type=Log4jService,service=Logging">
    <attribute name="ConfigurationURL">resource:log4j.xml</attribute>
    <attribute name="Log4jQuietMode">>true</attribute>
    <attribute name="RefreshPeriod">60</attribute>
</mbean>
</server>
```

With this configuration JBoss starts without any problems But standard shutdown script does not work - invoker is not present. To add several invokers services we will need to deploy transaction manager service (as JMX invokers depend on transaction manager service).

Adding transaction manager and invokers

04-transaction-manager-service.xml

```
<server>
<!-- ===== -->
<!-- Transactions -->
<!-- ===== -->
<!-- The configurable Xid factory. For use with Oracle, set pad to true -->
<mbean code="org.jboss.tm.XidFactory"
    name="jboss:service=XidFactory">
</mbean>

<mbean code="org.jboss.tm.TransactionManagerService"
    name="jboss:service=TransactionManager">
    <attribute name="TransactionTimeout">300</attribute>
    <attribute name="GlobalIdsEnabled">>true</attribute>
    <depends optional-attribute-name="XidFactory">jboss:service=XidFactory</depends>
</mbean>
</server>
```

05-invokers-service.xml

```
<server>
<!-- ===== -->
<!-- Invokers to the JMX node -->
<!-- ===== -->

<!-- RMI/JRMP invoker -->
<mbean code="org.jboss.invocation.jrmp.server.JRMPInvoker"
    name="jboss:service=invoker,type=jrmp">
    <attribute name="RMIObjectPort">4444</attribute>
    <attribute name="ServerAddress">${jboss.bind.address}</attribute>
    <depends>jboss:service=TransactionManager</depends>
</mbean>
```

```

<mbean code="org.jboss.invocation.local.LocalInvoker"
  name="jboss:service=invoker,type=local">
  <depends>jboss:service=TransactionManager</depends>
</mbean>

<mbean code="org.jboss.invocation.pooled.server.PooledInvoker"
  name="jboss:service=invoker,type=pooled">
  <attribute name="NumAcceptThreads">1</attribute>
  <attribute name="MaxPoolSize">300</attribute>
  <attribute name="ClientMaxPoolSize">300</attribute>
  <attribute name="SocketTimeout">60000</attribute>
  <attribute name="ServerBindAddress">${jboss.bind.address}</attribute>
  <attribute name="ServerBindPort">4445</attribute>
  <attribute name="ClientConnectAddress">${jboss.bind.address}</attribute>
  <attribute name="ClientConnectPort">0</attribute>
  <attribute name="ClientRetryCount">1</attribute>
  <attribute name="EnableTcpNoDelay">>false</attribute>

  <depends optional-attribute-name="TransactionManagerService">
    jboss:service=TransactionManager
  </depends>
</mbean>

<!-- The JRMP invoker proxy configuration for the InvokerAdaptorService -->
<mbean code="org.jboss.invocation.jrmp.server.JRMPProxyFactory"
  name="jboss.jmx:type=adaptor,name=Invoker,protocol=jrmp,service=proxyFactory">
  <depends optional-attribute-name="InvokerName">jboss:service=invoker,type=jrmp</depends>
  <depends optional-attribute-name="TargetName">jboss.jmx:type=adaptor,name=Invoker</depends>
  <attribute name="JndiName">jmx/invoker/RMIAdaptor</attribute>
  <attribute name="ExportedInterfaces">org.jboss.jmx.adaptor.rmi.RMIAdaptor,
    org.jboss.jmx.adaptor.rmi.RMIAdaptorExt
  </attribute>
  <attribute name="ClientInterceptors">
    <interceptors>
      <interceptor>org.jboss.proxy.ClientMethodInterceptor</interceptor>
      <interceptor>org.jboss.proxy.SecurityInterceptor</interceptor>
<interceptor>org.jboss.jmx.connector.invoker.client.InvokerAdaptorClientInterceptor</interceptor>
      <interceptor>org.jboss.invocation.InvokerInterceptor</interceptor>
    </interceptors>
  </attribute>
  <depends>jboss:service=Naming</depends>
</mbean>

<!-- Create a mapping from the legacy jmx-rmi-adaptor.sar binding to the
jmx-invoker-adaptor-server.sar JndiName specified above for backwards
portability.
-->
<mbean code="org.jboss.naming.NamingAlias" name="jboss.jmx:alias=jmx/rmi/RMIAdaptor">
  <attribute name="FromName">jmx/rmi/RMIAdaptor</attribute>
  <attribute name="ToName">jmx/invoker/RMIAdaptor</attribute>
  <depends>jboss:service=Naming</depends>
</mbean>

<!-- This is the service that handles the RMIAdaptor invocations by routing
them to the MBeanServer the service is deployed under. -->
<mbean code="org.jboss.jmx.connector.invoker.InvokerAdaptorService"
  name="jboss.jmx:type=adaptor,name=Invoker"
  xmbean-dd="">
  <xmbean>
    <attribute access="read-only" getMethod="getName">
      <description>The class name of the MBean</description>
      <name>Name</name>
      <type>java.lang.String</type>
    </attribute>
    <attribute access="read-only" getMethod="getState">
      <description>The status of the MBean</description>
      <name>State</name>
      <type>int</type>
    </attribute>
    <attribute access="read-only" getMethod="getStateString">
      <description>The status of the MBean in text form</description>
      <name>StateString</name>
    </attribute>
  </xmbean>

```

```

        <type>java.lang.String</type>
    </attribute>
    <attribute access="read-write" getMethod="getExportedInterfaces"
setMethod="setExportedInterfaces">
        <description>The interfaces the invoker proxy supports</description>
        <name>ExportedInterfaces</name>
        <type>[Ljava.lang.Class;</type>
    </attribute>
    <attribute access="read-only" getMethod="getMethodMap">
        <description>Map(Long hash, Method) of the proxy interface methods</description>
        <name>MethodMap</name>
        <type>java.util.Map</type>
    </attribute>
    <!-- Operations -->
    <operation>
        <description>The start lifecycle operation</description>
        <name>start</name>
    </operation>
    <operation>
        <description>The stop lifecycle operation</description>
        <name>stop</name>
    </operation>
    <operation>
        <description>The detyped lifecycle operation (for internal use only)</description>
        <name>jbossInternalLifecycle</name>
        <parameter>
            <description>The lifecycle operation</description>
            <name>method</name>
            <type>java.lang.String</type>
        </parameter>
        <return-type>void</return-type>
    </operation>

    <operation>
        <description>The detached invoker entry point</description>
        <name>invoke</name>
        <parameter>
            <description>The method invocation context</description>
            <name>invocation</name>
            <type>org.jboss.invocation.Invocation</type>
        </parameter>
        <return-type>java.lang.Object</return-type>
    </operation>
</mbean>
<attribute name="ExportedInterfaces">org.jboss.jmx.adaptor.rmi.RMIAdaptor ,
    org.jboss.jmx.adaptor.rmi.RMIAdaptorExt
</attribute>
</mbean>
</server>

```

For this services to load properly we need to add the following libraries to **lib** folder.

- jboss-transaction.jar (required by TransactionManager)
- jboss-j2ee.jar (required by TransactionManager)
- jboss.jar (required by invokers-service)
- jmx-adaptor-plugin.jar (required by invokers-service)

The resulting configuration folder is shown below

```

D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\conf
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\deploy
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\lib
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\log
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\tmp
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\conf\jboss-service.xml
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\conf\jndi.properties
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\conf\log4j.xml
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\deploy\01-basic-threadpool-service.xml
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\deploy\02-naming-service.xml

```

```
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\deploy\03-logging-service.xml
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\deploy\04-transaction-manager-service.xml
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\deploy\05-invokers-service.xml
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\lib\jboss-j2ee.jar
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\lib\jboss-management.jar
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\lib\jboss-minimal.jar
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\lib\jboss-transaction.jar
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\lib\jboss.jar
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\lib\jmx-adaptor-plugin.jar
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\lib\jnpserver.jar
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\lib\log4j.jar
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\log\boot.log
D:\dev-tmp\jboss-4.0.3SP1\server\min-wiki-portal-1\log\server.log
```

In addition to the minimal configuration, this configuration has transaction manager service and several invoker services.

Deploying Tomcat and jmx-console

The above configuration has everything required to run Tomcat, except JaasSecurityManager. To install security service we add 06-security-service.xml to **deploy** folder and copy jar dependencies to **lib** folder.

06-security-service.xml

```
<server>
  <!-- ===== -->
  <!-- Security -->
  <!-- ===== -->
  <!-- JAAS security manager and realm mapping -->
  <mbean code="org.jboss.security.plugins.JaasSecurityManagerService"
    name="jboss.security:service=JaasSecurityManager">
    <attribute
name="SecurityManagerClassName">org.jboss.security.plugins.JaasSecurityManager</attribute>
    <attribute name="DefaultUnauthenticatedPrincipal">anonymous</attribute>
    <attribute name="DefaultCacheTimeout">1800</attribute>
    <attribute name="DefaultCacheResolution">60</attribute>
  </mbean>
</server>
```

Copy the following to lib folder

- jbossex.jar
- commons-logging.jar

At this point everything is ready to deploy tomcat.

Copy **jbossweb-tomcat55.sar** to **deploy** folder and **javax.servlet.jar** and **javax.servlet.jsp.jar** to **lib** folder in your configuration.

And the final step to get minimal useful configuration with Tomcat Service is to deploy jmx-console application.

Copy the **jmx-console.war** to deploy folder.