। समुद्योगो हि विश्वजित् ।

How to Integrate IIS with JBoss / Tomcat Under Windows XP and Linux

Yogesh Chaudhari

IT SHASTRA (INDIA) PVT. LTD.
106, Bldg 2, Sector-1,
Millennium Business Park,
Mahape, Navi Mumbai –400 701.
INDIA
Phone: 91-22-2778 1021 / 1022
Web Site : http://www.itshastra.com/
Email : info@itshastra.com

Introduction

This is a “how to” procedure to make IIS redirect JSP's and Servlets execution to JBoss application server, which is useful to integrate existing IIS application with newly developed modules using Java technology.

The existing IIS application should not be disturb and system should work with the minimum settings. The ISAPI filter is having capability of checking and redirecting any request made to IIS web server to any other web server or application server. It works with Jboss application server running on windows or Linux based systems.

Softwares required for Integration

1 Web Server
IIS version: 5.0

2 Application Server
JBoss 4.0 (having in-built Tomcat 5.0)

3 Java Software Development Kit
J2sdk1.4

Concept

Normally IIS can not execute Servlets and Java Server Pages (JSPs), configuring IIS to use the JK ISAPI redirector plug in will let IIS send servlet and JSP requests to Tomcat (and this way, serve them to clients).

ISAPI Filter and Redirector for Tomcat

1) The IIS-Tomcat redirector is an IIS plugin (filter + extension), IIS loads the redirector plugin and calls its filter function for each in-coming request.
2) The filter then tests the request URL against a list of URI-paths held inside uriworkermap.properties, If the current request matches one of the entries in the list of URI-paths, the filter transfers the request to the extension.
3) The extension collects the request parameters and forwards them to the appropriate worker using the ajp1X protocol.
4) The extension collects the response from the worker and returns it to the browser.

Tomcat Workers

A Tomcat worker (like a thread) is a Tomcat instance that is waiting to execute servlets on behalf of some web server. For example, we can have a web server such as Apache/IIS forwarding servlet requests to a Tomcat process (the worker) running behind it.

The scenario described above is a very simple one; in fact one can configure multiple Tomcat workers to serve servlets on behalf of a certain web server. The reasons for such configuration can be:

We want different contexts to be served by different Tomcat workers to provide a development environment where all the developers share the same web server but own a Tomcat worker of their own.
We want different virtual hosts served by different Tomcat processes to provide a clear separation between sites belonging to different companies.
We want to provide load balancing, meaning run multiple Tomcat workers each on a machine of its own and distribute the requests between them.
There are probably more reasons for having multiple workers but I guess that this list is enough...
Tomcat workers are defined in a properties file dubbed workers.properties and this tutorial explains how to work with it.

The IIS-Tomcat redirector was developed and tested on

WinNT4.0-i386 SP4/SP5/SP6a (should be able to work with other service packs), Win2K and WinXP and Win98.
IIS4.0 and PWS4.0 (numerous people have working IIS 5 and IIS 6 configurations)
Tomcat 3.2 and later, Tomcat 4.x and Tomcat 5

The redirector uses ajp12 and ajp13 to send requests to the Tomcat containers.

Apache JServ Protocol version 1.3 (ajp13)

The ajp13 protocol is packet-oriented. A binary format was presumably chosen over the more readable plain text for reasons of performance. The web server communicates with the servlet container over TCP connections. To cut down on the expensive process of socket creation, the web server will attempt to maintain persistent TCP connections to the servlet container, and to reuse a connection for multiple request/response cycles.

Once a connection is assigned to a particular request, it will not be used for any others until the request-handling cycle has terminated. In other words, requests are not multiplexed over connections. This makes for much simpler code at either end of the connection, although it does cause more connections to be open at once.

Once the web server has opened a connection to the servlet container, the connection can be in one of the following states:

Idle - No request is being handled over this connection.
Assigned - The connecton is handling a specific request.

Once a connection is assigned to handle a particular request, the basic request information (e.g. HTTP headers, etc) is sent over the connection in a highly condensed form (e.g. common strings are encoded as integers). Details of that format are below in Request Packet Structure. If there is a body to the request (content-length > 0), that is sent in a separate packet immediately after.

Integration and Configuration Steps

STEP 1

On IIS Box create folder C:\IISSettings.

We have to create following files in this directory -
iis_redirect.reg – Registry settings file
uriworkermap.properties – URI mapping for applications from IIS to Jboss
worker.properties – Workers information to redirect requests to Jboss along with load balancer settings for more than one host.

Download Apache Jakarta Tomcat Connectors from http://tomcat.apache.org/connectors-doc/

isapi_redirect.dll – ISAPI Filter dll which filters and redirect requests to Jboss.

sample.war – Sample application to test Integration.
sample – Sample virtual directory required under IIS to run Jboss application "sample".
sampleTest – Sample application to use with Jboss.

STEP 2

Create the file iis_redirect.reg with Notepad to reflect the location if .properties files are installed in another location. The file will look like this:

```
REGEDIT4
[HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\Jakarta Isapi Redirector\1.0]
"extension_uri"="/jakarta/isapi_redirect.dll"
"log_file"="C:\\IISSettings\\iis_redirect.log"
"log_level"="emerg"
"worker_file"="C:\\IISSettings\\worker.properties"
"worker_mount_file"="C:\\IISSettings\\uriworkermap.properties"
```

Save this file, and then double click on it. You will be asked whether you want to add this information to your registry. Click "Yes".

STEP 3

Create the file C:\IISSettings\worker.properties with Notepad to reflect the location where you installed the JDK and .properties files. i.e.

The following sample setting shows how three worker nodes are defined with load balancing behavior to work with IIS.

```
# Begins worker.properties here

# Define path of .properties files and jdk
workers.tomcat_home=C:\IISSettings
workers.java_home=C:\j2sdk1.4.2_04
ps=\

# Define workers using ajp13
#worker.list=worker1, worker2
```

Specify the name of Jboss servers running on linux box.

```
# Set properties for worker1 (ajp13)
worker.worker1.type=ajp13
worker.worker1.host=JBossAppServ1
worker.worker1.port=8009
worker.worker1.lbfactor=50
worker.worker1.cachesize=10
worker.worker1.cache_timeout=600
worker.worker1.socket_keepalive=1
worker.worker1.reclycle_timeout=300

# Set properties for worker2 (ajp13)
worker.worker2.type=ajp13
worker.worker2.host=JBossAppServ2
worker.worker2.port=8009
worker.worker2.lbfactor=50
worker.worker2.cachesize=10
worker.worker2.cache_timeout=600
worker.worker2.socket_keepalive=1
worker.worker2.reclycle_timeout=300

# Set properties for worker3 (ajp13)
worker.worker3.type=ajp13
worker.worker3.host=JBossAppServ3
worker.worker3.port=8009
worker.worker3.lbfactor=50
worker.worker3.cachesize=10
worker.worker3.cache_timeout=600
worker.worker3.socket_keepalive=1
worker.worker3.reclycle_timeout=300

# Load-balancing behaviour
worker.loadbalancer.type=lb
worker.loadbalancer.balanced_workers=worker1, worker2, worker3
worker.loadbalancer.sticky_session=1
worker.loadbalancer.local_worker_only=1
worker.list=loadbalancer

# Ends worker.properties here
```

Note

1) Change property worker.<<worker name>>.host=<<host name>> depending on number of application servers required.

2) The property worker.loadbalancer.balanced_workers=<<list of workers>> must be changed if any worker is added or removed.

3) To remove load balancer behavior uncomment property and
#worker.list=worker1, worker2

4) JDK is not required on IIS Box, if we don't want to install Jboss on IIS Box.

STEP 4

Create the file C:\IISSettings\uriworkermap.properties with Notepad to register the URL's that will be executed by JBoss.

```
# Begins URI mapping configuration here

# Set the default worker
# default.worker=worker1

# Mount the Servlet context to the ajp13 worker
#/servlet/*=$(default.worker)

# Mount the context to the ajp13 worker
/sample=loadbalancer
/sample/*=loadbalancer

# Advanced mount of the context
/sample/servlet/*=loadbalancer

# Ends URI mapping configuration here
```

Where sample is the name of your JBoss application.

STEP 5

The Jboss 4.0 and above has default connector settings for 8009.

For Jboss other than 4.0, check the AJP connector is set on port 8009 in the <JBoss Directory>\server\default\deploy\jbossweb-tomcat41.sar\META-INF\jboss-service.xml file.

```
<!-- A AJP 1.3 Connector on port 8009 -->
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
  port="8009" minProcessors="5" maxProcessors="75"
  enableLookups="true" redirectPort="8443"
  acceptCount="10" debug="0" connectionTimeout="20000"
  useURIValidationHack="false"
  protocolHandlerClassName="org.apache.jk.server.JkCoyoteHandler"/>
```

If there is another connector setting the port 8009, comment it.

STEP 6

Start the Internet Services Manager.

Go to Web Site (for which integration is required) and add a New Virtual Directory called jakarta which points to C:\IISSettings or wherever you installed isapi_redirect.dll. (Download Apache Jakarta Tomcat Connectors from http://tomcat.apache.org/connectors-doc/)

Provide only read and execute privileges for security purposes.

STEP 7

Define filter for in IIS for JBoss.

1. Right click on the server, at the Internet Services Manager, and select Properties.
2. Select WWW Service and then click Edit.
3. Select ISAPI Filters and then click Add.
4. Browse the file C:\IISSettings\isapi_redirect.dll and set the filter name to jakarta.

After adding the isapi filter, the status will be a red down arrow. You need to stop the IIS server, and start the IIS server. The status should now be green.

STEP 8

Create one IIS virtual directory for each JBoss application that you will run. The virtual directory name must be the same as the JBoss application and it will point to some directory where a default.htm or index.htm file must exists.
The default.htm or index.htm file must contain the following:

```
<META HTTP-EQUIV="refresh" CONTENT="0;URL=http://localhost/application_name/">
```

Where application_name is the name of your JBoss application.

Go to Web Site (for which integration is required) and add a New Virtual Directory called “sample” which points to C:\Inetpub\wwwroot\sample or wherever you moved sample folder containing index.htm. Make sure in the properties of virtual directory Default Document is enabled as index.htm.

Provide only read and execute privileges for security purposes.

STEP 9

The Jboss must be installed on all the host configured in worker.properties file.
The host can be either of Windows or Linux.
Check is Jboss is running successfully. Using url http://<<host name>>:8080/.
Check AJP connector is running on port 8009 in the log. Like -

```
<<Date>> <<Time>> INFO  [org.apache.jk.common.ChannelSocket] JK2: ajp13 listening on /0.0.0.0:8009
```

If AJP Connector is not running on port 8009, go through the step 5 again.

Copy the sample Jboss application sample.war folder in <Jboss_home>\server\default\deploy\ folder.

Check is Jboss application is running successfully. Using url http://<<host name>>:8080/sample.

STEP 10

Set the sample application under IIS to run with Jboss application.
Create directory with the name sampleTest under C:\inetpub\wwwroot\. Copy all the Html, ASP and Dll files from sampleTest folder extracted in Step 1 to C:\inetpub\wwwroot\sampleTest folder. Register the Dlls using regsvr32.exe, if any. Create IIS virtual directory with the name sampleTest under required Web Site.

Provide only read and execute privileges for security purposes.

STEP 11

If the ISAPI redirector does not work properly and not showing green status in step 7 then you should restart the computer.

Conclusion

The existing IIS application can be integrated with new modules developed using Java technoloty with the minimum settings. It works with Jboss application server running on Linux based systems which is the cost effective solution.

References

1 Jakarta Tomcat Connector - IIS HowTo

http://jakarta.apache.org/tomcat/connectors-doc/howto/workers.html

2 IIS Integration with jboss-3.2.1_tomcat-4.1.24

http://jboss.org/index.html?module=bb&op=viewtopic&p=3816794

3 Using mod_jk 1.2.x with JBoss/Tomcat bundle and Apache2

This wiki outlines the various steps required to install a basic load-balancing solution based on JBoss/Tomcat and mod_jk 1.2.

http://www.jboss.org/wiki/Wiki.jsp?page=UsingMod_jk1.2WithJBoss

4 IIS Jboss/Tomcat

http://www.talkaboutprogramming.com/group/comp.lang.java.help/messages/215200.html

5 Tomcat IIS HowTo

By Gal Shachor <shachor@il.ibm.com>

http://jakarta.apache.org/tomcat/tomcat-3.3-doc/tomcat-iis-howto.html

6 Jakarta Tomcat Connector - Workers HowTo

http://jakarta.apache.org/tomcat/connectors-doc/howto/workers.html

7 Jakarta Tomcat Connector - Quick Start HowTo

http://jakarta.apache.org/tomcat/connectors-doc/howto/quick.html

8 How To Setup TOMCAT 4.0.1 on Windows 2000 with IIS and NT Service

http://www.vacodi.com/howto/tomcat/iisnt/

9 IIS HowTo

http://jakarta.apache.org/tomcat/tomcat-4.1-doc/jk2/jk/iishowto.html

10 AJPv13

Apache JServ Protocol version 1.3

http://jakarta.apache.org/tomcat/tomcat-4.1-doc/jk2/common/AJPv13.html