

# **Proxy Servlet for the HTTP nodes**

## **Version 2.0**

14 February 2006

Trevor Dolby

Vicente Suarez

IBM United Kingdom Ltd  
Hursley Park  
Winchester, Hampshire  
United Kingdom

**SupportPac IE01**

**Property of IBM**

**Take Note!**

Before using this report be sure to read the general information under “Notices”.

**Second Edition, February 2006**

This edition applies to Version 2.0 of the Proxy Servlet for the HTTP input and reply nodes and to all subsequent releases and modifications unless otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2004, 2006**. All rights reserved. Note to US Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule contract with IBM Corp.

---

## Table of contents

Proxy Servlet for the HTTP nodes .....	i
Table of contents.....	iii
Table of figures .....	iv
Notices .....	v
Trademarks and service marks .....	v
Summary of amendments .....	vii
Preface.....	viii
Bibliography .....	ix
Chapter 2. Overview .....	1
Description of the solution components .....	1
Web Services Clients.....	1
Servlet Container .....	2
Proxy Servlet.....	2
Message Broker.....	3
Chapter 3. Installing the Proxy Servlet.....	4
SupportPac contents .....	4
Platforms supported .....	4
Prerequisites.....	4
Installing the Proxy Servlet.....	4
Installation on Apache Tomcat 5 .....	4
Configuration of Apache Tomcat for HTTPS (SSL).....	12
Chapter 4. Using the Proxy Servlet.....	15

---

## Table of figures

Figure 1 - Proxy Servlet Implementation.....	1
Figure 2 - Tomcat home page.....	5
Figure 3 - Tomcat Manager.....	10
Figure 4 - WAR file to deploy .....	10
Figure 5 - Servlet deployed.....	11
Figure 6 - Proxy Servlet test response.....	11
Figure 7 - Proxy Servlet trace output .....	12
Figure 8 - Tomcat certificate verification .....	13
Figure 9 - Sample Test Message Flow .....	16
Figure 10 - URL Selector property of a HTTP Input node .....	16

---

## Notices

**Note: The provisions set out in the following two paragraphs do not apply in any country, to the extent only that in such provision is inconsistent with mandatory local law.**

The information contained, and any technique described, in this document has not been submitted to any formal IBM test and is distributed “AS-IS”. The use of such information or technique (including any implementation of any such information or technique) is the user’s responsibility and depends on the user’s ability to evaluate and integrate them into the user’s operational environment. While IBM has reviewed each item for accuracy in a specific situation, IBM offers no guarantee that the same or similar results will be obtained elsewhere. Users attempting to adapt any technique described in this document to their own environments do so at their own risk.

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION □□AS IS□□ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTY OF NON-INFRINGEMENT, MERCHANTABILITY, SATISFACTORY QUALITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states and countries do not allow the disclaimer of express or implied warranties in certain transactions, therefore the above statements may not apply to you. In any such event, you will only benefit from the minimum applicable implied warranty.

Any reference in this publication to an IBM product, program or service does not imply that IBM intends to make it available in all countries in which IBM operates.

Any reference in this publication to an IBM licensed program or another IBM product is not intended to state or imply that only IBM’s program or other product may be used. Any functionally equivalent program that does not infringe applicable intellectual property rights may be used instead of the IBM licensed program or other IBM product.

The user is responsible for evaluating and verifying the operation of the material supplied in conjunction with this document in conjunction with other products, except those expressly designated by IBM.

International Business Machines Corporation may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

The IBM Director of Licensing, International Business Machines Corporation, North Castle Drive, Armonk, New York 10504-1785 USA.

## Trademarks and service marks

The following terms, used in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

- IBM
- MQSeries
- WebSphere MQ (WMQ)
- MQSI
- WebSphere MQ Integrator (WMQI)
- WebSphere Business Integration Message Broker (WBIMB)
- WebSphere Message Brokers (WMB)
- WebSphere Application Server (WAS)
- Rational Application Developer (RAD)
- DB2

The following terms are trademarks of other companies:

- Windows NT/2000/XP – Microsoft
- Apache Jakarta Project – Apache Software Foundation
- Apache Tomcat Server V5 – Apache Software Foundation

---

## Summary of amendments

Date	Changes
21 June 2004	Initial release
29 September 2004	Corrections to the text
03 November 2004	Add use of sample test flow and test message
14 February 2006	New version of the Proxy Servlet for WMB V6

---

## **Preface**

The WebSphere Business Integration Message Broker V5.0 (WBIMB) and WebSphere Message Broker V6.0 (WMB) support Web Services requests over HTTP. The WBIMB or WMB brokers have an internal HTTP listener that receives requests that are propagated to message flows that use HTTP Input and HTTP Reply nodes. The WBIMB internal HTTP listener does not support the secure communications protocol HTTPS (SSL) but WMB V6 does support it.

This SupportPac is a Java Proxy Servlet that implements the functionality of the WBIMB or WMB internal HTTP listener in an external Web (Servlet) Container like WebSphere Application Server (WAS) or Apache Tomcat.

This Proxy Servlet can be used in WBIMB to support HTTPS secure protocol when it is deployed in properly configured Web Containers. This enables Web Services requests to arrive to the message flows in WBIMB over a secure communications protocol.

With WBIMB and WMB the Proxy Servlet can be used to enhance the capabilities of the Message Broker internal HTTP listener. Deploying the Proxy Servlet on external Web Containers can provide larger number of concurrent HTTP sessions, high availability, load distribution and access to the Message Broker over multiple IP addresses and ports.

The remote deployment of the Proxy Servlet, on a Web Container running in a separate server from the Message Broker, is supported in WMB V6 Fix Pack 1 or later.



---

## **Bibliography**

- *IBM WebSphere Business Integration Message Broker Version 5.0 Toolkit – Information Center* - <http://publib.boulder.ibm.com/infocenter/wbihelp/v6rxmx/index.jsp>
- *IBM WebSphere Message Broker Version 6.0 Toolkit – Information Center* - <http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp>
- *Rational Application Developer – Information Center* - [http://publib.boulder.ibm.com/infocenter/rtnl0600/index.jsp?topic=/com.ibm.rational.rad.books/icwelcome\\_product\\_rad.htm](http://publib.boulder.ibm.com/infocenter/rtnl0600/index.jsp?topic=/com.ibm.rational.rad.books/icwelcome_product_rad.htm)
- *Tomcat Documentation* - <http://jakarta.apache.org/tomcat/tomcat-5.0-doc/index.html>

## Chapter 2. Overview

WebSphere Business Integration Message Broker V5 (WBIMB) introduced three new HTTP nodes (HTTP Input, HTTP Request and HTTP Reply) as part of the Web Services support. Message Broker has an internal HTTP listener function (in port 7080 by default). The WBIMB HTTP listener does not support SSL or HTTPS but the WMB listener does support it.

This SupportPac is a Java Proxy Servlet that implements the function of the Message Broker HTTP listener in an external Servlet Container (for example, WebSphere Application Server (WAS) or Apache Tomcat). Once the Proxy Servlet is deployed and running on the Servlet Container then it uses the HTTP listener of the container to receive HTTP requests. If the Servlet Container is configured to support HTTPS (SSL) then Web Services requests are received by the message flows using a secure communications protocol.

The following figure shows the components required for the implementation of the Proxy Servlet.

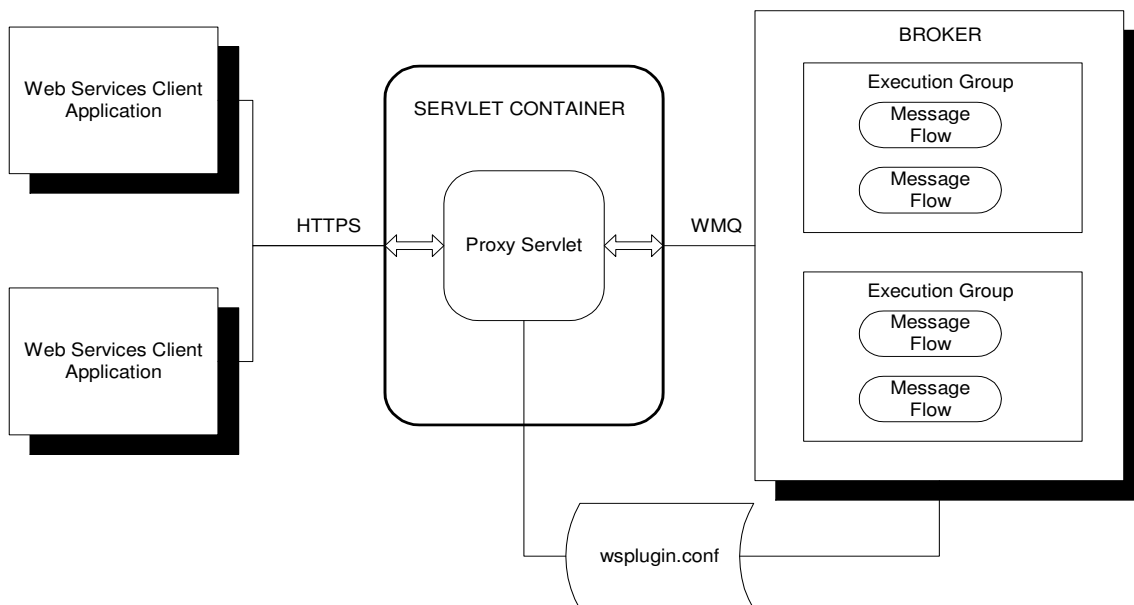


Figure 1 - Proxy Servlet Implementation

### Description of the solution components

The following are the components required by this SupportPac:

#### Web Services Clients

These are applications that send and receive SOAP (or simple HTTP) requests and responses to and from Web Services implemented in message flows running under a Message Broker.

## Servlet Container

This is the runtime environment for Servlets and Java Server Pages (JSP). WAS and Tomcat are two examples of Servlet Containers available in the market. The Proxy Servlet requires that the Servlet Container runs on the same server as the Message Broker for WBIMB. Remote deployment of the Proxy Servlet is supported in WMB V6 Fix Pack 1 or later. It is a requirement that the Servlet Container allows the Proxy Servlet to call the WebSphere MQ classes for java.

It is the Servlet Container that provides the HTTPS (SSL) listener support. HTTPS must be configured and available in the container. The Proxy Servlet is not aware whether HTTP or HTTPS is used to receive the Web Services requests.

## Proxy Servlet

This is the Java code developed for this SupportPac. It is a Java Servlet that receives HTTP requests, it matches the received URL with the URLs that the HTTP Input nodes in any of the message flows are monitoring and then it passes the request to the correct HTTP Input node flow using WebSphere MQ. The Proxy Servlet receives response messages from the HTTP Reply nodes and sends them back to the Client Applications over HTTP or HTTPS. The Message Broker has several internal WMQ queues (“SYSTEM.BROKER.WS.\*”) that are used for the communication between the Proxy Servlet and the HTTP Input and Reply nodes.

The Proxy Servlet has a “many” to “one” affinity with the Message Brokers (Many Proxy Servlets can service one broker but one Servlet can only service one broker).

Each HTTP Input node monitors the arrival of requests associated to specific URLs. The Message Broker has an internal configuration file (or an internal WMQ queue in WMB V6 Fix Pack1) that contains the list of URLs that are monitored by the different HTTP Input nodes in the different message flows deployed in any of the execution groups. The Proxy Servlet accesses this internal file (or queue) to match the URLs received in HTTP/HTTPS requests with the URLs that the HTTP Input nodes that are waiting for. The configuration file (or queue) has a unique correlation ID associated with each URL. The HTTP Input node uses this correlation ID to get the messages from the internal queue (“SYSTEM.BROKER.WS.INPUT”) and the Proxy Servlet uses the same correlation ID to put the messages on this queue. This is the mechanism used to correlate the incoming HTTP/HTTPS requests and the HTTP Input nodes in the message flows.

The HTTP Input node copies the MQ input Message ID in the LocalEnvironment.Destination.HTTP.RequestIdentifier to be used by the HTTP Reply node for MQ output message Correlation ID. The Proxy Servlet does a selective get by correlation ID to the reply queue (“SYSTEM.BROKER.WS.REPLY”) and it receives the response messages.

The Proxy Servlet accepts “GET” and “POST” HTTP/HTTPS requests.

## URLs

The Universal Resource Locator (URL) has an important role when HTTP/HTTPS protocols are used. In Message Broker, each HTTP Input node expects to receive requests for a specific URL (or URLs when wild card characters are used). The Servlet Container also uses the URLs to locate the Servlets that are going to process the HTTP/HTTPS requests received by the listener in the Container.

The Proxy Servlet passes the requests from the Servlet Container to Message Broker and vice versa. URLs have the dual function of locating Servlets and locating HTTP Input nodes. This affects the format of the URLs used for Message Broker.

Any URL has the following format:

```
<schema>://<hostname>:<port>/<url_path>
```

Where:

<schema>. It is “http” or “https”.

<hostname>. It is the hostname (or IP address) of the server where the Servlet Container is running (and it should be the same server as for the Message Broker).

<port>. This is the port number that the Servlet Container is listening on.

<url\_path>. It is a series of tokens separated by slashes (“/”). This is used to indicate the location of the Servlet and the location of the HTTP Input nodes.

Since the “<url\_path>” is used for mapping two resources (instead of one with the Message Broker internal listener) then the format of the URL will change when the Proxy Servlet is used.

The Message Broker format of the “<url\_path>” is:

```
/<url_path> = /<context root>/<node_url_path>
```

Where:

<context root>. It is the <url\_path> allocated to the Proxy Servlet by the container when the Servlet is installed and deployed.

<node\_url\_path>. This is the part of URL path that is added to make the URL unique to a specific HTTP Input node.

The entire “<url\_path>” has to be configured in the properties of the HTTP Input node.

**Note.** In some Servlet Containers, it is possible to configure the Proxy Servlet to receive all the HTTP/HTTPS requests arriving to the container (Context Root = “/\*”). In this case, the existing URLs in the HTTP nodes do not have to change when the Proxy Servlet is implemented.

## Message Broker

The Message Broker is the runtime environment for the message flows. Message Flows can receive requests over HTTP when there are HTTP Input nodes in the flow. The Message Broker has its own internal HTTP listener but in WBIMB V5.0 this listener can only receive HTTP requests and not HTTPS (SSL) requests.

The Proxy Servlet implements the HTTP/HTTPS listener function in an external Servlet Container.

### *Message Flows*

Message flows execute the functions of transforming, logging, enriching and routing messages. Message Flows can use HTTP Input nodes to receive requests and HTTP Reply nodes to send responses. If the message flows implement Web Services then the HTTP Input nodes receive SOAP request messages. Each HTTP Input node has a URL configured in the node properties. The HTTP Input nodes receive the request messages addressed to the configured URL.

The Proxy Servlet is not aware of the execution groups that the message flows are deployed.

---

## Chapter 3. Installing the Proxy Servlet

### SupportPac contents

The supplied “ie01.zip” file should be unzipped in a temporary Windows 2000/XP directory. The following are the files created:

- \ProxyServlet\IE01ProxyServletUserGuideV2.pdf (This document)
- \ProxyServlet\IE01.war (Proxy Servlet WAR file)
- \ProxyServlet\TestHttps (Message Broker Toolkit project with sample flow)
- \ProxyServlet\ws-request.txt (Sample HTTP request message)

The “IE01.war” file should be deployed to the server where the Servlet Container is running.

### Platforms supported

This SupportPac runs on any platform that is supported by the Servlet Container.

This user guide describes the installation on Windows XP but the Proxy Servlet could be installed on other platforms.

### Prerequisites

This SupportPac requires:

- WBIMB V5.0 Fix Pack 3 or later.
- WMB V6.0 Fix Pack 1 or later. (WMB V6.0 GA does not support running the Proxy Servlet on a separate server from the Message Broker).
- WMQ V5.3 Fix Pack 5 (WMQ Classes for Java) or later.
- A Servlet Container and Web Server (WAS V5, WAS V6 or Apache Tomcat V5).

### Installing the Proxy Servlet

This Proxy Servlet was tested with RAD V6.0.1 (WebSphere V6.0 Server) and with Apache Tomcat V5.5.15 on Windows XP.

This document describes the installation process with Apache Tomcat 5. The installation with any other Servlet Container should be very similar to what is described here.

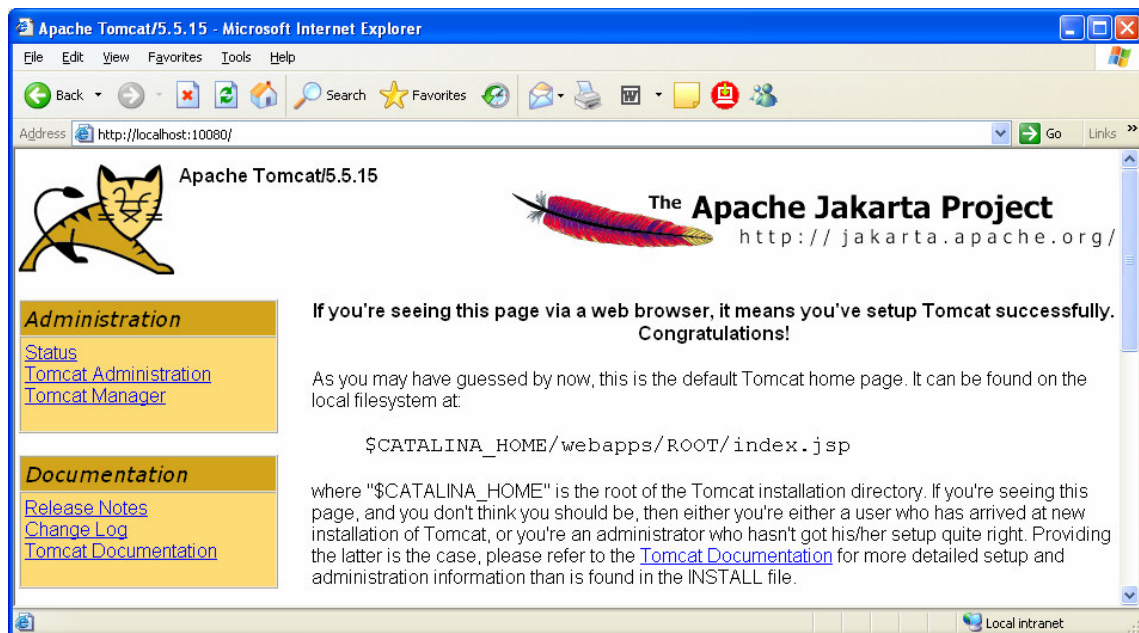
#### Installation on Apache Tomcat 5

It is assumed that the Message Broker is installed and running on the same server that Apache Tomcat is going to be installed. A Java SDK that is compatible with Apache Tomcat V5 must be installed.

#### *Download and installation of Apache Tomcat 5*

- Download the file “jakarta-tomcat-5.5.15.exe” from the following URL  
<http://tomcat.apache.org/download-55.cgi>

- Check that the environment variable “JAVA\_HOME” is defined and it points to the Java SDK installation directory.
- Execute the file “jakarta-tomcat-5.5.15.exe” to run the installer.
  - Accept license agreement.
  - Select “Custom” type of install and check the Service box.
  - Specify the destination folder for Tomcat. For example, “C:\Tomcat5”.
  - Specify HTTP listener port. For example, “10080”.
  - Specify the administrator userid and password. For example “admin”.
  - Specify the JVM installation directory. For example, “C:\Java142”. (Note: If using Java 1.4.2 install the compatibility patches “apache-tomcat-5.5.15-compat.zip”)
  - Click on “Install” to initiate the installation process.
- Restart Windows.
- Apache Tomcat (Tomcat5) should be running as a Windows Service.
- Use Internet Explorer and enter the following URL: <http://localhost:10080/>. This should display Tomcat’s home page.



**Figure 2 - Tomcat home page**

### ***Installation of the WMQ 5.3 Java libraries in Tomcat***

This Proxy Servlet uses WMQ to communicate with the WBIMB broker. It is required to install copies of the following “JAR” files in Tomcat.

Copy from “<WMQ\_Install\_Dir>\Java\lib” to “<Tomcat\_Install\_Dir>\shared\lib” the following files:

- com.ibm.mq.jar

- connector.jar

### **Configuration of the Proxy Servlet**

Configure the Servlet initialization parameters and URL mappings in the Web Deployment Descriptor (web.xml) before this Proxy Servlet can be deployed to the Servlet Container.

The following initialization parameters are used by the Proxy Servlet and they need to be configured for the environment that the Servlet is going to run:

1. **brokerName.** The name of the Message Broker associated with this instance of the Proxy Servlet.
2. **configFilePath.** The paths to the HTTP node configuration file (wsplugin.conf in WBIMB or wsplugin6.conf in WMB). In Windows, the file is found in "C:\[wbimb\_install\_dir]\config\wsplugin.conf" for WBIMB or in "C:\Documents and Settings\All Users\Application Data\IBM\MQSI\components\[broker\_name]\config\wsplugin6.conf" for WMB. In Unix, the file is found in "/var/mqsi/config/wsplugin.conf" in WBIMB or in "/var/mqsi/components/[broker\_name]/config/wsplugin6.conf" for WMB.
3. **turnTraceOn.** To enable debug trace on standard out for problem determination. Enter 0 for no debug, 1 for normal and 2 for debug.
4. **useClientMode.** To specify if MQ Client or MQ Bindings connection is used. MQ Client should be used when the Proxy Servlet is running in a separate server from the Message Broker. Specify "true" or "false".
5. **clientModeChannelName.** This is name of the MQ SVRCONN channel used to connect to the Message Broker queue manager.
6. **clientModeChannelHostname.** This is the server hostname (or IP address) where the Message Broker queue manager is running.
7. **clientModeChannelPortNumber.** This is the Message Broker queue manager listening port.
8. **useQueueManagerDataInsteadOfConfigFile.** Use this if you are running WMB V6 Fix Pack 1 or later. It configures the Proxy Servlet to use an MQ Queue instead of the Configuration File for the URL correlation data. Enter the name of the Message Broker queue manager or leave blank to use the file.
9. **sleepBeforeGet.** This is a performance tuning parameter. It sends the Proxy Servlet thread to sleep before executing an MQGET to retrieve the response message from the reply queue. Specify the number of seconds to sleep. This becomes the minimum response time for HTTP requests.
10. **disconnectBeforeSleep.** This is a performance tuning parameter. It forces the Proxy Servlet to disconnect from the queue manager (release the connection back to the pool) before sleeping. Specify "true" or "false".
11. **useFastpathBindingsConnection.** This is a performance tuning parameter. When MQ Bindings connection is used (not MQ client), it is possible to connect using "fastpath" bindings. Use this parameter with care; it can compromise the integrity of the queue manager. Specify "true" or "false".

Define the URL mappings supported by the Proxy Servlet. It is possible to define one or more mappings. These mappings are used by the Servlet Container to filter URL requests before executing the Proxy Servlet.

The mappings are the “/(<node\_url\_path>” paths described in section “URLs” on page 2 (/<url\_path> = /<context\_root>/<node\_url\_path>). It is possible to define a “/(<node\_url\_path” equal to “/\*” to accept URL paths similar to this example: “/IE01/anything\_here”.

The SupportPac has the following sample URL mappings defined:

- /\*

The initialization parameters and the URL mappings are in the “Web Deployment Descriptor” of the Servlet (WEB-INF/web.xml). Unzip the “IE01.war” file to find the “WEB-INF/web.xml” file.

The following is the sample “web.xml” file shipped with this SupportPac:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
<web-app id="WebApp">
  <!-- Proxy servlet for WBIMB HTTP input nodes -->
  <display-name>WBIMB Proxy Servlet V2 - SupportPac IE01</display-name>
  <servlet>
    <servlet-name>WBIMBServlet</servlet-name>
    <display-name>WBIMBServletV2</display-name>
    <servlet-class>
      com.ibm.broker.httpproxy.WBIMBServlet
    </servlet-class>
    <init-param>
      <param-name>brokerName</param-name>
      <!-- Set the message broker name here -->
      <param-value> WBIMBBK1</param-value>
      <description>Set the message broker name here</description>
    </init-param>
    <init-param>
      <param-name>configFilePath</param-name>
      <!-- This would be /var/mqsi/config/wsplugin.conf on UNIX -->
      <param-value>
        C:\\wbimb\\config\\wsplugin.conf
      </param-value>
      <description>
        This would be
        /var/mqsi/components/[broker_name]/config/wsplugin6.conf
        on UNIX
        or
        C:\\Documents and Settings\\All Users
        \\Application Data\\IBM\\MQSI\\components
        \\[broker_name]\\config\\wsplugin6.conf
        on Windows for WMB V6
        and
        /var/mqsi/config/wsplugin.conf on UNIX
        or
        C:\\[wbimb_install_dir]\\config\\wsplugin.conf on
        Windows for WBIMB V5
      </description>
    </init-param>
    <init-param>
      <param-name>turnTraceOn</param-name>
      <!-- Set 0 for no trace, 1 for normal or 2 for debug -->
      <param-value>2</param-value>
      <description>
        Set 0 for no trace, 1 for normal or 2 for debug
      </description>
    </init-param>
    <init-param>
      <param-name>useClientMode</param-name>
      <param-value>>false</param-value>
      <description>
        Use MQ client (true) or bindings connection (false)
      </description>
    </init-param>
  </servlet>
</web-app>
```



```

        </description>
    </init-param>
    <init-param>
        <param-name>clientModeHostname</param-name>
        <param-value>localhost</param-value>
        <description>
            Hostname or IP for the Queue Manager
        </description>
    </init-param>

    <init-param>
        <param-name>clientModeChannelName</param-name>
        <param-value>SYSTEM.DEF.SVRCONN</param-value>
        <description>MQ SVRCONN channel name</description>
    </init-param>
    <init-param>
        <param-name>clientModePortNumber</param-name>
        <param-value>31002</param-value>
        <description>MQ Listener port number</description>
    </init-param>

    <init-param>
        <param-name>
            useQueueManagerDataInsteadOfConfigFile
        </param-name>
        <!--Queue manager name or blank for none -->
        <param-value></param-value>
        <description>
            Queue manager name (remote proxy) or blank for none
            (local proxy). Supported on WMB6 Fix Pack 1 or later.
        </description>
    </init-param>

    <init-param>
        <param-name>sleepBeforeGet</param-name>
        <param-value>0</param-value>
        <description>Sleep time in seconds</description>
    </init-param>

    <init-param>
        <param-name>disconnectBeforeSleep</param-name>
        <param-value>>false</param-value>
        <description>
            To release MQ connection while sleeping
        </description>
    </init-param>

    <init-param>
        <param-name>useFastpathBindingsConnection</param-name>
        <param-value>>false</param-value>
        <description>
            Used when useClientMode is false. Be careful with this
            option it can compromise the integrity of the queue
            manager.
        </description>
    </init-param>
</servlet>
<servlet-mapping>
    <servlet-name>WBIMBServlet</servlet-name>
    <url-pattern>/*</url-pattern>
</servlet-mapping>
<welcome-file-list>
    <welcome-file>index.html</welcome-file>
</welcome-file-list>
</web-app>

```

Edit this file using “notepad” or any other tool that enables the editing of WAR files and/or Web Deployment Descriptors.

## ***Proxy Servlet configuration using RAD***

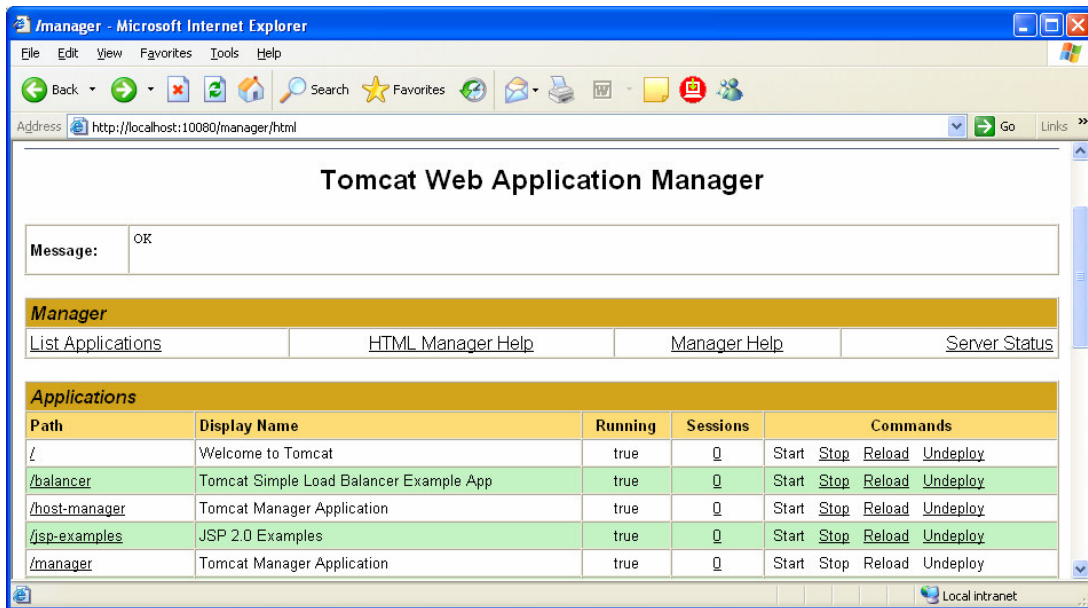
Rational Application Developer (RAD) is the recommended tool to configure this Proxy Servlet. The following are the steps to configure this SupportPac using RAD:

- Start RAD.
- Import the WAR file (File -> Import -> WAR File -> Next -> enter names for Servlet project ("IE01"), context root ("IE01") and Enterprise application project ("IE01EAR") -> Finish).
- Add the WMQ libraries to the Servlet project properties (Right click on the Servlet project -> Properties -> Java Build Path -> Libraries -> Add external JARs -> Look into <WMQ\_Install\_Dir>\Java\lib -> Select "com.ibm.mq.jar" -> Open -> OK). This step is only required if the Proxy Servlet is deployed in a WAS 6 Server.
- The Servlet is ready for configuration using the Business Integration perspective (Double click on file "web.xml" on folder "<Servlet Project>\Web Content\WEB\_INF" -> Click on the Servlet "WBIMBServlet" -> Edit URL mappings and Initialization parameters -> Save Web Deployment Descriptor).
- Now is possible to deploy the Proxy Servlet in a WebSphere V6 server to test it.
  - To run the Servlet, it is necessary to change the WebSphere environment variable "MQ\_INSTALL\_ROOT" to point to "<WMQ\_Install\_Dir>". Do not use the MQ jars provided with WebSphere.
  - To change the environment variable, start WAS → run the administration console → log on → Select Environment → WebSphere Variables → Double click on "MQ\_INSTALL\_ROOT" → Enter Value = "<WMQ\_Install\_Dir>" → OK → Save WAS configuration changes → restart WAS.
- Or it is also possible to export the WAR file and deploy it in a Tomcat Container (File -> Export -> WAR file -> Next -> Enter the name of the Servlet project and the name of output WAR file -> Finish).

## ***Installation and deploy of the Proxy Servlet***

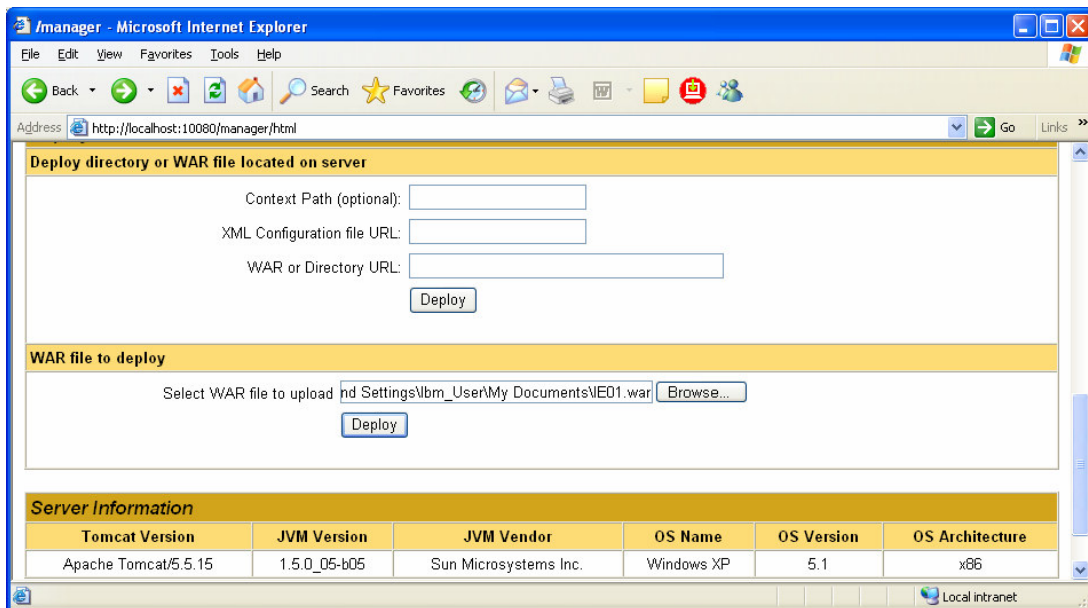
The following are the steps to load and install the "IE01.war" file in Tomcat:

- Start Internet Explorer and display the Tomcat home page (<http://localhost:10080/>).
- Click on "Tomcat Manager" and enter the "admin" userid and password.



**Figure 3 - Tomcat Manager**

- Scroll down to the section “WAR file to deploy” and click on “Browse” to search for the “IE01.war” file.



**Figure 4 - WAR file to deploy**

- Click on “Deploy”.

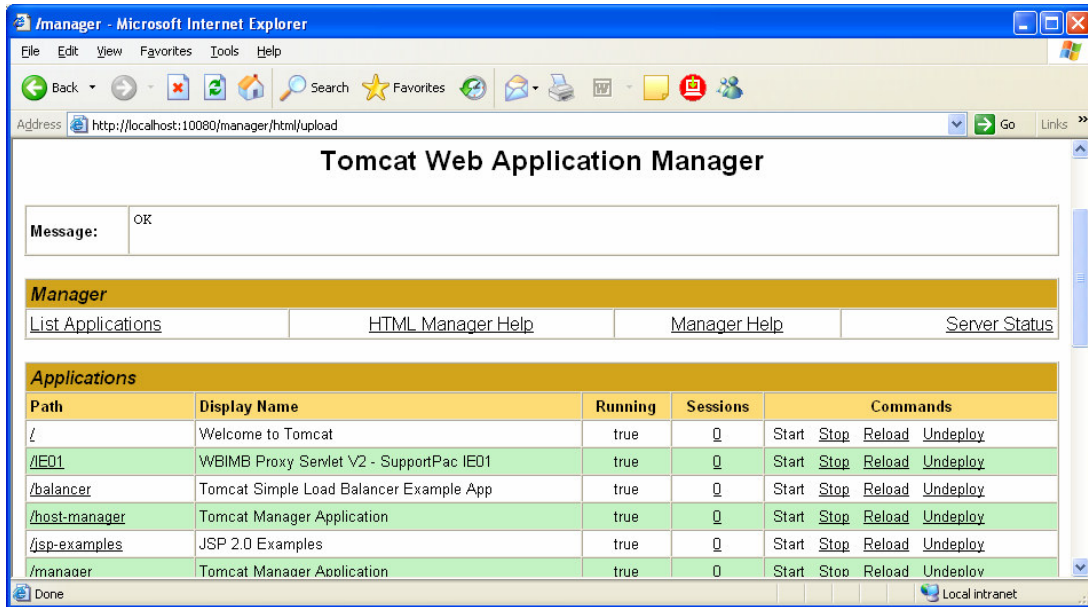


Figure 5 - Servlet deployed

- Note that the “context root” associated to the Proxy Servlet is the same as the WAR filename. It is possible to rename the “.war” file before deploying it to Tomcat to have a different “context root”.
- Click on “/IE01” on the Applications view to test if the Servlet is active.

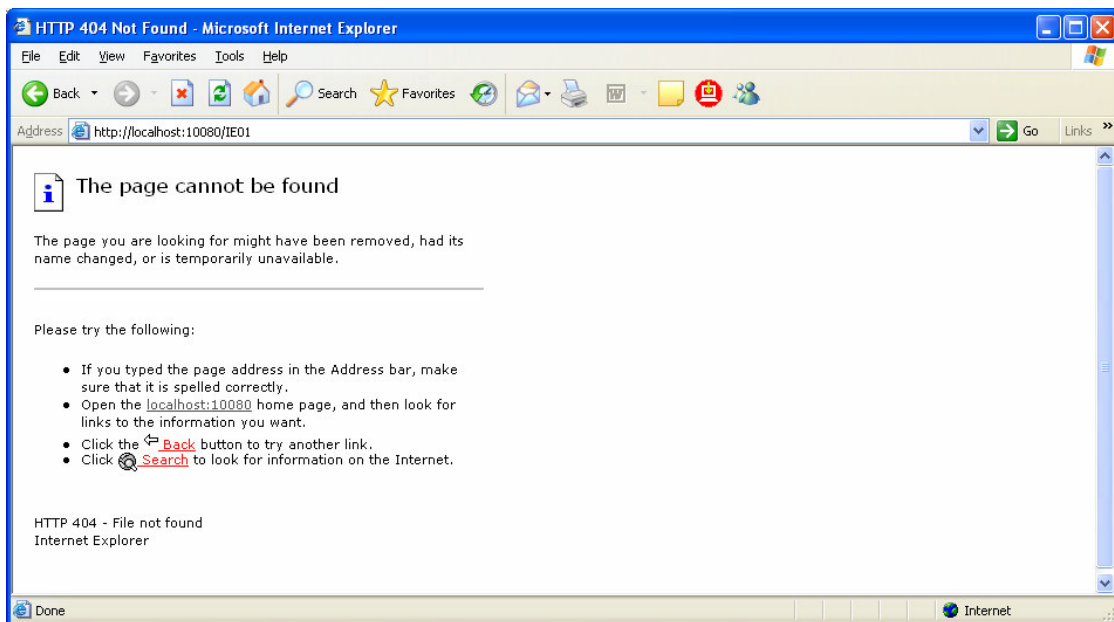
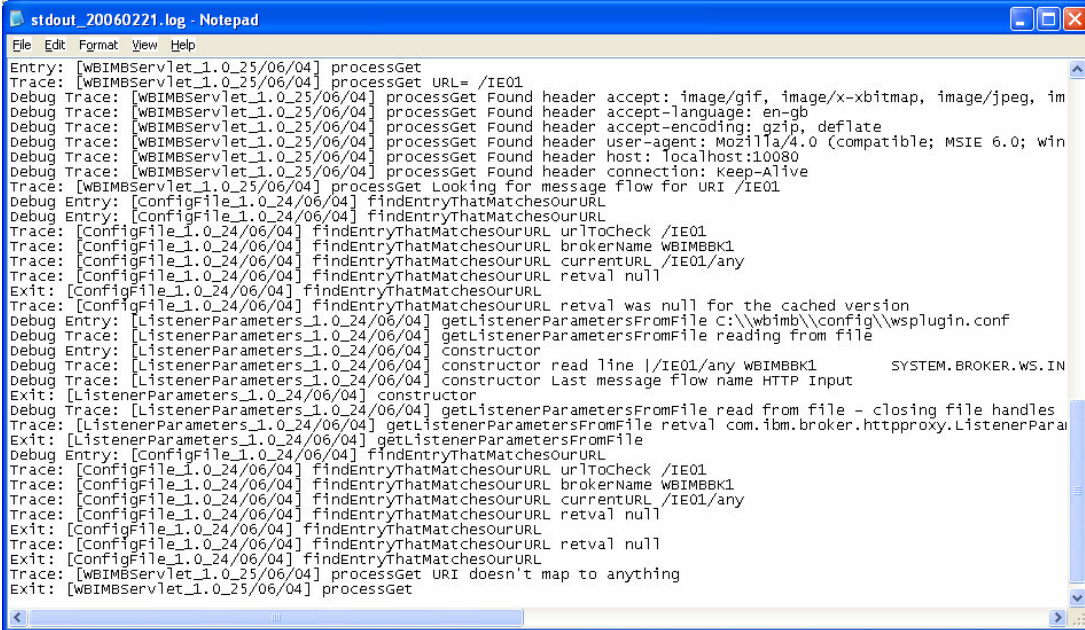


Figure 6 - Proxy Servlet test response

If the Proxy Servlet trace is enabled then the log will show that the Servlet has received the request and rejected it. The stdout log is found on “C:\Program Files\Apache Software Foundation\Tomcat 5.5\logs”.



```

Entry: [WBIMBServlet_1.0.25/06/04] processGet
Trace: [WBIMBServlet_1.0.25/06/04] processGet URL= /IE01
Debug Trace: [WBIMBServlet_1.0.25/06/04] processGet Found header accept: image/gif, image/x-bitmap, image/jpeg, im
Debug Trace: [WBIMBServlet_1.0.25/06/04] processGet Found header accept-language: en-gb
Debug Trace: [WBIMBServlet_1.0.25/06/04] processGet Found header accept-encoding: gzip, deflate
Debug Trace: [WBIMBServlet_1.0.25/06/04] processGet Found header user-agent: Mozilla/4.0 (compatible; MSIE 6.0; win
Debug Trace: [WBIMBServlet_1.0.25/06/04] processGet Found header host: localhost:10080
Debug Trace: [WBIMBServlet_1.0.25/06/04] processGet Found header connection: keep-alive
Trace: [WBIMBServlet_1.0.25/06/04] processGet Looking for message flow for URI /IE01
Debug Entry: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL
Debug Entry: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL
Trace: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL urlTocheck /IE01
Trace: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL brokerName WBIMBBK1
Trace: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL currentURL /IE01/any
Trace: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL retval null
Exit: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL
Trace: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL retval was null for the cached version
Debug Entry: [ListenerParameters_1.0.24/06/04] getListenerParametersFromFile C:\wbimb\config\wsplugin.conf
Debug Trace: [ListenerParameters_1.0.24/06/04] getListenerParametersFromFile reading from file
Debug Entry: [ListenerParameters_1.0.24/06/04] constructor
Debug Trace: [ListenerParameters_1.0.24/06/04] constructor read line /IE01/any WBIMBBK1 SYSTEM.BROKER.WS.IN
Debug Trace: [ListenerParameters_1.0.24/06/04] constructor Last message flow name HTTP Input
Exit: [ListenerParameters_1.0.24/06/04] constructor
Debug Trace: [ListenerParameters_1.0.24/06/04] getListenerParametersFromFile read from file - closing file handles
Trace: [ListenerParameters_1.0.24/06/04] getListenerParametersFromFile retval com.ibm.broker.httpproxy.ListenerPara
Exit: [ListenerParameters_1.0.24/06/04] getListenerParametersFromFile
Debug Entry: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL
Trace: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL urlTocheck /IE01
Trace: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL brokerName WBIMBBK1
Trace: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL currentURL /IE01/any
Exit: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL retval null
Trace: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL
Exit: [ConfigFile_1.0.24/06/04] findEntryThatMatchesourURL retval null
Trace: [WBIMBServlet_1.0.25/06/04] processGet URI doesn't map to anything
Exit: [WBIMBServlet_1.0.25/06/04] processGet

```

Figure 7 - Proxy Servlet trace output

- Now the Proxy Servlet is ready to be tested with a Message Broker receiving HTTP (no HTTPS) requests and passing them to a message flow..

## Configuration of Apache Tomcat for HTTPS (SSL)

Refer to “SSL Configuration HOW-TO” in chapter 12 “SSL” in the Tomcat Documentation.

There are two basic steps to enable SSL in Tomcat:

1. Create a certificate keystore and a self-signed certificate (for testing purposes only):
  - a. In Windows command prompt, change directory to <tomcat\_install\_dir>.
  - b. Execute the following command:

```
%JAVA_HOME%bin/keytool -genkey -alias tomcat -keyalg RSA -keystore  
<tomcat_install_dir>.keystore
```

**Note:** Make sure that %JAVA\_HOME% and <tomcat\_install\_dir> do not have any spaces in the directory names. You must use the “short” directory names. Use the command “dir /x” to display the short directory names. For example, “Program Files” should change to “PROGRA~1”.

```

C:\Tomcat5>c:\Java142\bin\keytool -genkey -alias tomcat -keyalg RSA -keystore
c:\tomcat5\.keystore
Enter keystore password: changeit
What is your first and last name?
  [Unknown]: Vicente Suarez
What is the name of your organizational unit?
  [Unknown]: ISSW
What is the name of your organization?
  [Unknown]: IBM
What is the name of your City or Locality?
  [Unknown]: Hursley
What is the name of your State or Province?
  [Unknown]: Hampshire
What is the two-letter country code for this unit?
  [Unknown]: UK
Is CN=Vicente Suarez, OU=ISSW, O=IBM, L=Hursley, ST=Hampshire, C=UK correct?
(type "yes" or "no")
  [no]: yes

```

```
Enter key password for <tomcat>
(RETURN if same as keystore password):

C:\Tomcat5>
```

2. Uncomment the “SSL HTTP/1.1 Connector” entry in “<tomcat\_install\_dir>/conf/server.xml” and update as necessary.

The following is the entry used when Tomcat is running on the IBM JVM 1.4.2:

```
<!-- Define a SSL HTTP/1.1 Connector on port 10443 -->

<Connector port="10443"
keystore="C:\Program Files\Apache Software Foundation\Tomcat 5.5\keystore"
algorithm="IbmX509"
maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true" debug="0"
clientAuth="false" sslProtocol="SSL" />
```

The following is the entry used when Tomcat is running on the Sun JVM 1.5:

```
<!-- Define a SSL HTTP/1.1 Connector on port 8443 -->

<Connector port="10443"
keystore="C:\Program Files\Apache Software Foundation\Tomcat 5.5\keystore"
maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true" debug="0"
clientAuth="false" sslProtocol="TLS" />
```

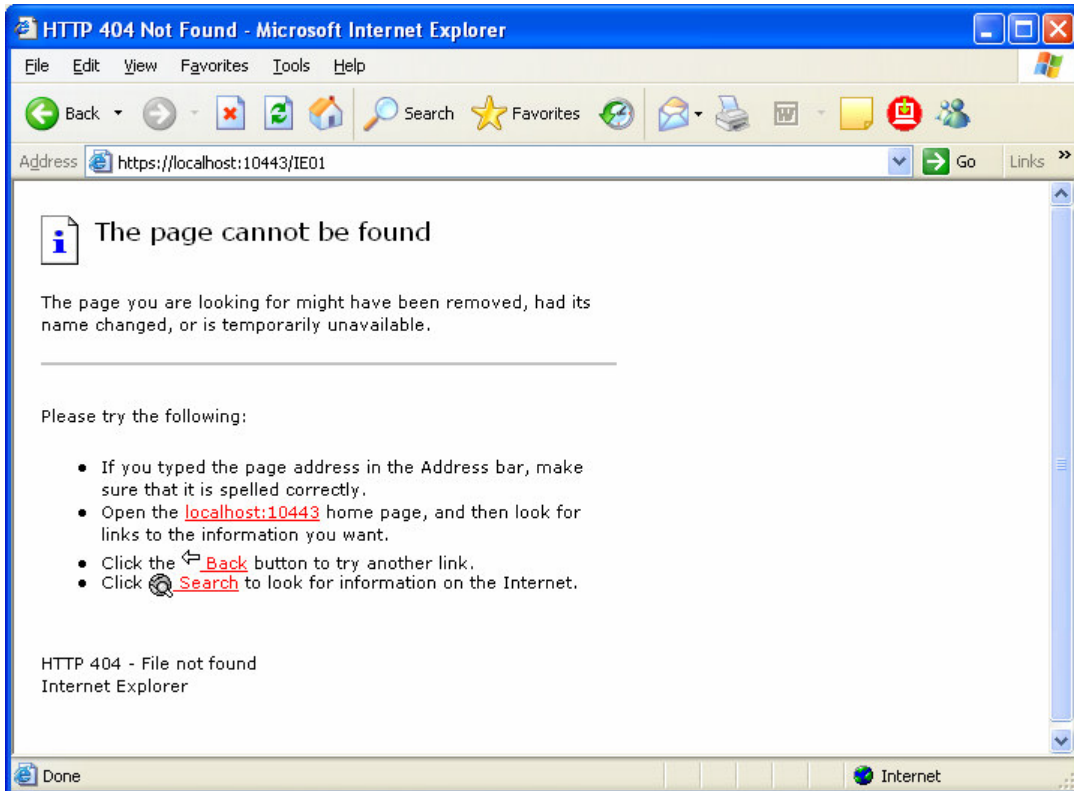
3. Stop and restart Tomcat. Now, the Servlet Container is ready to receive HTTPS (SSL) requests for the Proxy Servlet (on port 10443 in our test system) and pass the requests to message flows in a Message Broker.
4. Enter URL <https://localhost:10443/IE01> using a browser to test the request is getting to the Proxy Servlet:



**Figure 8 - Tomcat certificate verification**

5. Enter “Yes” to accept the certificate.





6. If trace has been enabled in the Proxy Servlet then check the stdout log to verify that the request has been received:

```

stdout_20060221.log - Notepad
File Edit Format View Help
Exit: [WBIMBServlet_1.0_25/06/04] processGet
Entry: [WBIMBServlet_1.0_25/06/04] processGet
Trace: [WBIMBServlet_1.0_25/06/04] processGet URL= /IE01
Debug Trace: [WBIMBServlet_1.0_25/06/04] processGet Found header accept: image/gif, image/x-xbitmap, image/jpeg, im
Debug Trace: [WBIMBServlet_1.0_25/06/04] processGet Found header accept-language: en-gb
Debug Trace: [WBIMBServlet_1.0_25/06/04] processGet Found header accept-encoding: gzip, deflate
Debug Trace: [WBIMBServlet_1.0_25/06/04] processGet Found header user-agent: Mozilla/4.0 (compatible; MSIE 6.0; win
Debug Trace: [WBIMBServlet_1.0_25/06/04] processGet Found header host: localhost:10443
Debug Trace: [WBIMBServlet_1.0_25/06/04] processGet Found header connection: keep-alive
Trace: [WBIMBServlet_1.0_25/06/04] processGet Looking for message flow for URI /IE01
Debug Entry: [ConfigFile_1.0_24/06/04] findEntryThatMatchesourURL
Debug Entry: [ConfigFile_1.0_24/06/04] findEntryThatMatchesourURL
Trace: [ConfigFile_1.0_24/06/04] findEntryThatMatchesourURL urlTocheck /IE01
Trace: [ConfigFile_1.0_24/06/04] findEntryThatMatchesourURL brokerName WBIMBBK1
Trace: [ConfigFile_1.0_24/06/04] findEntryThatMatchesourURL retval null
Exit: [ConfigFile_1.0_24/06/04] findEntryThatMatchesourURL
Trace: [ConfigFile_1.0_24/06/04] findEntryThatMatchesourURL retval was null for the cached version
Debug Entry: [ListenerParameters_1.0_24/06/04] getListenerParametersFromFile c:\wibmb\config\wsplugin.conf
Debug Trace: [ListenerParameters_1.0_24/06/04] getListenerParametersFromFile reading from file
Debug Entry: [ListenerParameters_1.0_24/06/04] constructor
Debug Trace: [ListenerParameters_1.0_24/06/04] constructor read line /IE01/any WBIMBBK1 SYSTEM.BROKER.WS.IN
Debug Trace: [ListenerParameters_1.0_24/06/04] constructor Last message flow name HTTP Input
Exit: [ListenerParameters_1.0_24/06/04] constructor
Debug Trace: [ListenerParameters_1.0_24/06/04] getListenerParametersFromFile read from file - closing file handles
Trace: [ListenerParameters_1.0_24/06/04] getListenerParametersFromFile retval com.ibm.broker.httpproxy.ListenerPara
Exit: [ListenerParameters_1.0_24/06/04] getListenerParametersFromFile
Debug Entry: [ConfigFile_1.0_24/06/04] findEntryThatMatchesourURL
Trace: [ConfigFile_1.0_24/06/04] findEntryThatMatchesourURL urlTocheck /IE01
Trace: [ConfigFile_1.0_24/06/04] findEntryThatMatchesourURL brokerName WBIMBBK1
Trace: [ConfigFile_1.0_24/06/04] findEntryThatMatchesourURL retval null
Exit: [ConfigFile_1.0_24/06/04] findEntryThatMatchesourURL
Trace: [ConfigFile_1.0_24/06/04] findEntryThatMatchesourURL
Exit: [ConfigFile_1.0_24/06/04] findEntryThatMatchesourURL
Trace: [WBIMBServlet_1.0_25/06/04] processGet URI doesn't map to anything
Exit: [WBIMBServlet_1.0_25/06/04] processGet

```

---

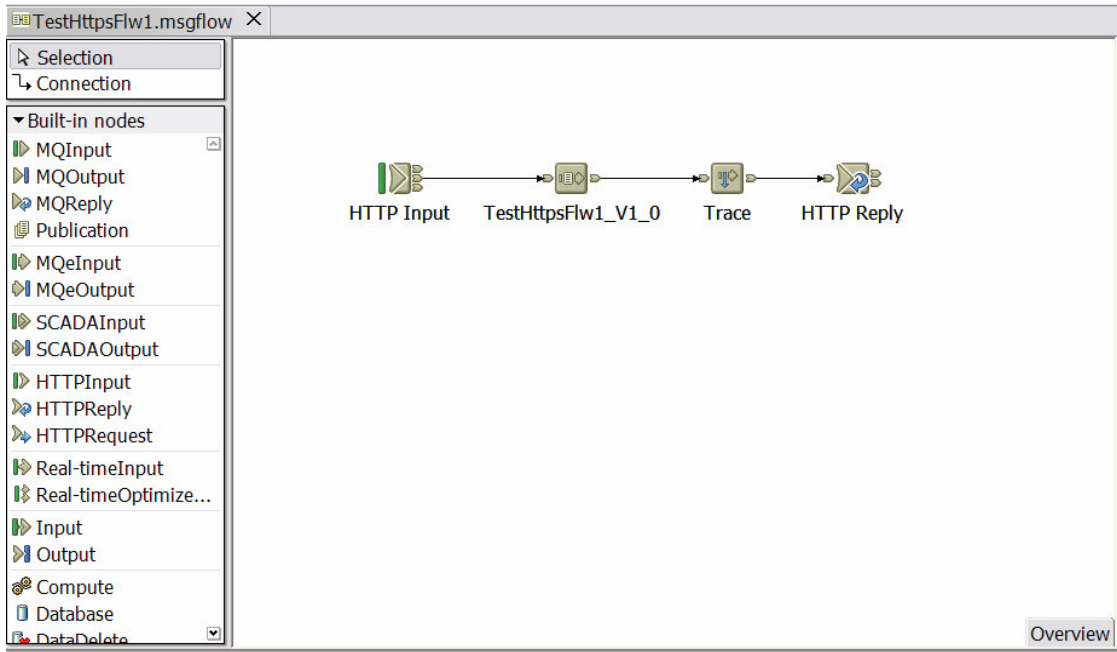
## Chapter 4. Using the Proxy Servlet

**Note:** At the moment of writing this document, there were no standard SSL test programs that could be recommended from searching the internet. The recommendation is to use an existing Web Services client application or to write your own SSL test client application using java.

The following are the components that are required to test the Proxy Servlet:

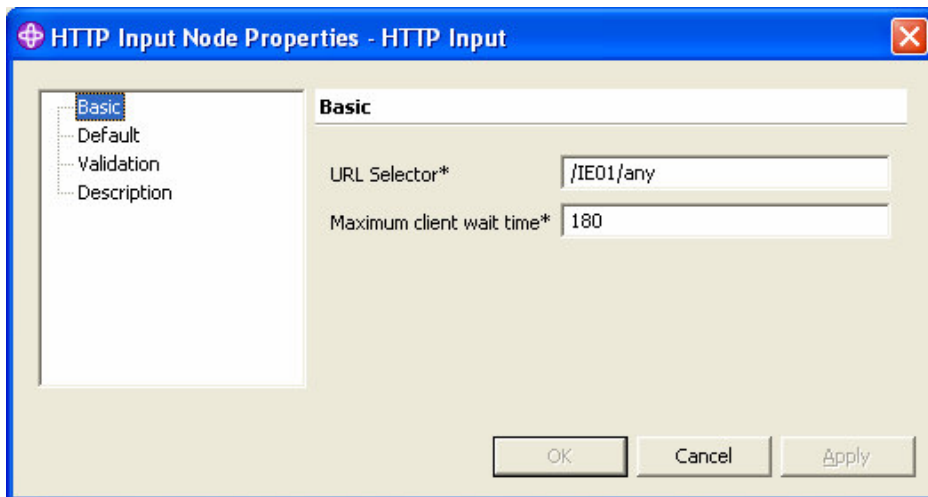
- A client application that can send HTTP/HTTPS requests. There are several options that can be used here:
  - “Netcat” is a shareware utility from the Internet ([http://www.atstake.com/research/tools/network\\_utilities/](http://www.atstake.com/research/tools/network_utilities/)). It can send the contents of a file to a HTTP listener. It does not support SSL.
  - “Nettool” is an open source graphical utility from the internet (<http://www.nettool.org>). This tool allows to send HTTP requests or to tunnel a request.
  - There is another shareware utility from the Internet called “stunnel” (<http://www.stunnel.org>) that can be used in combination with “Netcat” to send HTTPS requests to the Web container.
  - A Message Broker flow that has a HTTP request node. It can generate and send HTTP requests to a HTTP listener. It does not support SSL in WBIMB V5 but it does in WMB V6.
  - A Web browser using web pages or JSPs that can send HTTP POST requests. Most Web Browsers support HTTP/HTTPS.
  - A client application that sends requests using HTTP and/or HTTPS.
- A Message Broker. The name of this broker is set in one of the Proxy Servlet initialization parameters (brokerName) in the “web.xml” file.
- A message flow with HTTP Input and Reply nodes. This flow receives the messages from the Proxy Servlet and if a HTTP Reply is configured then responses are sent back to the Proxy Servlet. There is a Message Broker Toolkit sample project (“\IE01\ProxyServlet\TestHttps”) that can be imported to the workspace.





**Figure 9 - Sample Test Message Flow**

- HTTP Input nodes configured to monitor the correct URL.



**Figure 10 - URL Selector property of a HTTP Input node**

- Test messages. If “Netcat” is used, it requires a file with the HTTP request. The following is an example of a Web Services SOAP request message (found in “\IE01\ProxyServlet\ws-request.txt”):

```
POST /IE01/any HTTP/1.0
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: 520
SOAPAction: ""
Cookie: JSESSIONID=0000B5OSLFIUDMQZFAUXKHD5ZDQ:-1

<?xml version='1.0' encoding='UTF-8'?>
```

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
<ns1:getArticleByArchiveArticleID
xmlns:ns1="http://tempuri.org/forum.WSArticleGenerator" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<archiveID xsi:type="xsd:int">1234</archiveID>
</ns1:getArticleByArchiveArticleID>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

- This is the “Netcat” command for Windows “nc -w 60 localhost 10080 < ws-request.txt” or “nc -n 121.0.0.1 10080 < ws-request.txt” and the following is the text of the response:

```
HTTP/1.1 200 OK
```

```
Server: WebSphere Application Server/5.1
```

```
Content-Type: text/xml; charset=utf-8
```

```
Content-Length: 520
```

```
Content-Language: en-GB
```

```
Connection: close
```

```

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xm
lns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org
/2001/XMLSchema">
<SOAP-ENV:Body>
<ns1:getArticleByArchiveArticleID
xmlns:ns1="http://tempuri.org/forum.WSArticleGenerator" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<archiveID xsi:type="xsd:int">1234</archiveID>
</ns1:getArticleByArchiveArticleID>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

----- End of Document -----