JMS Publish/Subscribe Sample Application with WebSphere Application Server 5.x and Willow Technology’s WebSphere MQ

Introduction

In this document, we will demonstrate how to configure the application server for a Publish/Subscribe application. It is designed to be used in a test environment with WebSphere Application Server 5.x using Willow Technology's WebSphere MQ. The JMS application used in this tutorial was obtained from Sun's developer site at: http://java.sun.com/developer/EJTechTips/2003/tt0415.html.

This document assumes you have already intalled and configured your WebSphere MQ environment. Instead, it focuses on the installation and configuration of WebSphere Application Server as well as the creation of the system queues and preparing your MQ environment to use JMS Publish/Subscribe. Please read the Prerequisites section that follows to ensure that your test environment has been configured to work with this tutorial.

Tasks to be Completed

The following tasks are listed in the order for which they will be completed.

- Check Prerequisites For This Tutorial
- Start The Application Server And Verify Its Installation
- Set MQ Environment Variable In Websphere Application Server
- Prepare For MQ JMS Pub/Sub Implementation
- Configure Topic Connection Factory
- Configure Topic Destination
- Deploy The Web Application
- Edit And Run The Client Application
- Prepare For SSL

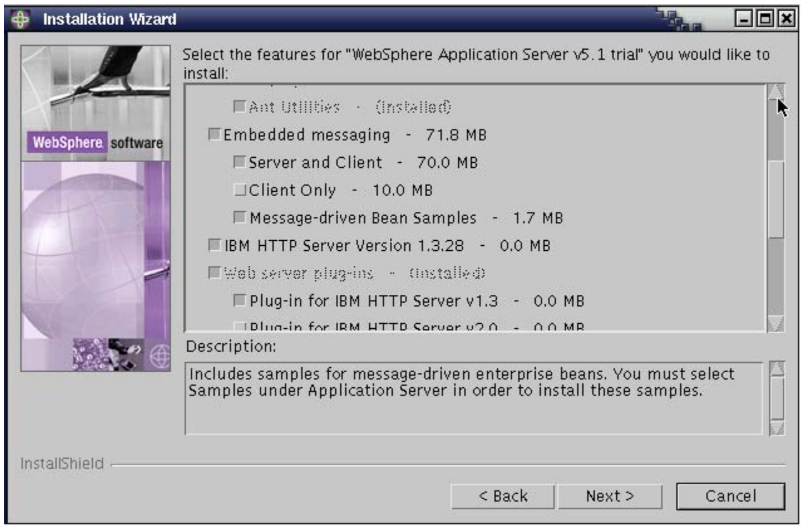
Prerequisites

Ensure that the following list of prerequisites have been met prior to beginning this tutorial.

- Install Willow's WebSphere MQ. Refer to your platform's version of WebSphere MQ documention for installing and configuring this product.

- Install WebSphere Application Server 5.x without Embedded Messaging. Refer to the installation instructions for installing this product.

NOTE: WebSphere Application Server installs Embedded Messaging by default. To ensure that this feature is not installed, select the Custom Install option from the Installation Wizard and de-select Embedded Messaging when the following window appears.



Select the features for "WebSphere Application Server v5.1 trial" you would like to install:

- Ant Utilities - (Installed)
- Embedded messaging - 71.8 MB
  - Server and Client - 70.0 MB
  - Client Only - 10.0 MB
  - Message-driven Bean Samples - 1.7 MB
- IBM HTTP Server Version 1.3.28 - 0.0 MB
- Web server plug-ins - (Installed)
  - Plug-in for IBM HTTP Server v1.3 - 0.0 MB
  - Plug-in for IBM HTTP Server v2.0 - 0.0 MB

Description:

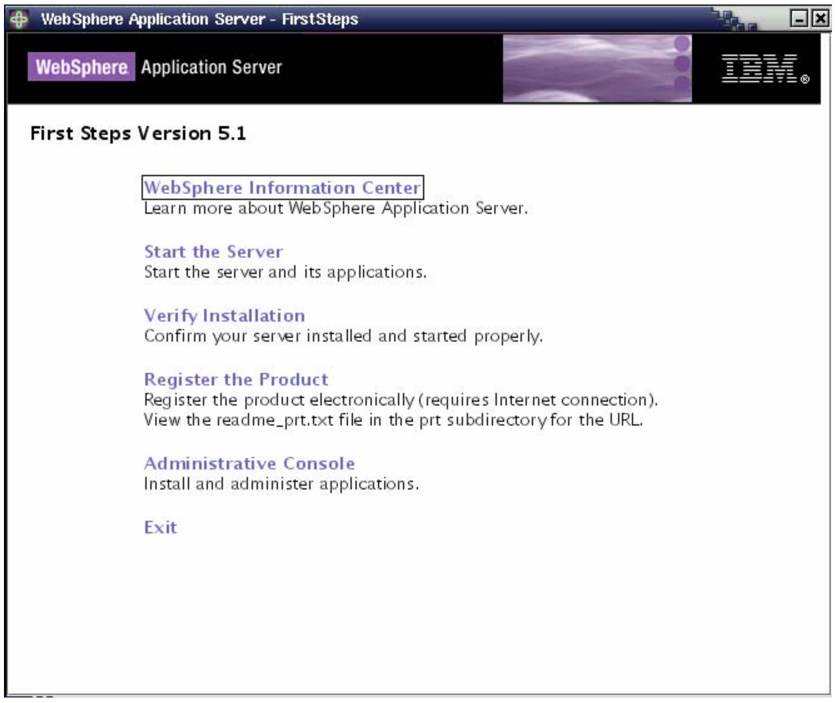
Includes samples for message-driven enterprise beans. You must select Samples under Application Server in order to install these samples.

Starting the Application Server and Verifying Installation

After completing the installation of WebSphere Application Server, start the server and verify the installation. To accomplish this, run the test from the FirstSteps tool that appears immediately upon completion of the installation. The FirstSteps tool provides a place for you to easily access links to commonly performed tasks (depending on which components you chose to install) such as starting and stopping the Administrative Console, running Sample Applications, and running the Verification Application. If this tool does not appear after setup has completed, run the following command:

- On Windows platforms: `install_root\bin\firststeps.bat`
- On Unix platforms: `install_root\bin\firststeps.sh`

The FirstSteps tool looks like this:


WebSphere Application Server - FirstSteps
WebSphere Application Server
IBM

First Steps Version 5.1

WebSphere Information Center
Learn more about WebSphere Application Server.

Start the Server
Start the server and its applications.

Verify Installation
Confirm your server installed and started properly.

Register the Product
Register the product electronically (requires Internet connection).
View the readme_prt.txt file in the prt subdirectory for the URL.

Administrative Console
Install and administer applications.

Exit

Select the option "Verify Installation". This will first start the server, and then run a program that checks that the server is installed and operating normally.

As mentioned previously, you can also use the FirstSteps tool to start and stop the server. Alternatively, issue the following commands:

To start the server:
- On Windows platforms: `install_root\bin\startServer.bat server1`
- On Unix platforms: `install_root\bin\startServer.sh server1`

To stop the server:
- On Windows platforms: `install_root\bin\stopServer.bat server1`
- On Unix platforms: `install_root\bin\stopServer.sh server1`

Set WebSphere Application Server Environment Variable for MQ Install Root

So that WebSphere Application Server can communicate with MQ, ensure that the **MQ_INSTALL_ROOT** environment variable has been set and accurately reflects the filesystem path to the installation directory for MQSeries.

- Expand **Environment** and then select **Manage WebSphere Variables**.
- A list of WebSphere's variables will be displayed in the pane to the right. Scroll down through the list and locate **MQ_INSTALL_ROOT**. Verify that a value has been assigned to this variable and that it is indeed the correct installation root for your MQ configuration.

Preparing for MQ JMS Pub/Sub Implementation

In this section we will ensure that the MQ Message Broker has been started, create the system queues, and run the pub/sub verification test. The following URL provides more detailed information about preparing for MQ JMS Pub/Sub Implementation:

http://www-306.ibm.com/software/integration/mqfamily/library/manuals99/csqzaw/csqzaw0s.htm

Step 1 – Start the MQ Message Broker:

If you haven't already started the MQ Message Broker, start it now by issuing the following command (replace MY.QUEUE.MANAGER with the name of the queue manager you would like to use):

```
strmqbrk –m MY.QUEUE.MANAGER
```

Alternatively, to stop the broker:

```
endmqbrk –m MY.QUEUE.MANAGER
```

To check the status of the broker:

```
dspmqbrk –m MY.QUEUE.MANAGER
```

Step 2 – Create System Queues:

For the JMS Pub/Sub implementation to work correctly, a number of system queues must be created. A script has been supplied in the bin subdirectory of the MQ JMS installation to assist with this task. To use the script, enter the following command:

```
runmqsc MY.QUEUE.MANAGER < MQJMS_PSQ.mqsc
```

Step 3 – Run the Pub/Sub Installation Verification Test:

Ensure the previous step was a success by running the IVT (Installation Verification Test). The MQ JMS Publish/Subscribe Installation Verification Test (MQJMSPSIVT) is supplied in compiled form only, and can be found in the com.ibm.mq.jms package. To run this test, issue the following command (replace HOST.NAME and PORT.NUM with your hostname and port number):

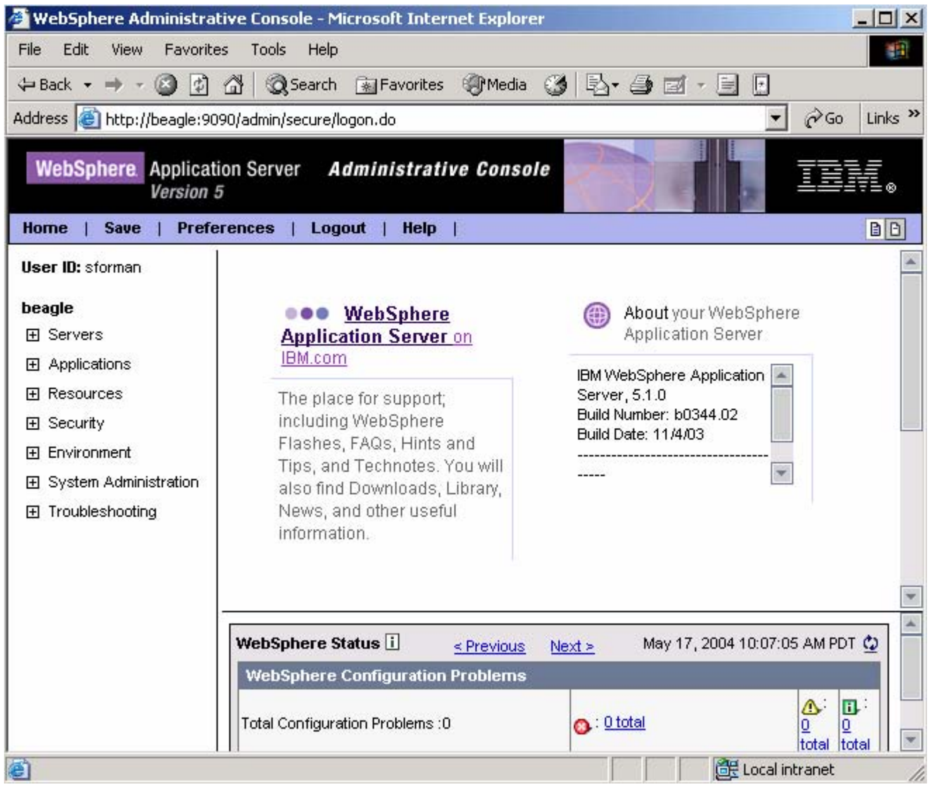
```
com.ibm.mq.jms.MQJMSPSIVT –nojndi –m MY.QUEUE.MANAGER –
client –host HOST.NAME -port PORT.NUM
```

Configure WebSphere MQ Topic Connection Factory

In this section we will configure the Topic Connection Factory settings in the WebSphere Application Server Administrative Console. A Topic connection Factory provides a connection to a JMS Provider of a Topic Destination. We will configure the settings for the corresponding Topic Destination later in the section that follows this one.

Step 1 – Open the WebSphere Application Server Admin Console:

This can be done in a number of ways. You can either select the "Administrative Console" option from the FirstSteps tool, or you can open a browser and browse to the following address: http://HOST_NAME:9090/admin. Where HOST_NAME is the name of the machine where WebSphere Application Server is installed. When the Admin Console first opens, you will be required to provide a user name. This name is used to track the work you perform while logged on to the console and can be any name you choose. Pick a name that easily identifies you as the user. Once logged on, here is what you will see:


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://beagle:9090/admin/secure/logon.do
Go Links
WebSphere Application Server Version 5 Administrative Console
IBM
Home | Save | Preferences | Logout | Help |
User ID: sforman
beagle
Servers
Applications
Resources
Security
Environment
System Administration
Troubleshooting
WebSphere Application Server on IBM.com
About your WebSphere Application Server
WebSphere Status
< Previous
Next >
WebSphere Configuration Problems
Local intranet

The place for support; including WebSphere Flashes, FAQs, Hints and Tips, and Technotes. You will also find Downloads, Library, News, and other useful information.

IBM WebSphere Application Server, 5.1.0
Build Number: b0344.02
Build Date: 11/4/03

May 17, 2004 10:07:05 AM PDT

Total Configuration Problems :0 — 0 total — 0 total — 0 total

Step 2 – Configure the Topic Connection Factory Settings:

- Expand **Resources** and then select **WebSphere MQ JMS Provider**.
- In the pane to the right, under **Additional Properties**, select **WebSphere MQ Topic Connection Factories**.
- Click on the **New** button and enter the following information into the form that appears:

    Name = TopicConnectionFactory

```
JNDI Name = jms/TopicConnectionFactory
Queue Manager = YOUR.QUEUE.MANAGER
```

- Replace YOUR.QUEUE.MANAGER with the name of your queue manager.
- Accept the defaults for all other fields on this form and then click the **OK** button.
- Save your changes to the configuration by clicking on the **Save** link and then the **Save** button.

Configure WebSphere MQ Topic Destination

Follow the steps outlined below to configure the settings for the Topic Destination in the Admin Console:

- Expand **Resources** and then select **WebSphere MQ JMS Provider**.
- In the pane to the right, under **Additional Properties**, select **WebSphere MQ Topic Destinations.**
- Click on the **New** button and enter the following information into the form that appears:

    Name = Topic
    JNDI Name = jms/Topic
    Base Topic Name = Topic

- Accept the defaults for all other fields on this form and then click the **OK** button.
- Save your changes to the configuration.

Deploy the Web Application

The sample application consists of two pieces:

1. A servlet (web application), PublishWeatherServlet, that is displayed in an html page and uses a web form to publish data to the topic.
2. A client GUI, WeatherClient, which subscribes to the topic and receives the data that was published by the web application.

In this section we will install the servlet. Follow the steps below to complete this task:

Step 1 – Install the Web Application

- From the Admin Console, expand **Applications** and select **Install New Application**.
- In the pane to the right, browse to the location where you saved the ear file and then click **Next**.
- Continue to click **Next** until you are able to click **Finish**, then click **Finish.**
- Once the application has been installed, save the changes to your configuration.

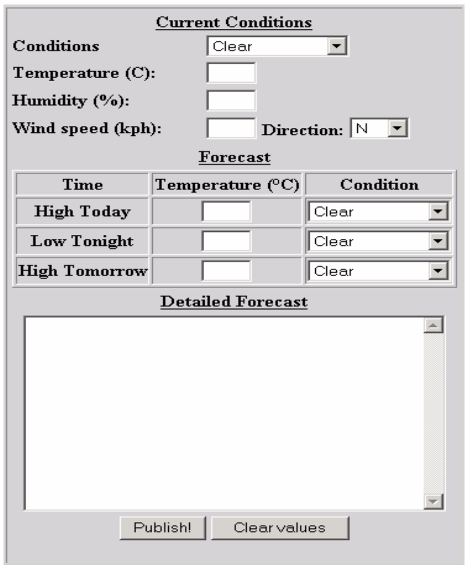
Step 2 – Edit the HTML File

The name of the html file that needs to be changed is **index.html** and can be found in the directory **WAS_HOME/installedApps/HOSTNAME/Apr2003.ear/ttapr2003.war**. The value of the action attribute for the <form> tag needs to be changed to reflect your machine's name as well as the port number assigned to the HTTP transport for communicating requests to the web container in WebSphere Application Server (port 9080). Follow the steps below to accomplish this:

- Browse to **WAS_HOME/installedApps/HOSTNAME/Apr2003.ear/ttapr2003.war** and locate **index.html**. Open this file in a text editor.
- Locate the <form> tag in the document and change the action attribute's value from **"http://localhost:8000/ttapr2003/publishWeather"** to **"http://HOSTNAME:9080/ttapr2003/publishWeather"**.
- Save your changes.
- Stop and restart the application server.

Step 3 – Test the Web Application

You can access the application you just installed by entering the url, **http://HOSTNAME:9080/ttapr2003**, in a web browser. Replace HOSTNAME with the name of your WebSphere Application Server machine. You should see an html form for submitting weather information. This is what it will look like:



Conditions	Clear
Temperature (C):	
Humidity (%):	
Wind speed (kph):	Direction: N

Time	Temperature (°C)	Condition
High Today		Clear
Low Tonight		Clear
High Tomorrow		Clear

Try entering some data and clicking the **Publish!** button. If all is working as it should, you will be redirected to a page in your browser that produces the message "XML Transmitted". In the following section, we will configure and start the client application thereby allowing us to receive data submitted via the web application.

Edit and Run the Client Application

The client application source code can be found in the following directory:

WAS_HOME/installedApps/HOSTNAME/Apr2003.ear/ttapr2003.war/src

Here you will find five java files. PublishWeatherServlet.java is the only file here that does not pertain to the client application. This is the code used in the web application. The other four files will need to be edited according to how you would like to access the Topic Connection Factory and Topic. The following provides a brief example of how you could edit the source; however, keep in mind that you will need to look more closely at the source and your configuration to determine what changes you will need to make:

- Comment out the following block of code in the file SubscriptionHelper.java:

```
/*try
{
        InitialContext ic = new Initialcontext();
        tcf = (TopicConnectionFactory)
        ic.lookup(tcfName);
        topic = (Topic) ic.lookup(topicName);
}
catch(NamingException e)
{
        System.err.println(e.toString());
        e.printStackTrace(System.err);
}*/
```

- Add the following:

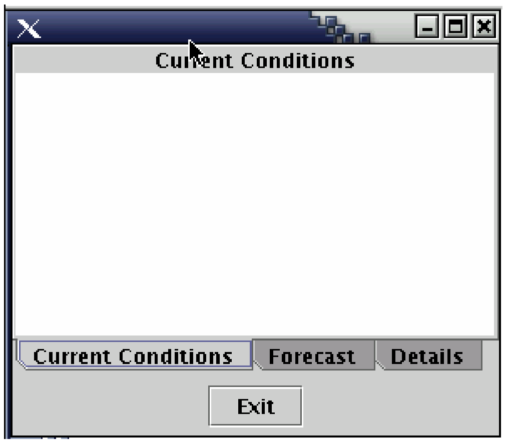
```
try
{
        MQTopicConnectionFactory mqtcf = new MQTopicConnectionFactory();
        mqtcf.setTransportType(JMSC.MQJMS_TP_CLIENT_MQ_TCPIP);
        mqtcf.setHostName("YOUR.HOST.NAME");
        mqtcf.setPort(YOUR.PORT.NUMBER);
        mqtcf.setQueueManager("YOUR.QUEUE.MANAGER");
        tcf = mqtcf;

         topic = new MQTopic("topic://Topic");
}
catch (JMSException jmse)
{
        jmse.printStackTrace();
        System.exit(1);
}
```

- Replace YOUR.HOST.NAME, YOUR.PORT.NUMBER, AND YOUR.QUEUE.MANAGER with the appropriate values for your configuration.
- Re-compile.
- To run the GUI application, type the following from the command line:

```
java WeatherClient
```

A java GUI application will appear. Here is what it will look like:



You can now use the web application to submit (publish) information to the client (subscriber). You can test multiple subscribers to the topic by starting multiple clients. When you submit data from the publisher, all open clients will receive the information that was submitted.

For more details about the application, refer to:
http://java.sun.com/developer/EJTechTips/2003/tt0415.html

Prepare for SSL

In this section we will configure WebSphere Application Server and IBM HTTP Server to enable SSL to protect connections between the application server and the web server plug-in. We will test our configuration with the Publish/Subscribe application installed and configured in the previous sections. Complete the steps that follow to accomplish this task:

Note: The following was taken from the WebSphere Application Server InfoCenter documentation.

Step 1:
First create a self-signed certificate for the web server plug-in. The Web server plug-in requires a key ring file to store its own private and public key files and to store the public certificate from the Web container key file. The following steps are required to generate a self-signed certificate for the Web server plug-in.

- Create a directory on the Web server host for storing the key ring file referenced by the plug-in and associated files, for example:

  /opt/IBMHttpServer/conf/keys

- Launch the key management utility (iKeyman) packaged with the IBM HTTP Server.
- From the iKeyman menu, click **Key Database File > New**.
- Enter the following settings:

  Key database file = CMS Key Database File
  File name = WASplugin.kdb
  Location= /opt/IBMHttpServer/conf/keys (or file of your choice)

- Click **OK**.
- Set the password of your choice at the password prompt. Select the **Stash the Password to a File** check box to save the password to a stash file. This action allows the plug-in to use the password, which provides access to the certificates contained in the key database.
- From the iKeyman menu, click **Create > New Self-Signed Certificate** to create a new self-signed certificate key pair. Specify the following options. Optionally, you can choose to complete all of the remaining fields.

  Key label = WASplugin
  Version = X509 V3
  Key size = 1024
  Common name = droplet.austin.ibm.com (or name of your choice)
  Organization = IBM (replace with your organization's name)
  Country = US
  Validity period = 365

- Click **OK**.
- Extract the public self-signed certificate key: this key is used later by the embedded HTTP server peer to authenticate connections originating from the plug-in. Click **Personal Certificates** in the menu and select the **WASplugin** certificate that you just created.

- Click **Extract Certificate**. Extract the certificate to a file:

  Data type = Base64-encoded ASCII data
  Certificate file name = WASpluginPubCert.arm
  Location = /opt/IBMHttpServer/conf/keys (or directory of your choice)

- Click **OK**.
- Close the key database and exit the iKeyman when you finish.

Step 2:
Generate a self-signed certificate for the Web container.

- Launch the JKS capable iKeyman version located in the product /bin directory (WAS_HOME/bin).
- Click **Key Database File > New** from the iKeyman menu.
- Enter the following settings:

  Key database file = JKS
  File name = WASWebContainer.jks
  Location = /opt/WebSphere/AppServer/etc (or directory of your choice)

- Click **OK**.
- Enter the password of your choice at the password prompt window.
- Click **Create > New Self-Signed Certificate** from the iKeyman menu. The following values were used in this example:

  Key Label = WASWebContainer
  Version = X509 V3
  Key size = 1024
  Common name = droplet.austin.ibm.com (or name of your choice)
  Organization = IBM (replace with your organization's name)
  Country = US
  Validity Period = 365

- Click **OK**.
- Extract the public self-signed certificate key: this key is used later by the Web server plug-in peer to authenticate connections originating from the embedded HTTP server in the product. Click **Personal Certificates** from the list. Select the **WASWebContainer** certificate that you just created. Click **Extract Certificate**. Extract the certificate to a file:

  Data type = Base64-encoded ASCII data
  Certificate file name = WASWebContainerPubCert.arm
  Location = /opt/WebSphere/AppServer/etc

- Click **OK**.
- Close the database and exit the key management utility.

Step 3:
Exchange the public certificates.

- Copy the WASpluginPubCert.arm file from the Web server machine to the WebSphere Application Server machine. The source directory in this case is

/opt/IBMHttpServer/conf/keys, while the destination is **/opt/WebSphere/AppServer/etc**.

- Copy the **WASWebContainerPubCert.arm** file from the product machine to the Web server machine. The source directory in this case is **/opt/WebSphere/AppServer/etc**, while the destination is **/opt/IBMHttpServer/conf/keys**.

Step 4:
Import the certificate into the Web server plug-in key file.

- On the Web server machine, launch the key management utility that supports the CMS key database format.
- From the iKeyman menu, click **Key Database File > Open** and select the previously created key database file: **WASplugin.kdb**.
- In the password prompt window, enter the password. Click **OK**.
- Click **Signer Certificates** from the list and click **Add**. This action imports the public certificate previously extracted from the embedded HTTP server (Web container) keystore file.

  Data type = Base64-encoded ASCII data
  Certificate file name = WASWebContainerPubCert.arm
  Location = /opt/WebSphere/AppServer/etc

- Click **OK**.
- You are prompted for a label name that represents the trusted signer public certificate. Enter a label for the certificate: **WASWebContainer**.
- Close the key database and exit IKeyman when you finish.

Step 5:
Import the certificate into the web container keystore file.

- On the WebSphere Application Server machine, launch the JKS capable iKeyman version, located in the product /bin directory.
- From the iKeyman menu, select **Key Database File > Open**. Select the previously created **WASWebContainer.jks** file.
- In the password prompt window, enter the password. Click **OK**.
- Click **Signer Certificates from** the list. Click **Add**. This action imports the public certificate previously extracted from the embedded HTTP server (Web container) keystore file.

  Data type = Base64-encoded ASCII data
  Certificate file name = WASpluginPubCert.arm
  Location = /opt/WebSphere/AppServer/etc

- Click **OK**.
- You are prompted for a label name that represents the trusted signer public certificate. Enter a label for the certificate: **WASplugin**.
- Close the key database and exit iKeyman when you finish.

Step 6:
Modify the web server plug-in file.

In a production environment, add the secure transport definition, port 9443, to the plugin-cfg.xml file. For example, your modified plugin-key.kdb file contains the following lines:

```
<Transport Hostname="hpws07" Port="9080" Protocol="http"/>
<Transport Hostname="hpws07" Port="9443" Protocol="https"/>
```

After you verify that the proper plugin-key.kdb and plugin-key.sth files exist on the Web server, modify the plugin-cfg.xml file that resides on the Web server. You must specify the local path to both the plugin-key.kdb and plugin-key.sth files in the plugin-cfg.xml file. For more information, see plugin-cfg.xml file and Situations requiring manual editing of the plug-in configuration.

Note: If you manually edit the plugin-cfg.xml file and an automatic regeneration of the file occurs, you must replace your manual edits.

Step 7:
Modify the web container to enable SSL.

To complete the configuration between Web server plug-in and Web container, modify the WebSphere Application Server Web container to use the previously created self-signed certificates.

- Start the WebSphere Application Server administrative console.
- Click **Security > SSL Configuration Repertoires**.
- Click **New** to create a new entry in the repertoire. Provide the following values to complete the form:

  Alias = WebContainerSSLSettings
  Key file name = /opt/WebSphere/AppServer/etc/WASWebContainer.jks
  Key file password = <key_file_password>
  Key file format = JKS
  Trust file name = /opt/WebSphere/AppServer/etc/WASWebContainer.jks
  Trust file password = <trust_file_password>
  Trust file format = JKSClient authentication
  Security level = HIGH

- Click **OK**.
- If you want mutual SSL between the two parties, select the **Client Authentication** check box.
- Save the configuration in the administrative console.
- Click **Servers > Application Servers**, **server_name**, in this example, **server1**.
- Click the Web container located in the server navigation tree.
- Click **HTTP Transport** located in the Web container navigation tree.
- Select the entry for the transfer you want to secure. Click the item under the Host column. Select the asterisk **(*)**, in this case, in the line of port **9443**.
- On the configuration panel, select the **Enable SSL** check box. Click the desired SSL entry from the SSL repertoire list. In this example, the **WebContainerSSLSettings**.
- Click **OK**.

Step 8:
Test the secure connection.

Test the secure connection by accessing a Web application on the WebSphere Application Server using port **9443**. For example, **https://droplet.austin.ibm.com:9443/snoop**.

Step 9:
Import the correct certificate with public and private keys into the browser to test the secured connection, when client-side certification is required.

- Launch the iKeyman utility that supports the CMS key database file, on the Web server machine.
- Open the key file for the plug-in, **/opt/IBMHttpServer/conf/keys/WASplugin.kdb**. Provide the password when prompted.
- Click **WASplugin** certificate, located under the **Personal Certificates**.
- Click **Export**.
- Save the certificate in **PKCS12** format to a file. For example, **/opt/IBMHttpServer/conf/keys/WASplugin.p12**. Provide a password to secure the PKCS12 certificate file.
- Close the key file and exit iKeyman.
- Copy the saved **WASplugin.p12** file to the client machine from where you access the product server.
- Import the PKCS12 file into your browser. Then access **https://droplet.austin.ibm.com:9443/snoop**.
- The browser asks which personal certificate to use for the connection. Select the certificate, and continue connecting.
- Once the browser test with direct product access is successful, test the connection through the Web server using port **9443**, and client certificate, **https://droplet.austin.ibm.com:443/snoop**.

Step 10:
Enable SSL in IBM HTTP Server.

- Open **IHS_HOME/conf/httpd.conf**
- Add the following to the end of the file:

```
LoadModule ibm_ssl_module libexec/mod_ibm_ssl_128.so
Listen 443
SSLEnable
Keyfile /opt/IBMHttpServer/conf/keys/WASplugin.kdb
```

- Stop and restart the http server.

Step 11:
Test the Weather Application with SSL using port 9443.

- In a text editor, open index.html located in the directory:

```
WAS_HOME/installedApps/HOSTNAME/Apr2003.ear/ttapr2003.war
```

- Edit the <form> tag's action attribute to use **"https://"** instead of **"http://"** and port **9443** instead of port **9080**.
- Stop and restart the application server.

- Browse to the following address in a web browser (replace HOSTNAME with your hostname):
  https://HOSTNAME:9443/ttapr2003

You should receive a security alert in the web browser informing you that the security certificate is from an untrusted source. Click **Yes** to proceed. You can now test the web application with the client application just as before.