



WebSphere Application Server for iSeries  
Basic configuration scenarios

*Version 5.0.2*







@server

WebSphere Application Server for iSeries  
Basic configuration scenarios

*Version 5.0.2*

**Note**

Before using this information and the product it supports, be sure to read the information in "Notices," on page 35.

**Fifth Edition (September 2004)**

This edition applies to version 5.0.2 of IBM WebSphere Application Server for iSeries (product number 5733-WS5) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

© Copyright International Business Machines Corporation 1998, 2004. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

## Basic configuration scenarios . . . . . 1

Basic configuration scenario: HitCount. . . . .	1
Overview of the HitCount scenario. . . . .	2
Step 1: Verify the prerequisites . . . . .	2
Step 2: Assemble the application. . . . .	3
Assemble the application: Prepare the HitCount sample application . . . . .	4
Assemble the application: Compile the enterprise bean files . . . . .	5
Assemble the application: Assemble the EJB module . . . . .	5
Assemble the application: Compile the servlet . . . . .	7
Assemble the application: Package the Web application files . . . . .	8
Assemble the application: Assemble the Web module . . . . .	8
Assemble the application: Assemble the application into an Enterprise Archive file . . . . .	10
Step 3: Configure the server . . . . .	12
Configure the server: Create a new instance . . . . .	13
Configure the server: Start the instance . . . . .	13
Configure the server: Create the database tables . . . . .	13
Configure the server: Configure the instance . . . . .	15
Configure the server: Set up security (optional) . . . . .	16
Step 4: Deploy the application . . . . .	17
Step 5: Verify the application . . . . .	18
Basic configuration scenario: Message-driven beans . . . . .	18
Overview of the message-driven beans scenarios . . . . .	19
Step 1: Verify the prerequisites . . . . .	19

Step 2: Assemble the application . . . . .	21
Assemble the application: Prepare the message-driven beans sample application . . . . .	22
Assemble the application: Compile the source code files . . . . .	22
Assemble the application: Package the Web application files . . . . .	23
Assemble the application: Assemble the EJB module. . . . .	23
Assemble the application: Assemble the application client . . . . .	24
Assemble the application: Assemble the application . . . . .	26
Assemble the application: Configure the application client . . . . .	27
Step 3: Configure the server . . . . .	28
Configure the server: Create a new instance . . . . .	29
Configure the server: Start the instance . . . . .	29
Configure the server: Enable your instance to use Java Message Service (JMS). . . . .	29
Configure the server: Set up security (optional) . . . . .	31
Step 4: Deploy the application . . . . .	33
Step 5: Verify the application . . . . .	33

## Appendix. Notices . . . . . 35

Trademarks . . . . .	36
Terms and conditions for downloading and printing publications . . . . .	37



---

## Basic configuration scenarios

If you are new to configuring WebSphere Application Server, start here. This topic shows you how to configure the product and deploy sample applications.

Select one of the following scenarios to follow through the basic configuration process:

**“Basic configuration scenario: HitCount”**

This sample application counts how many times a page is visited. The application includes a Java servlet, a JavaServer Pages (JSP) file, and an enterprise bean. The basic configuration scenario shows you how to assemble the application, create a new instance, set up security, and deploy the application to the instance.

**“Basic configuration scenario: Message-driven beans” on page 18**

This scenario uses the embedded Java Message Service (JMS) support in WebSphere Application Server. The embedded JMS support is installed if you used the default install options when installing WebSphere Application Server. If you chose not to install WebSphere MQ for iSeries 5.3 and WebSphere MQ classes for Java and JMS 5.3 products, you must install the products before attempting this scenario.

---

### Basic configuration scenario: HitCount

This scenario shows you how to configure WebSphere Application Server for the HitCount sample application.

“Overview of the HitCount scenario” on page 2

To configure and deploy the HitCount sample application to a Websphere Application Server instance, follow these steps:

**“Step 1: Verify the prerequisites” on page 2**

Make sure that you meet the necessary prerequisites for the scenario.

**“Step 2: Assemble the application” on page 3**

Package the HitCount sample application for deployment.

**“Step 3: Configure the server” on page 12**

Create and configure an instance of WebSphere Application Server for the scenario.

**“Step 4: Deploy the application” on page 17**

Install the application into your application server.

**“Step 5: Verify the application” on page 18**

Verify that the application was successfully deployed and that it works correctly.



“Step 1: Verify the prerequisites” on page 2

## Overview of the HitCount scenario

The HitCount sample application counts how many times a page is visited. The application includes a Java servlet, a JavaServer Pages (JSP) file, and an enterprise bean.

The HitCount sample application is installed as part of the WebSphere Application Server examples and is already running in the default instance of WebSphere Application Server. It is generally used to verify that your application server was set up correctly. In this scenario you reuse the sample code, compile the appropriate Java files, assemble the class files into a new application, and deploy that application into a new WebSphere Application Server instance.

Assumptions:

- This is a single-server environment. (Your application server has not been federated into a Network Deployment domain.)
- A Windows 32-bit platform (NT or 2000) is used for workstation steps.

### Step 1: Verify the prerequisites

The first step in the basic configuration process is to verify that you have the necessary software installed on your iSeries server. Additionally, it is important that you have the workstation tools installed to continue with the basic configuration process.

1. **Verify installation** (page 2)  
Make sure that WebSphere Application Server is correctly installed.
2. **Install workstation tools** (page 2)  
Make sure you have WebSphere Application Server Application Assembly Tool and iSeries Navigator installed on your workstation.
3. **Map a network drive to your iSeries server** (page 3)  
Make sure you have a network drive mapped to your iSeries server.

#### Verify installation

WebSphere Application Server v5.0 for iSeries should already be installed on your iSeries server before starting the basic configuration process. The following options must be installed for this scenario:

- Base, WebSphere Application Server
- Option 1, Client development and run time
- Option 2, Application server run time
- Option 3, Samples

To verify that WebSphere Application Server v5.0 and the necessary options are already installed on your iSeries server, perform these steps:

1. Enter the Display Software Resources (DSPSFWRSC) command on an OS/400 command line.
2. Look for entries with these product Resource IDs:
  - 5733WS5 \*BASE 5050 WebSphere Application Server V5.0
  - 5733WS5 1 5050 WAS V5.0 Client development and runtime
  - 5733WS5 2 5050 WAS V5.0 Application server runtime
  - 5733WS5 3 5050 WAS V5.0 Samples

If you do not find the product Resource IDs, then these options have not been installed on your iSeries server. See the *Installation* topic for information on how to obtain and install WebSphere Application Server V5.0 for iSeries.

#### Install workstation tools



The workstation components of WebSphere Application Server run on any of the supported WebSphere Application Server workstation platforms. The HitCount sample application requires the installation of the application assembly tool (AAT) to your workstation. For a list of supported workstation platforms and the prerequisites for the workstation tools, see *Workstation prerequisites for installing and running WebSphere Application Server* in the *Installation* topic before installing.

For more information on how to install the application assembly tool, see *Install the workstation tools for WebSphere Application Server* in the *Installation* topic.

You should also have iSeries Navigator installed on your workstation. For more information about how to install iSeries Navigator, see *Installing iSeries Navigator* in the *Connect to iSeries* topic.

### Map a network drive to your iSeries server

The HitCount scenario assumes that you have created and connected to a file share on the root directory of your iSeries server. The scenario requires you to navigate to numerous sub-directories on the iSeries server (including a user-defined directory and the /QIBM directory).

Your iSeries server directories are accessible by mapping a network drive to the integrated file system of your iSeries server. For information on network sharing, see *Create an iSeries NetServer file share* in the *Networking* topic for more information. Perform the following steps to map a network drive to your iSeries server:

1. Right-click **Start**, and click **Explore** to open Windows Explorer on your Windows PC.
2. Select the **Tools -> Map Network Drive...** menu item.
3. Select a letter of a free drive (such as the I:\ drive).
4. In the folder field, enter the name of your iSeries server and file share:

`\\MYISERIES\root`

where *MYISERIES* is the name of your Series server, and *root* is the name of the file share you want to use.

5. Click **Finish**.



“Step 2: Assemble the application”

## Step 2: Assemble the application

Use the application assembly tool to package the HitCount sample application for WebSphere Application Server. Application assembly is the process of creating an enterprise archive (EAR) file that contains all of the files related to an application.

1. “Assemble the application: Prepare the HitCount sample application” on page 4  
Locate the HitCount sample application files, and copy them into your own directory.
2. “Assemble the application: Compile the enterprise bean files” on page 5  
Compile the enterprise bean files into .class files.
3. “Assemble the application: Compile the servlet” on page 7  
Compile the HitCount servlet files into .class files.
4. “Assemble the application: Package the Web application files” on page 8  
The application assembly tool requires a JAR file for input when creating a Web module. Package the HitCount servlet and JSP files into a JAR file.

5. “Assemble the application: Assemble the EJB module” on page 5  
Use the application assembly tool to create an EJB module for use by a Web module.
6. “Assemble the application: Assemble the Web module” on page 8  
Assemble your Web application JAR file and your enterprise bean JAR file into a WAR file.
7. “Assemble the application: Assemble the application into an Enterprise Archive file” on page 10  
Assemble your JAR files and your WAR file into an EAR file, which is used to deploy the application to your WebSphere Application Server instance.



“Assemble the application: Prepare the HitCount sample application”

### Assemble the application: Prepare the HitCount sample application

One of the benefits of WebSphere Application Server is that it does not matter where the source code files for the application are developed. You can assemble the application on any workstation supported by the product, and then deploy the application to an application server installed on any of the supported WebSphere Application Server platforms. This scenario demonstrates how to package an application you have written. The scenario does not describe how to write the application itself. For more information about programming applications for WebSphere Application Server, see the *Application development* topic.

Follow these steps to find the source code files for the HitCount sample application and copy them into a new directory:

1. Enter the Start Qshell (STRQSH) command on an OS/400 command line.
2. To create your sample application directory, enter this command on the Qshell command line:  
`mkdir /hcsample`
3. To locate the existing sample application source files, enter this command:  
`cd /QIBM/UserData/WebAS5/Base/default/installedApps/nodename/DefaultApplication.ear`  
where *nodename* is the host name of the iSeries server.
4. Copy the HitCount sample application source JAR file to the directory that you created. Enter this command:  
`cp Increment_src.jar /hcsample`
5. Copy the HitCount servlet source JAR file to the directory that you created. Enter this command:  
`cp DefaultWebApplication_src.jar /hcsample`
6. On the Qshell command line, enter this command:  
`cd DefaultWebApplication.war`
7. Copy the HitCount JSP source file. On the Qshell command line, enter this command:  
`cp HitCount.jsp /hcsample`
8. On the Qshell command line, enter this command:  
`cd /hcsample`
9. The HitCount sample application source files are packaged in JAR files. To use the source files for compilation and deployment in the basic configuration scenario, you must unpackage the HitCount sample application JAR files. On the Qshell command line, enter these commands:  
`jar xvf Increment_src.jar`  
`jar xvf DefaultWebApplication_src.jar`

**Continue**



“Assemble the application: Compile the enterprise bean files”

### **Assemble the application: Compile the enterprise bean files**

After you copy the source code files for your sample application, you must compile the enterprise bean files into .class files. You must compile the enterprise bean files before you compile the servlet, because the servlet refers to the enterprise bean files.

Follow these steps to compile the code:

1. Enter the Start Qshell (STRQSH) command on an OS/400 command line.
2. Set the classpath to include the J2EE classes, which are contained in the j2ee.jar file. Enter this command:

```
export -s CLASSPATH=./QIBM/ProdData/WebAS5/Base/lib/j2ee.jar
```

3. On the QSH command line, enter this command:

```
cd /hcsample
```

4. Compile the enterprise bean files. Enter this command:

```
javac /hcsample/com/ibm/defaultapplication/*.java
```

### **Troubleshooting**

If your code compiles with errors, check if the correct Java version is being used. On the Qshell command line, enter this command:

```
java -version
```

If it does not indicate the JDK 1.3.x level, you need to explicitly specify the version to use when compiling by entering this command:

```
javac -J-Djava.version=1.3 /hcsample/com/ibm/defaultapplication/*.java
```

If you still have compilation errors after checking your Java version, check your classpath. On a Qshell command line, enter this command:

```
echo $CLASSPATH
```

The correct classpath is ./QIBM/ProdData/WebAS5/Base/lib/j2ee.jar. The first period refers to the current directory, and /QIBM/ProdData/WebAS5/Base/lib/j2ee.jar refers to the J2EE APIs.

**Continue**



“Assemble the application: Compile the servlet” on page 7

### **Assemble the application: Assemble the EJB module**

The application assembly tool is used to package the enterprise bean files into an EJB module. The EJB module is used in later steps to assemble an application and deploy it into a WebSphere Application Server instance. The end result of an EJB module is a JAR file that contains enterprise bean files.

1. Start the application assembly tool. On your workstation, select **Start** —> **Programs** -> **IBM Websphere** -> **Application Server v5.0** -> **Application Assembly Tool**.

2. The **Welcome to the Application Assembly Tool** window is displayed. On the **New** tab, click **EJB Module**, and click **OK**.

The application assembly tool uses a navigation bar to access various parts of the interface. The EJB module file has a default name and directory structure and contains general EJB module information.

3. Set up general EJB module information.
  - On the **General** tab, specify the following:
    - In the **Display name** field, specify **Increment**.
    - Optionally, you can specify a description in the **Description** field.
    - Click **Apply**.
  - On the **CMP Resource Bindings** tab, specify the following:
    - In the **JNDI name** field, specify **jdbc/HitCount**.
    - Click **Apply**.
4. Add the HitCount sample application files to the application assembly tool.
  - In the navigation bar, right-click **Files**, and select **Add Files**.
  - Click **Browse**, and navigate to the **/hcsample** directory.
  - Select **hcsample**, and click **Select**.

**Note:** Do not double-click to open the hcsample directory. You may have to go up one level to be able to select the hcsample folder, because it is created at the root of your server's directory path.
  - Select **com** in the list that appears in the window.
  - Click **Add**.
  - Click **OK**.
5. Add the enterprise bean files.
  - In the navigation bar, right-click **Entity Beans**, and select **New -> CMP**.
  - In the **EJB name** field, specify **Increment**.
  - In the **Display name** field, specify **Incrementejb**.
  - Optionally, you can specify a description in the **Description** field.
  - In the **EJB class** field, click **Browse**.
  - Navigate to the defaultapplication folder (located in the **/com/ibm** directory structure), and select **IncrementBean.class**.
  - Click **OK**.
  - In the **Remote: Home** field, click **Browse**.
  - Navigate to the defaultapplication folder, and select **IncrementHome.class**.
  - Click **OK**.
  - In the **Remote: Interface** field, click **Browse**.
  - Navigate to the defaultapplication folder, and select **Increment.class**.
  - Click **OK**.
  - On the **Advanced** tab, click **Browse** next to the **Primary key** class field.
  - Navigate to the defaultapplication folder, and select **IncrementKey.class**.
  - Click **OK**.
  - In the **Primary key** field, make sure **Compound key** is displayed.
  - In the **Version** field, make sure **2.x** is displayed.
  - In the **Abstract schema name** field, specify **Increment**.
  - Click **OK**.
  - In the navigation bar, expand **Entity Beans**, expand **Incrementejb**, right-click **CMP fields**, and click **New**.
  - Select **Primary Key** from the drop-down list, and click **OK**.

- Right-click **CMP fields** again, and click **New**.
  - Select **TheValue** from the drop-down list, and click **OK**.
6. Save your EJB module.
    - Select the **File -> Save** menu item to save your EJB module in the hcsample folder.
    - Save your EJB module as **Increment.jar**.
    - Click **OK**.
  7. Select the **File -> Close** menu item to close the file.



“Assemble the application: Assemble the Web module” on page 8

### Assemble the application: Compile the servlet

After you copy the source code files for your sample application and compiled your enterprise bean files, you must compile the servlet into a .class file before you can package it into an EAR file and deploy it on your iSeries server.

Follow these steps to compile the servlet:

1. Enter the Start Qshell (STRQSH) command on an OS/400 command line.
2. On the Qshell command line, enter this command:
 

```
cd /hcsample
```
3. Set the classpath to include the J2EE classes, which are contained in the j2ee.jar file. Enter this command:
 

```
export -s CLASSPATH=./QIBM/ProdData/WebAS5/Base/lib/j2ee.jar
```
4. Compile your servlet. Enter this command:
 

```
javac HitCount.java
```

### Troubleshooting

If your code compiles with errors, check if the correct Java version is being used. On the Qshell command line, enter this command:

```
java -version
```

If it does not indicate the JDK 1.3.x level, you need to explicitly specify the version to use when compiling by entering this command:

```
javac -J-Djava.version=1.3 /hcsample/com/ibm/defaultapplication/*.java
```

If you still have compilation errors after checking your Java version, check your classpath. On the Qshell command line, enter this command:

```
echo $CLASSPATH
```

The correct classpath is `./QIBM/ProdData/WebAS5/Base/lib/j2ee.jar`. The first period refers to the current directory, and `/QIBM/ProdData/WebAS5/Base/lib/j2ee.jar` refers to the J2EE APIs.

**Continue**



“Assemble the application: Package the Web application files”

### **Assemble the application: Package the Web application files**

The application assembly tool requires a JAR file for input when creating a Web module. Follow these steps to package the HitCount servlet and JSP files into a JAR file.

1. Enter the Start Qshell (STRQSH) command on an OS/400 command line.
2. On the Qshell command line, enter this command:  
`cd /hcsample`
3. Package the servlet and JSP files into a JAR file named HitCount.jar. Enter this command:  
`jar cf HitCount.jar HitCount.*`

**Continue**



“Assemble the application: Assemble the EJB module” on page 5

### **Assemble the application: Assemble the Web module**

A Web module can contain JAR, HTML, or graphic files that your application uses. The HitCount sample application requires the packaging of the JSP file, the servlet, and an XML descriptor into the Web module. To create a Web module, perform the following steps:

1. Start the application assembly tool. On your workstation, select **Start** —> **Programs** -> **IBM Websphere** -> **Application Server v5.0** -> **Application Assembly Tool**.

**Note:** If you already have the application assembly tool open, select the **File** -> **New** -> **Web Module** menu item.

2. The **Welcome to the Application Assembly Tool** window is displayed. On the **New** tab, click **Web Module**, and click **OK**.

The application assembly tool uses a navigation bar to access various parts of the interface. The Web module file has a default name and directory structure and contains general Web module information.

3. Add the general Web module information.
  - On the **General** tab, specify HitCount in the **Display Name** field.
  - On the **IBM Extensions** tab, check **Precompile JSP** to obtain better runtime performance.
  - Click **Apply**.
4. Add the files needed in the Web module.
  - Expand **Files** in the navigation bar, right-click **Class Files**, and select **Add Files**.
  - In the **Add Files** window, click **Browse** next to the **Root Directory or Archive** field.
  - In the window that appears, select **HitCount.jar**, and click **Select**.
  - In the **Add Files** window, select **HitCount.class** from the list, and click **Add**.
  - Click **OK**.
  - Expand **Files** in the navigation bar, right-click **Resource Files**, and select **Add Files**.
  - In the **Add Files** window, click **Browse** next to the **Root Directory or Archive** field.
  - In the window that appears, select **HitCount.jsp**, and click **Select**.
  - In the **Add Files** window, select **HitCount.jsp** from the list, and click **Add**.

- Click **OK**.
5. Add the Web components.
    - In the navigation bar, right-click **Web Components**, and click **New**.
    - On the **General** tab, specify **HitCount** in the **Component name** field.
    - Specify **HitCount** in the **Display name** field.
    - Optionally, you can specify a description of the WAR file in the **Description** field.
    - On the **General** tab, select **Servlet**, and click **Browse**.
    - Expand the **WEB-INF** folder, and select **classes**.
    - Select **HitCount.class**.
    - Click **OK**.
    - Click **OK** to save your changes.
  6. Add the servlet mapping. The servlet mapping value becomes a part of the URL that is used to access the application once it has been assembled and deployed. For example: `http://your.server.name:port/hc/hitcount`.
    - In the navigation bar, right-click **Servlet Mapping**, and click **New**.
    - Specify `/hitcount` in the **URL pattern** field.
    - Make sure **HitCount** is specified in the **Servlet** field.
    - Click **OK**.
  7. Add the EJB references.
    - In the navigation bar, right-click **EJB References**, and click **New**.
    - On the **General** tab, specify **Increment** in the **Name** field.
    - Optionally, you can specify a description in the **Description** field.
    - Click **Browse** next to the **Home** field.
    - Click **Browse** again to navigate to the file **Increment.jar**.  
**Note:** The file resides in the **hcsample** folder you created on your iSeries server.
    - Select **Increment.jar**, and click **Select**.
    - Navigate to the defaultapplication folder (located in the **/com/ibm/** directory structure), and select **IncrementHome.class**.
    - Click **OK**.
    - Click **Browse** next to the **Remote** field.
    - Click **Browse** again to navigate to **Increment.jar**.  
**Note:** The file resides in the **hcsample** folder you created on your iSeries server.
    - Select **Increment.jar**, and click **Select**.
    - Navigate to the defaultapplication folder, select **Increment.class**, and click **OK**.
    - On the **Bindings** tab, specify **Increment** in the **JNDI name** field.
    - Click **OK**.
    - Click **OK** to return to the main window.
  8. Save your Web module.
    - Select the **File -> Save** menu item to save your Web module in the **hcsample** folder.
    - Save your Web module as **HitCount.war**.
    - Click **OK**.
  9. Select the **File -> Close** menu item to close the file.

**Continue**



“Assemble the application: Assemble the application into an Enterprise Archive file”

### **Assemble the application: Assemble the application into an Enterprise Archive file**

A J2EE application is represented by and packaged in an enterprise archive (EAR) file. You must assemble your EJB module and your Web module into an EAR file, which you can then install into a WebSphere application server instance. Use the application assembly tool to package your EJB and Web modules into an EAR file.

1. Start the application assembly tool. On your workstation, select **Start** → **Programs** -> **IBM Websphere** -> **Application Server v5.0** -> **Application Assembly Tool**.

**Note:** If you already have the application assembly tool open, select the **File** -> **New** -> **Application** menu item.

2. The **Welcome to the Application Assembly Tool** window is displayed. On the **New** tab, click **Application**, and click **OK**.

The application assembly tool uses a navigation bar to access various parts of the interface. The EAR file has a default name and directory structure and contains general EAR file information.

3. Add general EAR file information.

- On the **General** tab, specify HitCount in the **Display name** field.
- On the **Bindings** tab, specify HitCount in the **Enterprise application name** field.
- Click **Apply**.

4. Add the files needed in the EAR file.

- In the navigation bar, right-click **Files**, and click **Add Files**.
- Click **Browse**, and navigate to **Increment.jar**.
- **Note:** The file resides in the hcsample folder you created on your iSeries server.
- Select **Increment.jar**, and click **Select**.
- Click **Add** to add the entire contents of the Increment.jar file to the EAR file.
- Click **OK**.
- Right-click **Files**, and click **Add Files**.
- Click **Browse**, and navigate to **HitCount.war**.

**Note:** The file resides in the hcsample folder you created on your iSeries server.

- Select **HitCount.war**, and click **Select**.
- Click **Add** to add the entire contents of the HitCount.war file to the EAR file.
- Click **OK**.

5. Import the EJB module you created into the EAR file.

- In the navigation bar, right-click **EJB Modules**, and click **Import**.
- Navigate to **Increment.jar**, and click **Open**.
- Click **OK** in the **Confirm Values** window that appears.
- Expand **EJB Modules**, expand **Increment**, and select **Entity Beans**.
- On the **CMP Resource Bindings** tab, make sure that **Per Resource** is selected in the **Resource authentication** field.

6. Import the Web module you created into the EAR file.

- In the navigation bar, right-click **Web Modules**, and click **Import**.
- Select **HitCount.war**, and click **Open**.



- On the **Confirm Values** window, type /hc in the **Context root** field. The context root value becomes part of the URL that is used to access the application once it has been assembled and deployed. For example: http://your.server.name:port/hc/hitcount.
  - Click **OK**.
7. Save your EAR file.
    - Select the **File -> Save** menu item to save your EAR file in the hcsample folder.
    - Save your EAR file as **HitCount.ear**.
    - Click **OK**.
  8. Optionally, you can configure security on your EJB module so that all authenticated users are authorized to the enterprise bean's methods.
    - In the navigation bar, right-click **Security Roles**, and click **New**.
    - On the **General** tab, specify **ALL Role** in the **Name** field.
    - Optionally, you can specify a description in the **Description** field.
    - On the **Bindings** tab, click **Add** next to the **Special subjects: Name** field.
    - Select **AllAuthenticatedUsers**, and click **OK**.
    - On the **Bindings** tab, click **OK** to save your changes.
    - In the navigation bar, expand **Entity Beans**, expand **Incrementejb**, right-click **Security Roles**, and click **New**.
    - On the **General** tab, specify **All Role** in the **Name** field.
    - Optionally, you can type a description in the **Description** field.
    - Click **OK**.
    - In the navigation bar, right-click **Method Permissions**, and click **New**.
    - Optionally, you can type a name in the **Method permission name** field and a description in the **Description** field.
    - Next to the **Methods** field, click **Add**.
    - Expand **Increment.jar**, and select **Increment(\*)**.
    - Click **OK**.
    - Next to the **Role** field, click **Add**.
    - Select **All Role**, and click **OK**.
    - On the **General** tab, click **OK** to save your changes.
    - Click **Save** to save the HitCount.ear file with the updated security information.
  9. Optionally, you can configure security on your Web module so that all authenticated users are authorized to the HitCount servlet's HTTP methods (GET and POST).
    - In the navigation bar, select **Web module**
    - On the **Advanced** tab, check **Login configuration**.
    - Select **Basic** for the **Authentication method**.
    - Click **Apply**.
    - In the navigation bar, expand **Web module**, expand **HitCount**, right-click **Security Roles**, and click **New**.
    - On the **General** tab, specify **All Role** in the **Name** field.
    - Optionally, you can specify a description in the **Description** field.
    - Click **OK**.
    - In the navigation bar, expand **Web module**, expand **HitCount**, right-click **Security Constraints**, and click **New**.
    - Click **Add**, and select **All Role**.
    - Click **OK**.

- Optionally, you can specify a name and description in **Security constraint name** and **Description** fields on the **General** tab.
  - Click **OK**.
  - In the navigation bar, expand **Security Constraints**, and expand the security constraint you created in the previous step.
  - Right-click **Web Resource Collection**, and select **New**.
  - In the **Web Resource Name** field, specify HitCount Servlet.
  - Next to the **HTTP methods:HTTP method** field, click **Add**.
  - Select **GET**, and click **Apply**.
  - Select **POST**, and click **OK**.
  - On the **General** tab, click **OK** to save your changes.
10. Generate code for deployment to a WebSphere Application Server instance.
- Select the **File -> Generate code for deployment...** menu item.
  - Take all of the default values except for the **Database type** field.
  - In the **Database type** field, select **DB2UDBISERIES** from the drop-down list.
  - Click **Generate now**.
- The tool generates a file called Deployed\_HitCount.ear, which you use in the next steps.
11. Once the deployed code is generated, click **Close**.



“Step 3: Configure the server”

### Step 3: Configure the server

You can create multiple WebSphere Application Server instances (each containing a single application server) on a single iSeries server. You can use multiple instances to create separate environments for application development and application testing. It is recommended that you create a new Websphere Application Server instance to configure the HitCount application sample in order to preserve the Default Server with its sample configuration.

1. “Configure the server: Create a new instance” on page 13  
Create a new WebSphere Application Server instance to run the sample application.
2. “Configure the server: Start the instance” on page 13  
After you create the instance, you must start the instance.
3. “Configure the server: Create the database tables” on page 13  
Create the underlying database table required by the HitCount sample application.
4. “Configure the server: Configure the instance” on page 15  
The instance must be configured with a JDBC provider and data source before the application can be deployed to the instance.
5. “Configure the server: Set up security (optional)” on page 16  
The WebSphere Application Server security system controls access to resources and the administrative console. Perform this step to enable security.

**Continue**



“Configure the server: Create a new instance”

### **Configure the server: Create a new instance**

To create a new instance, run the `crtwasinst` script from Qshell the command line. This script creates all new server directories and sets up the correct authorities. To run this script, your iSeries user profile must have \*ALLOBJ authority.

1. Enter the Start Qshell (STRQSH) command on an OS/400 command line.
2. On the Qshell command line, use the `cd` command to change to the directory that contains the script. For example:

```
cd /QIBM/ProdData/WebAS5/Base/bin
```

3. Run the `crtwasinst` script. Enter this command:

```
crtwasinst -instance basicconfig -portblock portblock
```

where *basicconfig* is the name of the instance that is created and *portblock* is the first of a block of 13 consecutive ports. For more information on the `crtwasinst` script and additional parameters, see *The crtwasinst script*.

**Continue**



“Configure the server: Start the instance”

### **Configure the server: Start the instance**

To start an instance, run the `startServer` script from the Qshell command line. To run the script, follow these steps:

1. Enter the Start Qshell (STRQSH) command on an OS/400 command line.
2. On the Qshell command line, use the `cd` command to change to the directory that contains the script. For example:

```
cd /QIBM/ProdData/WebAS5/Base/bin
```

3. Run the `startServer` script. Enter this command:

```
startServer -instance basicconfig
```

where *basicconfig* is the name of the instance you want to start. For more information on the `startServer` script and additional parameters, see *The startServer script*.

**Continue**



“Configure the server: Create the database tables”

### **Configure the server: Create the database tables**

You must create the underlying database tables required by the HitCount sample application. A Data Definition Language (DDL) script is created for each entity enterprise bean when the bean is packaged into an EJB module using the application assembly tool. This DDL script is named `Table.ddl`, and it is

stored in the enterprise bean's JAR file after it has been packaged in the EJB module. This DDL script contains the required commands for creating the underlying database table for the entity bean.

To use this generated script file to create the database table on your iSeries server, follow these steps:

1. Extract the Table.ddl file from your EJB Module.
  - a. Use a file compression utility to open your Deployed\_HitCount.ear file.
  - b. Select **META-INF/Table.ddl**.
  - c. Extract the file to the /hcsample directory.
2. Start iSeries Navigator.
3. Expand the iSeries icon for the system where you want to create the database file.
4. Expand **Database**, and right-click the system database.
5. Select **Run SQL Scripts...**
6. Select the **File -> Open** menu item.
7. Change the **Files of type** view to All Files (\*.\*\*).
8. Navigate to the Table.ddl file that was extracted to the META-INF folder in a previous step and select **Open**.
9. In the file, you must create a database (or collection) used for the HitCount sample application. Specify the following SQL statement as the first statement in the file:  
CREATE COLLECTION HITCOUNTDB;
10. You must also qualify the tables and constraints in the SQL statements with the database (or collection) name. Edit the SQL statements to contain the following values:

```
CREATE TABLE HITCOUNTDB.INCREMENT
ALTER TABLE HITCOUNTDB.INCREMENT
ADD CONSTRAINT HITCOUNTDB.PK_INCREMENT PRIMARY KEY (PRIMARYKEY);
```

11. Select the **Run -> All** menu item to execute all commands contained within the script.
12. Select the **View -> Job Log...** menu item and verify that the table was created successfully.
13. Select the **File -> Save** menu item to save the file.
14. Exit the Run SQL Scripts application.
15. Exit iSeries Navigator.

### Create the proper authorities

For the HitCount sample application and database tables, it is easiest to grant all users authority to the database tables. Alternatively, you could grant authority to just QEJBSVR. Run the following commands on an iSeries server command line to change authority to \*ALL for all users:

- GRTOBJAUT OBJ(HITCOUNTDB) OBJTYPE(\*LIB) USER(\*PUBLIC) AUT(\*ALL)
- GRTOBJAUT OBJ(HITCOUNTDB/\*ALL) OBJTYPE(\*ALL) USER(\*PUBLIC) AUT(\*ALL)

Because all users of the database tables now have authority, there is no need to specify a user ID and password in the data source. The data source accesses the tables using the QEJBSVR user profile, which it is running in.



“Configure the server: Configure the instance” on page 15

## Configure the server: Configure the instance

The first thing you must do to configure the instance to access your samples database is create a JDBC provider and data source. The WebSphere Application Server administrative console is used for this step. To start the console, open this URL in your browser:

`http://your.server.name:port/admin`

where *your.server.name* is the hostname of your iSeries server and *port* is your administrative port number. The administrative port number was assigned when you created your instance. For more information about how to display your port numbers, see the *The dspwasinst script* topic. For more information about the WebSphere Application Server administrative console, see the *The administrative console* topic.

### Create a JDBC provider

JDBC providers are used by the installed applications to access data from databases. Follow these steps to create a JDBC provider from the administrative console:

1. Start the administrative console.
2. In the topology tree, expand **Resources** and click **JDBC Providers**.
3. On the **JDBC Providers** page, specify **Server** as the scope of the resource. Click **Apply**.
4. Click **New**.
5. In the **JDBC Providers** field, select the appropriate Native driver for your OS/400 release level.  
**Note:** Do not select the Native-XA driver. An XA driver is not needed by the HitCount application and may introduce some performance overhead.
6. Click **Apply**.
7. Optionally, you can specify a description, classpath, and native library path.
8. Click **Apply** or **OK**.
9. Click **Save** on the toolbar to save the configuration.
10. Click **Save** again to update the master repository with your changes.

### Configure a data source

After you create a JDBC provider, you need to configure a data source for the provider. Data source is used by the application to access the data from the database. It is created under a JDBC provider, which provides the specific JDBC driver implementation class. To configure a data source, follow these steps:

1. On the **JDBC Providers** page of the administrative console, click the name of the JDBC provider for which you want to configure a data source.
2. On the next page, click **Data Sources**.
3. On the **Data Sources** page, click **New**.
4. Specify `jdbc/HitCount` in the **JNDI name** field. You can also specify several optional properties on this page.
5. Check **Container managed persistence** to use this Data Source in container managed persistence (CMP).
6. Click **Apply**.
7. Click **Custom properties**.
8. Click **databaseName**.
9. In the Value field, specify `*LOCAL/HITCOUNTDB`.
10. Click **Apply**.
11. Save the administrative configuration.
12. Restart your application server.
  - Enter the Start Qshell (STRQSH) command on an OS/400 command line.

- On the Qshell command line, enter this command:  
cd /QIBM/ProdData/WebAS5/Base/bin
- On the Qshell command line, enter this command:  
stopServer -instance basicconfig
- Once the instance has ended, start it again by entering this command:  
startServer -instance basicconfig



“Configure the server: Set up security (optional)”

### Configure the server: Set up security (optional)

The WebSphere Application Server security server controls access to application server resources (servlets, JSPs, HTML files and EJBs). It also controls access to the administrative console and to some administrative tools. The security system is not responsible for protecting native OS/400 resources (such as database files and program objects).

When you install WebSphere Application Server for iSeries, the product creates a default application server instance. This instance is named default. Security is initially disabled in the default application server instance and any other application server instance you create. Unless you believe your iSeries server is adequately protected by a firewall or some other mechanism, immediately enable security after creating an instance.

This step enables local operating system security, which uses an OS/400 user profile and password to secure the application server. For more information on WebSphere Application Server security, including security configuration options other than those for the local operating system, see the *WebSphere Application Server security* topic.

Perform the following steps to enable security in your instance:

1. Start the administrative console.

For more information, see *Start the administrative console*.

2. Expand **Security**.
3. Expand **User Registries**.
4. Click **Local OS**.
5. In the **Server User ID** field, specify a valid OS/400 user profile name.

**Note:** A user profile that is a part of a group profile cannot be used to configure the LocalOS user registry. A group profile is assigned a unique group ID number, which is not assigned to a regular user profile. Run the Display User Profile (DSPUSRPRF) command to determine if the user profile you want to use as the Server User ID has a defined group ID number. If the Group ID field is set to \*NONE, the user profile can be used as the administrative user ID.

6. In the **Server User Password** field, specify the password for the user profile you specified.
7. Click **OK**.
8. In the topology tree, click **Global Security**.
9. On the **General Properties** page, check **Enabled**. Accept the remaining default property values.
10. Click **OK**.
11. Click **Save** on the toolbar to save the configuration.
12. Click **Save** again to update the master repository with your changes.
13. Click **Logout** on the toolbar.

14. Restart your application server.
  - Enter the Start Qshell (STRQSH) command on an OS/400 command line.
  - On the Qshell command line, enter this command:  
`cd /QIBM/ProdData/WebAS5/Base/bin`
  - On the Qshell command line, enter this command:  
`stopServer -instance basicconfig`
  - Once the instance has ended, start it again by entering this command:  
`startServer -instance basicconfig`

The next time you start the administrative console, use the iSeries user profile that you specified when you enabled security to log into the console.



“Step 4: Deploy the application”

## Step 4: Deploy the application

Once you have created the EAR file and configured the JDBC driver and data source for the application, the next step is to install the application into your WebSphere Application Server instance. The WebSphere Application Server administrative console is used for this step. To start the console, open this URL in your browser:

`http://your.server.name:port/admin`

where *your.server.name* is the hostname of your iSeries server and *port* is your administrative port number. The administrative port number was assigned when you created your instance. For more information about how to display your port numbers, see the *The dspwasinst script* topic. For more information about the WebSphere Application Server administrative console, see the *The administrative console* topic.

To deploy the EAR file into your application server, use the Application Installation Wizard. To use the wizard, follow these steps:

1. In the topology tree, expand **Applications**, and click **Install new application**
2. In the **Local path** field, click **Browse**.
3. Navigate to **Deployed\_HitCount.ear**, and click **Open**.
4. Click **Next**.
5. On the **Preparing for application install** page, accept all defaults, and click **Next**.
6. On the **Install New Application Step 1** page, accept all defaults, and click **Next**.
7. On the **Install New Application Step 2** page, accept all defaults, and click **Next**.
8. On the **Install New Application Step 3** page, accept all defaults, and click **Next**.
9. On the **Install New Application Step 4** page, specify `jdbc/HitCount` in the **JNDI name** field, and click **Next**.
10. On the **Install New Application Step 5** page, accept all defaults, and click **Next**.
11. On the **Install New Application Step 6** page, accept all defaults, and click **Next**.
12. On the **Install New Application Step 7** page, accept all defaults, and click **Next**.
13. On the **Install New Application Step 8** page, accept all defaults, and click **Next**.
14. On the **Install New Application Step 9** page, review the summary, and click **Finish**.

15. Once you receive a message that the EAR file installed correctly, click **Save** in the toolbar. Click **Save** again to update the master repository with your changes.
16. In the topology tree, expand **Applications**, and click **Enterprise Applications**.
17. Check **HitCount**, and click **Start**.

You will receive a message stating that the application started successfully.

18. Restart your application server.
  - Enter the Start Qshell (STRQSH) command on an OS/400 command line.
  - On the Qshell command line, enter this command:  
`cd /QIBM/ProdData/WebAS5/Base/bin`
  - On the Qshell command line, enter this command:  
`stopServer -instance basicconfig`
  - Once the instance has ended, start it again by entering this command:  
`startServer -instance basicconfig`



“Step 5: Verify the application”

## Step 5: Verify the application

Open your browser to the following URL:

`http://your.server.name:port/hc/hitcount`

where *your.server.name* is the name of your iSeries server and *port* is your internal HTTP port number. The internal HTTP port number was assigned when you created your instance. For more information about how to display your port numbers, see the *The dspwasinst script* topic.

If security is enabled, you will be prompted for a valid User ID and password.

The HitCount application is a simple demonstration of a variety of methods to increment a counter value. Select a method of execution, a namespace lookup method for the enterprise bean files, and a transaction type to obtain an incremented counter value.

If the page loads with the HitCount application, the deployment was successful.

---

## Basic configuration scenario: Message-driven beans

This scenario shows you how to configure WebSphere Application Server for the message-driven beans sample application. The message-driven beans sample application is a part of the WebSphere Application Server samples gallery, which provides a set of small, generic samples that show how to perform common enterprise application tasks. For more information about the samples gallery, see *WebSphere Application Server samples gallery* in the *Installation* topic.

“Overview of the message-driven beans scenarios” on page 19

To configure and deploy the message-driven beans sample application to a WebSphere Application Server instance, follow these steps:

### “Step 1: Verify the prerequisites” on page 19

Make sure that you meet the necessary prerequisites for the scenario.



**“Step 2: Assemble the application” on page 21**

Package the sample application for deployment.

**“Step 3: Configure the server” on page 28**

Set up JMS resources needed for the message-driven beans sample application.

**“Step 4: Deploy the application” on page 33**

Install the application into your application server.

**“Step 5: Verify the application” on page 33**

Verify that the application was successfully deployed and that it works correctly.



“Step 1: Verify the prerequisites”

## Overview of the message-driven beans scenarios

The WebSphere Application Server samples gallery provides a set of small, generic samples that show how to perform common enterprise application tasks. These samples demonstrate the use of session and entity enterprise beans, JDBC access, connection pooling, Java Mail, message driven beans, and other Web techniques and reusable componentry. For more information about the samples gallery, see *WebSphere Application Server samples gallery* in the *Installation* topic.

The message-driven beans sample applications demonstrate how to:

- Assemble and deploy a message-driven bean.
- Assemble and run a Java Message Service (JMS) publish and subscribe application client.
- Configure JMS resources within WebSphere Application Server.
- Configure JMS listener ports so a message-driven bean receives messages.

For the basic configuration scenario, the Publish and Subscribe message-driven bean sample is used. The Publish and Subscribe message-driven bean sample consists of two parts: The message-driven bean and the application client. The application client publishes a message on a topic and exits. The message-driven bean receives the message and prints the contents of the message to standard output of the application server.

Assumptions:

- This is a single-server environment. (Your application server has not been federated into a Network Deployment domain.)
- A Windows 32-bit platform (NT or 2000) is used for workstation steps.

## Step 1: Verify the prerequisites

The first step in the basic configuration process is to verify that you have the necessary software installed on your iSeries server. Because this basic configuration scenario requires the use of the samples gallery, it is important that you make sure your version of WebSphere Application Server V5.0 has Option 3 installed. The embedded Java Message Service (JMS) support must also be installed in order for the basic configuration process to run JMS. The embedded JMS support is installed if you used the default install options when installing WebSphere Application Server V5.0. If you chose not to install the WebSphere MQ for iSeries 5.3 and WebSphere MQ classes for Java and JMS 5.3 products, you must install the products before attempting this scenario. See the *Installation* topic for more information.

Additionally, it is required that you install the workstation tools on your workstation to configure the message-driven bean client files.

1. **Verify installation** (page 20)  
Make sure that WebSphere Application Server V5.0 and the product options for Samples and JMS are correctly installed.
2. **Install workstation tools** (page 20)  
Make sure you have the WebSphere Application Server Application Assembly Tool and the WebSphere Application Server Application Client Resource Configuration Tool installed on your workstation.
3. **Map a network drive to your iSeries server** (page 20)  
Make sure you have a network drive mapped to your iSeries server.

### Verify installation

WebSphere Application Server V5.0 for iSeries should already be installed on your iSeries server before starting the basic configuration process. The following options must be installed for this scenario:

- \*BASE option, WebSphere Application Server
- Option 1, Client development and run time
- Option 2, Application server run time
- Option 3, Samples
- \*BASE option, WebSphere MQ V5.3 for iSeries
- \*BASE option, WebSphere MQ classes for Java and JMS V5.3 for iSeries

To verify that WebSphere Application Server V5.0 and the necessary options are already installed on your iSeries server, perform these steps:

1. Enter the Display Software Resources (DSPSFWRSC) command on an OS/400 command line.
2. Look for entries with these product Resource IDs:
  - 5733WS5 \*BASE 5050 WebSphere Application Server V5.0
  - 5733WS5 1 5050 WAS V5.0 Client development and runtime
  - 5733WS5 2 5050 WAS V5.0 Application server runtime
  - 5733WS5 3 5050 WAS V5.0 Samples
  - 5724B41 WebSphere MQ V5.3 for iSeries
  - 5639C34 MQSeries classes for Java and JMS

If you do not find the product Resource IDs, then these options have not been installed on your iSeries server. See the *Installation* topic for information on how to obtain and install WebSphere Application Server V5.0 for iSeries.

### Install workstation tools

The workstation components of WebSphere Application Server run on any of the supported WebSphere Application Server workstation platforms. The message-driven beans sample application requires the installation of the application assembly tool (AAT) and WebSphere Application Client Resource Configuration Tool to your workstation. For a list of supported workstation platforms and the prerequisites for the workstation tools, see *Workstation prerequisites for installing and running WebSphere Application Server* in the *Installation* topic before installing.

For more information on how to install the tools required for the message-driven beans basic configuration scenario, see *Install the workstation tools for WebSphere Application Server* in the *Installation* topic.

### Map a network drive to your iSeries server

The message-driven beans scenario assumes that you have created and connected to a file share on the root directory of your iSeries server. The scenario requires you to navigate to numerous sub-directories on the iSeries server (including a user-defined directory and the /QIBM directory).

Your iSeries server directories are accessible by mapping a network drive to the integrated file system of your iSeries server. For information on network sharing, see *Create an iSeries NetServer file share* in the *Networking* topic for more information. Perform the following steps to map a network drive to your iSeries server:

1. Right-click **Start**, and click **Explore** to open Windows Explorer on your Windows PC.
2. Select the **Tools -> Map Network Drive...** menu item.
3. Select a letter of a free drive (such as the I:\ drive).
4. In the folder field, enter the name of your iSeries server and file share:  
`\\MYISERIES\root`  
where *MYISERIES* is the name of your iSeries server, and *root* is the name of the file share you want to use.
5. Click **Finish**.



“Step 2: Assemble the application”

## Step 2: Assemble the application

Use the application assembly tool to package the sample application for WebSphere Application Server. Application assembly is the process of creating an enterprise archive (EAR) file that contains all of the files related to an application.

1. “Assemble the application: Prepare the message-driven beans sample application” on page 22  
Locate the message-driven beans sample application files and copy them into your own directory.
2. “Assemble the application: Compile the source code files” on page 22  
Compile the source code files into .class files.
3. “Assemble the application: Package the Web application files” on page 23  
Package the servlet and .class files into a JAR file. The application assembly tool requires a JAR file for input when creating a application client.
4. “Assemble the application: Assemble the EJB module” on page 23  
Use the application assembly tool to create an EJB module.
5. “Assemble the application: Assemble the application client” on page 24  
Use the application assembly tool to create an application client.
6. “Assemble the application: Assemble the application” on page 26  
Assemble your JAR files into an EAR file, which is used to deploy the application to your WebSphere Application Server instance.
7. “Assemble the application: Configure the application client” on page 27  
Use the Application Client Resource Configuration Tool to configure the application client and add the WebSphere TopicConnectionFactory needed to run your application.



“Assemble the application: Prepare the message-driven beans sample application”

## Assemble the application: Prepare the message-driven beans sample application

One of the benefits of WebSphere Application Server is that it does not matter where the source code files for the application are developed. You can assemble the application on any workstation supported by the product, and then deploy the application to an application server installed on any of the supported WebSphere Application Server platforms. This scenario demonstrates how to package an application you have written. The scenario does not describe how to write the application itself. For more information about programming applications for WebSphere Application Server, see the *Application development* topic.

The sample application source files are located in the /QIBM/ProdData/WebAS5/Base/samples/src/MessageDrivenBeans/com directory. Follow these steps to obtain the source code files for the message-driven beans sample application and copy them into a new directory:

1. Enter the Start Qshell (STRQSH) command on an OS/400 command line.
2. To create your sample application directory, enter this command on the Qshell command line:  

```
mkdir /mdbsamp
```
3. Copy the message-driven beans sample application files and the corresponding directory structure to the directory you created. Enter this command:  

```
cp -R /QIBM/ProdData/WebAS5/Base/samples/src/MessageDrivenBeans/com /mdbsamp
```
4. Change directory to your sample application directory. On the Qshell command line, enter this command:  

```
cd /mdbsamp
```
5. The message-driven beans sample application has two parts: A Point-to-Point sample application and a Publish and Subscribe sample application. For this scenario, only Publish and Subscribe is configured; therefore, it is necessary to delete the files and folders associated with the Point-to-Point sample application. Failure to do so will cause errors in assembly. On the Qshell command line, enter this command:  

```
rm -R com/ibm/websphere/samples/messaging/ptop
```



“Assemble the application: Compile the source code files”

## Assemble the application: Compile the source code files

After you copy the source code files for your sample application, you must compile them into .class files.

Follow these steps to compile the code:

1. Enter the Start Qshell (STRQSH) command on an OS/400 command line.
2. Change directory to your sample application directory. On the QSH command line, enter this command:  

```
cd /mdbsamp
```
3. Set the classpath to include the J2EE classes, which are contained in the j2ee.jar file. Enter this command:  

```
export -s CLASSPATH=./QIBM/ProdData/WebAS5/Base/lib/j2ee.jar
```
4. Compile the source code files. Enter this command:  

```
javac com/ibm/websphere/samples/messaging/pubsub/*.java
```

## Troubleshooting

If your code compiles with errors, check if the correct Java version is being used. On the Qshell command line, enter this command:

```
java -version
```

If it does not indicate the JDK 1.3.x level, you need to explicitly specify the version to use when compiling by entering the following command:

```
javac -J-Djava.version=1.3 com/ibm/websphere/samples/messaging/pubsub/*.java
```

If you still have compilation errors after checking your Java version, check your classpath. On the Qshell command line, enter this command:

```
echo $CLASSPATH
```

The correct classpath is `./QIBM/ProdData/WebAS5/Base/lib/j2ee.jar`. The first period refers to the current directory, and `/QIBM/ProdData/WebAS5/Base/lib/j2ee.jar` refers to the J2EE APIs.



“Assemble the application: Package the Web application files”

### Assemble the application: Package the Web application files

The application assembly tool requires a JAR file for input when creating a Web module. Follow these steps to package the message-driven beans servlet and .class files into a JAR file:

1. Enter the Start Qshell (STRQSH) command on an OS/400 command line.
2. Change directory to your sample application directory. On the Qshell command line, enter this command:  

```
cd /mdbsamp
```
3. Package the enterprise bean file and its associated .class file into a JAR file named `mdb.jar`. Enter this command:  

```
jar cf mdb.jar com/ibm/websphere/samples/messaging/pubsub/JMSpsSampleMDBBean.*
```
4. Package the application client file and its associated .class files into a JAR file named `client.jar`. Enter this command:  

```
jar cf client.jar com/ibm/websphere/samples/messaging/pubsub/JMSpsSampleClient.*
```



“Assemble the application: Assemble the EJB module”

### Assemble the application: Assemble the EJB module

The application assembly tool is used to package the enterprise bean files into an EJB module. The EJB module is used in later steps to assemble an application and deploy it into a WebSphere Application Server instance. The end result of an EJB module is a JAR file that contains enterprise bean files.

1. Start the application assembly tool. On your workstation, select **Start** → **Programs** -> **IBM Websphere** -> **Application Server v5.0** -> **Application Assembly Tool**.
2. The **Welcome to the Application Assembly Tool** window is displayed. On the **New** tab, click **EJB Module**, and click **OK**.

The application assembly tool uses a navigation bar to access various parts of the interface. The EJB module file has a default name and directory structure and contains general EJB module information.

3. Set up general EJB module information.
  - On the **General** tab, specify the following:
    - In the **Display name** field, specify `PSSampleMDB`.
    - Optionally, you can specify a description in the **Description** field.
    - Click **Apply**.
4. Add the message-driven beans sample application files to the application assembly tool.
  - In the navigation bar, right-click **Files**, and select **Add Files**.
  - Click **Browse**, and navigate to the `mdb.jar` file.
  - Select `mdb.jar`, and click **Select**.
  - Select the `com` folder in the list that appears in the window.
  - Click **Add**.
  - Click **OK**.
5. Add the enterprise bean file.
  - In the navigation bar, right-click **Message Driven Beans**, and select **New**.
  - In the **EJB name** field, specify `PSSampleMDB`.
  - In the **Display name** field, specify `PSSampleMDB`.
  - Optionally, you can specify a description in the **Description** field.
  - Click **Browse** next to the **EJB class** field.
  - Click **Browse** next to the **Root Directory or Archive** field.
  - Locate the `mdb.jar` file in the `mdbsamp` folder, and click **Select**.
  - Expand the `com/ibm/websphere/samples/messaging/pubsub` directory.
  - Select `JMSpsSampleMDBBean.class`.
  - Click **OK**.
  - On the **General** tab, select **Container** from the **Transaction type** drop-down list.
  - On the **Advanced** tab, select **Topic** from the **Type** drop-down list.
  - On the **Advanced** tab, select **Durable** from the **Subscription durability** drop-down list.
  - On the **Bindings** tab, specify `SamplePubSubListenerPort` in the **Listener Port Name** field.
  - Click **OK**.
6. Save your EJB module.
  - Select the **File -> Save** menu item to save your EJB module in the `mdbsamp` folder.
  - Save your EJB module as `PSSampleMDB.jar`.
  - Click **OK**.
7. Select the **File -> Close** menu item to close the file.



“Assemble the application: Assemble the application client”

### **Assemble the application: Assemble the application client**

The application assembly tool is used to package the application client files into an application client. The end result of assembling an application client is a JAR file that contains the application client files.

1. Start the application assembly tool. On your workstation, select **Start** —> **Programs** -> **IBM Websphere** -> **Application Server v5.0** -> **Application Assembly Tool**.

**Note:** If you already have the application assembly tool open, select the **File** -> **New** -> **Application Client** menu item.

2. The **Welcome to the Application Assembly Tool** window is displayed. On the **New** tab, click **Application Client**, and click **OK**.

The application assembly tool uses a navigation bar to access various parts of the interface. The application client file has a default name and directory structure and contains general information.

3. Set up general application client file information.

- On the **General** tab, specify the following:
  - In the **Display name** field, specify `PSSampleClient`.
  - Optionally, you can specify a description in the **Description** field.
  - In the **Classpath** field, specify `PSSampleMDB.jar`.
  - Click **Browse** next to the **Main Class** field.
  - Click **Browse** next to the **Root Directory or Archive** field.
  - Locate the `client.jar` file in the `/mdbsamp` folder, and click **Select**.
  - Expand the `com/ibm/websphere/samples/messaging/pubsub` directory hierarchy.
  - Click `JMSpsSampleClient.class`.
  - Click **OK**.
  - Click **Apply**.

4. Define the resource reference for the application client. The application client uses one `TopicConnectionFactory` resource reference.

- Right-click **Resource References**, and click **New**.
- Specify the following to define a `TopicConnectionFactory` resource:
  - In the **Name** field, specify `jms/ConnectionFactory1`.
  - In the **Type** field, select `javax.jms.TopicConnectionFactory` from the drop-down list.
  - In the **Authentication** field, select **Application** from the drop-down list.
- Click **OK**.

5. Set up resource environment references for the application client. The application client uses three resource environment references. Perform the following steps to set up the first resource environment reference:

- Right-click **Resource Environment References**, and click **New**.
- On the **General** tab, specify the following:
  - In the **Name** field, specify `jms/news`.
  - In the **Type** field, select `javax.jms.Topic` from the drop-down list.
- On the **Bindings** tab, specify `thisNode/servers/server1/Sample/JMS/news` in the **JNDI Name** field, where `server1` is the name of the server you are going to run the sample on. The term `thisNode` is a special value which is replaced by the WebSphere Application Server runtime with the name of the node to which your application server belongs. It should be entered as shown above.
- Click **OK**.

6. Set up resource environment references for the application client. The application client uses three resource environment references. Perform the following steps to set up the second resource environment reference:

- Right-click **Resource Environment References**, and click **New**.
- On the **General** tab, specify the following:
  - In the **Name** field, specify `jms/sport`.
  - In the **Type** field, select `javax.jms.Topic` from the drop-down list.

- On the **Bindings** tab, specify `thisNode/servers/server1/Sample/JMS/sport` in the **JNDI Name** field, where *server1* is the name of the server you are going to run the sample on.
  - Click **OK**.
7. Set up resource environment references for the application client. The application client uses three resource environment references. Perform the following steps to set up the third resource environment reference:
    - Right-click **Resource Environment References**, and click **New**.
    - On the **General** tab, specify the following:
      - In the **Name** field, specify `jms/weather`.
      - In the **Type** field, select `javax.jms.Topic` from the drop-down list.
    - On the **Bindings** tab, specify `thisNode/servers/server1/Sample/JMS/weather` in the **JNDI Name** field, where *server1* is the name of the server you are going to run the sample on.
    - Click **OK**.
  8. Save your application client.
    - Select the **File -> Save** menu item to save your application client in the `mdbsamp` folder.
    - Save your application client as **PSSampleClient.jar**.
    - Click **OK**.
  9. Select the **File -> Close** menu item to close the file.



“Assemble the application: Assemble the application”

### Assemble the application: Assemble the application

A J2EE application is represented by and packaged in an enterprise archive (EAR) file. You must assemble your EJB module and your application client into an EAR file, which you can then install into an application server. Use the application assembly tool to package your EJB module and application client into an EAR file:

1. Start the application assembly tool. On your workstation, select **Start —> Programs -> IBM Websphere -> Application Server v5.0 -> Application Assembly Tool**.
 

**Note:** If you already have the application assembly tool open, select the **File -> New -> Application** menu item.
2. The **Welcome to the Application Assembly Tool** window is displayed. On the **New** tab, click **Application**, and click **OK**.
 

The application assembly tool uses a navigation bar to access various parts of the interface. The EAR file has a default name and directory structure and contains general EAR file information.
3. Set up general application file information.
  - On the **General** tab, specify the following:
    - In the **Display name** field, specify `MDBSamples`.
    - Optionally, you can specify a description in the **Description** field.
    - Click **Apply**.
4. Import the EJB module.
  - In the navigation bar, right-click **EJB Modules**, and click **Import**.
  - Navigate to **PSSampleMDB.jar**, and click **Open**.
  - Click **OK** in the **Confirm Values** window that appears.
5. Import the application client.



- In the navigation bar, right-click **Application Clients**, and click **Import**.
  - Navigate to **PSSampleClient.jar**, and click **Open**.
  - Click **OK** in the **Confirm Values** window that appears.
6. Save your EAR file.
    - Select the **File -> Save** menu item to save your application in the **mdbsamp** folder.
    - Save your application as **MDBSamples.ear**.
    - Click **OK**.
  7. Optionally, you can configure security on your EJB module so that all authenticated users are authorized to the enterprise bean's methods.
    - In the navigation bar, right-click **Security Roles**, and click **New**.
    - On the **General** tab, specify **ALL Role** in the **Name** field.
    - Optionally, you can specify a description in the **Description** field.
    - On the **Bindings** tab, click **Add** next to the **Special subjects: Name** field.
    - Select **AllAuthenticatedUsers**, and click **OK**.
    - On the **Bindings** tab, click **OK** to save your changes.
    - In the navigation bar, expand **Entity Beans**, expand **PSSampleMDB**, right-click **Security Roles**, and click **New**.
    - On the **General** tab, specify **All Role** in the **Name** field.
    - Optionally, you can specify a description in the **Description** field.
    - Click **OK**.
    - In the navigation bar, right-click **Method Permissions**, and click **New**.
    - Optionally, you can specify a name in the **Method permission name** field and a description in the **Description** field.
    - Next to the **Methods** field, click **Add**.
    - Expand **PSSampleMDB.jar**, and select **PSSampleMDB(\*)**.
    - Click **OK**.
    - Next to the **Role** field, click **Add**.
    - Select **All Role**, and click **OK**.
    - On the **General** tab, click **OK** to save your changes.
    - Click **Save** to save the **MDBSamples.ear** file with the updated security information.
  8. Once the file is saved, click **Close**.



“Assemble the application: Configure the application client”

### **Assemble the application: Configure the application client**

Once the EAR file is assembled, you need to configure the application client in order to correctly deploy the EAR file in a WebSphere Application Server instance. The WebSphere Application Client Resource Configuration Tool and the WebSphere Application Assembly Tool are used in this step.

1. Start the Application Client Resource Configuration Tool. On a Windows command line, specify *install\_root/bin/clientConfig*, where *install\_root* is the directory structure of the WebSphere Application Server on your workstation.
2. Open **MDBSamples.ear**, located in the **mdbsamp** folder.
3. Add the **TopicConnectionFactory** information to the EAR file.

- Expand **PSSampleClient.jar**.
- Expand **JMS Providers**.
- Expand **WebSphere JMS Provider**.
- Right-click **WAS Topic Connection Factories**, and click **New**.
- Specify the following information:
  - In the **Name** field, specify `ConnectionFactory1`.
  - In the **JNDI** field, specify `jms/ConnectionFactory1`.
  - In the **Node** field, specify the name of the node on which you are going to install the sample.
  - In the **Server** field, specify the name of the application server on which you are going to install the sample.

**Note:** If the server you are going to run the sample on is federated into a Network Deployment, specify `jmsserver` as the name of your server.
- 4. Select the **File -> Save** menu item to save your changes to `MDBSamples.ear`.
- 5. Close the Application Client Resource Configuration Tool.
- 6. Open **MDBSamples.ear** in the application assembly tool.
  - Start the application assembly tool. On your workstation, select **Start —> Programs -> IBM Websphere -> Application Server v5.0 -> Application Assembly Tool**.
  - Note:** If you already have the application assembly tool open, select the **File -> Open** menu item. Select `MDBSamples.ear` in the `mdbsamp` folder, and click **Open**.
- 7. Generate code for deployment.
  - Select the **File -> Generate code for deployment...** menu item.
  - Accept defaults for all values except database type.
  - In the **Database Type** field, select **DB2UDBISERIES** from the drop-down list.
  - Click **Generate now**.

The tool generates a file called `Deployed_MDBSamples.ear`, which you use in the next steps.
- 8. Once the deployed code is generated, click **Close**.



“Step 3: Configure the server”

### Step 3: Configure the server

You can create multiple WebSphere Application Server instances (each containing a single application server) on a single iSeries server. You can also use multiple instances to create separate environments for application development and application testing. It is recommended that you create a new WebSphere Application Server instance to configure the message-driven beans application sample in order to preserve the Default Server with its sample configuration.

1. “Configure the server: Create a new instance” on page 29  
Create a new WebSphere Application Server instance to run the sample application.
2. “Configure the server: Enable your instance to use Java Message Service (JMS)” on page 29  
The instance you created must be configured with Java Message Service resources before the application can be deployed to the instance.
3. “Configure the server: Set up security (optional)” on page 31  
The WebSphere Application Server security system controls access to resources and the administrative console. Perform this step to enable security.

**Continue**



“Configure the server: Create a new instance”

### **Configure the server: Create a new instance**

To create a new instance, run the `crtwasinst` script from Qshell the command line. This script creates all new server directories and sets up the correct authorities. To run this script, your iSeries user profile must have `*ALLOBJ` authority.

1. Enter the Start Qshell (STRQSH) command on an OS/400 command line.
2. On the Qshell command line, use the `cd` command to change to the directory that contains the script. For example:

```
cd /QIBM/ProdData/WebAS5/Base/bin
```

3. Run the `crtwasinst` script. Enter this command:

```
crtwasinst -instance mdbconfig -portblock portblock
```

where *mdbconfig* is the name of the instance that is created and *portblock* is the first of a block of 13 consecutive ports. For more information on the `crtwasinst` script and additional parameters, see *The crtwasinst script*.

**Continue**



“Configure the server: Enable your instance to use Java Message Service (JMS)”

**Configure the server: Start the instance:** To start an instance, run the `startServer` script from the Qshell command line. To run the script, follow these steps:

1. Enter the Start Qshell (STRQSH) command on an OS/400 command line.
2. On the Qshell command line, use the `cd` command to change to the directory that contains the script. For example:

```
cd /QIBM/ProdData/WebAS5/Base/bin
```

3. Run the `startServer` script. Enter this command:

```
startServer -instance mdbconfig serverName
```

where *mdbconfig* is the name of the instance you want to start and *serverName* is the name of the server you want start. For more information on the `startServer` script and additional parameters, see *The startServer script*.

**Continue**



“Configure the server: Enable your instance to use Java Message Service (JMS)”

### **Configure the server: Enable your instance to use Java Message Service (JMS)**

A JMS provider enables asynchronous messaging based on the Java Messaging Service (JMS). It provides J2EE connection factories to create connections for specific JMS queue or topic destinations. See the *Administer JMS resources* topic for more information.

1. Enable embedded JMS for your application server. For more information, see Administer the embedded JMS server.
2. “Configure the server: Start the instance” on page 29.
3. Start the WebSphere Application Server administrative console. To start the console, open this URL in your browser:  
`http://your.server.name:port/admin`  
 where *your.server.name* is the hostname of your iSeries server and *port* is your administrative port number. See the *The administrative console* topic for more information.
4. Create a new J2C Authentication Data Entry for use with JMS connection factories.
  - In the topology tree, expand **Security** -> **JAAS Configuration**, and click **J2C Authentication Data**.
  - Click **New**.
  - In the **Alias** field, specify the name of the authentication data entry. For the message-driven beans sample application, specify `mdb`.
  - You can use any valid user profile and password.  
**Note:** The user profile and password you specify must also be a valid iSeries server user ID and password.
  - Click **OK**.
5. Configure your JMS connection factory. A topic connection factory is used to create connections to the associated JMS provider of JMS topic destinations for publish and subscribe messaging.
  - In the topology tree, expand **Resources**, and click **WebSphere JMS Provider**.
  - Scroll down to the bottom of the page, and click **WebSphere Topic Connection Factories**.
  - Click **New**.
  - In the **Name** field, specify `SampleJMSTopicConnectionFactory`.
  - In the **JNDI name** field, specify `Sample/JMS/TCF`.
  - In the **Component-managed Authentication Alias** field, select `mdb` from the drop-down list. This is the Authentication Data Entry you created in the previous step.
  - In the **Container-managed Authentication Alias** field, select `mdb` from the drop-down list.
  - In the **Client ID** field, specify `MDBSampleClientID`.
  - Click **Apply**.
6. Configure your JMS topic resources. For the message-driven beans sample application, you must create four separate topics.
  - In the topology tree, expand **Resources**, and click **WebSphere JMS Provider**.
  - Scroll down to the bottom of the page, and click **WebSphere Topic Destinations**.
  - Click **New**.
  - In the **Name** field, specify `Sample.JMS.listen`.
  - In the **JNDI name** field, specify `Sample/JMS/listen`.
  - In the **Topic** field, specify `Sample/JMS/listen`.
  - Click **OK**.
  - Repeat the steps above to create the next topic, using the following values:
    - In the **Name** field, specify `Sample.JMS.news`.
    - In the **JNDI name** field, specify `Sample/JMS/news`.
    - In the **Topic** field, specify `Sample/JMS/news`.
    - Click **OK**.
  - Repeat the steps above to create the next topic, using the following values:
    - In the **Name** field, specify `Sample.JMS.sport`.
    - In the **JNDI name** field, specify `Sample/JMS/sport`.
    - In the **Topic** field, specify `Sample/JMS/sport`.

- Click **OK**.
  - Repeat the steps above to create the next topic, using the following values:
    - In the **Name** field, specify `Sample.JMS.weather`.
    - In the **JNDI name** field, specify `Sample/JMS/weather`.
    - In the **Topic** field, specify `Sample/JMS/weather`.
    - Click **OK**.
7. Configure your JMS listener port.
    - In the topology tree, expand **Servers**, and click **Application Servers**.
    - Click your application server.
    - Scroll down to the bottom of the page, and click **Message Listener Service**.
    - Click **Listener Ports**, and click **New**.
    - In the **Name** field, specify `SamplePubSubListenerPort`.
    - In the **ConnectionFactory JNDI name** field, specify `Sample/JMS/TCF`.
    - In the **Destination JNDI name** field, specify `Sample/JMS/listen`.
    - In the **Maximum sessions** field, specify 5.
    - In the **Maximum retries** field, specify 2.
    - In the **Maximum messages** field, specify 1.
    - Click **OK**.
  8. Save your administrative configuration.
    - In the **Messages** box at the top of the page, click **Save**.
    - Click **Save** to save your changes to the master configuration.
  9. Click **Logout** on the toolbar, and close your browser.
  10. Restart your application server instance.
    - Enter the Start Qshell (STRQSH) command on an OS/400 command line.
    - On the Qshell command line, enter this command:  
`stopServer -instance mdbconfig serverName`  
 where *mdbconfig* is the name of the instance you want to stop and *serverName* is the name of the server you want stop.
    - On the Qshell command line, enter this command:  
`startServer -instance mdbconfig serverName`  
 where *mdbconfig* is the name of the instance you want to start and *serverName* is the name of the server you want start.



“Configure the server: Set up security (optional)”

### Configure the server: Set up security (optional)

The WebSphere Application Server security server controls access to application server resources (servlets, JSPs, HTML files and EJBs). It also controls access to the administrative console and to some administrative tools. The security system is not responsible for protecting native OS/400 resources (such as database files and program objects).

When you install WebSphere Application Server for iSeries, the product creates a default application server instance. This instance is named default. Security is initially disabled in the default application

server instance and any other application server instance you create. Unless you believe your iSeries server is adequately protected by a firewall or some other mechanism, immediately enable security after creating an instance.

This step enables local operating system security, which uses an OS/400 user profile and password to secure the application server. For more information on WebSphere Application Server security, including security configuration options other than those for the local operating system, see the *WebSphere Application Server security* topic.

Perform the following steps to enable security:

1. Start the WebSphere Application Server administrative console. To start the console, open this URL in your browser:  
`http://your.server.name:port/admin`  
where *your.server.name* is the hostname of your iSeries server and *port* is your administrative port number. See the *The administrative console* topic for more information.
2. Expand **Security**.
3. Expand **User Registries**.
4. Click **Local OS**.
5. In the **Server User ID** field, specify a valid OS/400 user profile name.  
**Note:** A user profile that is a part of a group profile cannot be used to configure the LocalOS user registry. A group profile is assigned a unique group ID number, which is not assigned to a regular user profile. Run the Display User Profile (DSPUSRPRF) command to determine if the user profile you want to use as the Server User ID has a defined group ID number. If the Group ID field is set to \*NONE, the user profile can be used as the administrative user ID.
6. In the **Server User Password** field, specify the password for the user profile you specified.
7. Click **OK**.
8. In the topology tree, click **Global Security**.
9. On the **General Properties** page, check **Enabled**. Accept the remaining default property values.
10. Click **OK**.
11. Click **Save** on the toolbar to save the configuration.
12. Click **Save** again to update the master repository with your changes.
13. Click **Logout** on the toolbar.
14. Restart your application server instance.
  - Enter the Start Qshell (STRQSH) command on an OS/400 command line.
  - On the Qshell command line, enter this command:  
`stopServer -instance mdbconfig serverName`  
where *mdbconfig* is the name of the instance you want to stop and *serverName* is the name of the server you want stop.
  - On the Qshell command line, enter this command:  
`startServer -instance mdbconfig serverName`  
where *mdbconfig* is the name of the instance you want to start and *serverName* is the name of the server you want start.

The next time you start the administrative console, use the iSeries user profile that you specified when you enabled security to log into the console.

**Continue**



“Step 4: Deploy the application”

## Step 4: Deploy the application

After you create the EAR file and set up your Java Message Service (JMS) resources, the next step is to install the EAR file into your WebSphere Application Server instance. To deploy the EAR file into your application server runtime, use the Application Installation Wizard. To use the wizard, follow these steps:

1. Start the WebSphere Application Server administrative console. To start the console, open this URL in your browser:  
`http://your.server.name:port/admin`  
where *your.server.name* is the hostname of your iSeries server and *port* is your administrative port number. See the *The administrative console* topic for more information.
2. In the topology tree, expand **Applications**, and click **Install new application**.
3. In the **Server path** field, specify `/mbsamp/Deployed_MDBSamples.ear`.  
**Note:** The file resides on your iSeries server; however, you can map a network drive to access the iSeries server from your workstation and use the Local path installation. For more information, see “Step 1: Verify the prerequisites” on page 19.
4. Click **Next**.
5. On the **Preparing for application install** page, accept all default values, and click **Next**.
6. On the **Install New Application: Step 1** page, accept all default values, and click **Next**.
7. On the **Install New Application: Step 2** page, accept all default values, and click **Next**.
8. On the **Install New Application: Step 3** page, check the module, and click **Next**.
9. On the **Install New Application: Step 4** page, click **Finish**.
10. Once you receive a message that the EAR file installed correctly, click **Save** in the toolbar. Click **Save** again to update the master repository with your changes.
11. In the topology tree, expand **Applications**, and click **Enterprise Applications**.
12. Check **MDBSamples**, and click **Start**.  
You will receive a message stating that the application started successfully.

**Continue**



“Step 5: Verify the application”

## Step 5: Verify the application

After the successful deployment of the message-driven beans sample application to your WebSphere Application Server instance, place a copy of the installed EAR file in your `installableApps` folder, and run the application on your iSeries server.

1. Enter the Start Qshell (STRQSH) command on an OS/400 command line.
2. To locate the installed EAR file, enter this command:  
`cd /QIBM/UserData/WebAS5/Base/mdbconfig/installedApps/hostname_mdbconfig`  
where *hostname* is the host name of your iSeries server.

3. Copy the message-driven beans sample application EAR file to your installableApps directory. Enter this command:

```
cp -R MDBSamples.ear /QIBM/UserData/WebAS5/Base/mdbconfig/installableApps
```

The successful deployment of the EAR file must be tested to make sure that all JMS resources have been correctly set up. Perform the following steps to run the application client on the iSeries server:

1. Enter the Start Qshell (STRQSH) command on an OS/400 command line.
2. Enter the following command to run the application client:

```
/QIBM/ProdData/WebAS5/Base/bin/launchClient -instance mdbconfig
/qibm/userdata/webas5/base/mdbconfig/installableApps/MDBSamples.ear
-CCjar=PSSampleClient.jar -CCbootstrapPort=your_bootstrap_port
-CCsoapConnectorPort=your_soap_port -verbose -topic news -msg
"Message Text"
```

where *your\_bootstrap\_port* is the Name Service port of your WebSphere Application Server instance, and *your\_soap\_port* is the SOAP Connector port of your WebSphere Application Server instance.

**Note:** Port numbers were assigned after you created your instance. To display the port numbers assigned to your instance, see *Display instance properties*.

The JMS client sends the message as described above and reports the progress. The output is similar to the following:

```
IBM WebSphere Application Server, Release 5.0
J2EE Application Client Tool
Copyright IBM Corp., 1997-2002
WSCL0012I: Processing command line arguments.
WSCL0013I: Initializing the J2EE Application Client Environment.
Attaching Java program to /tmp/WSTMPCC61074.tmp/PSSampleClient.JAR.
WSCL0035I: Initialization of the J2EE Application Client Environment has completed.
WSCL0014I: Invoking the Application Client
class com.ibm.websphere.samples.messaging.pubsub.JMSpsSampleClient
Topic:news.
Sending message: 'Message Text'
Retrieving a TopicConnectionFactory from JNDI
com.ibm.ws.client.applicationclient.ClientJMSTopicConnectionFactory
Retrieving Topic from JNDI
Creating a Connection
Created a Connection
Starting the Connection
Creating a Session
Creating a TopicPublisher
Creating a TextMessage
Publish the message to topic://Sample/JMS/news?brokerVersion=1
Message ID is ID:c1d4d840e6c1e26dd3d7c1d9f2f4f7d43d624d980000d022
Closing TopicPublisher
Closing Session
Closing Connection
End of Sample
$
```



---

## Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM 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:

IBM Director of Licensing  
IBM Corporation  
500 Columbus Avenue  
Thornwood, NY 10594-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Software Interoperability Coordinator, Department 49XA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both: WebSphere

AS/400  
e (logo)  
IBM  
iSeries  
Operating System/400  
OS/400  
400

Lotus, Freelance, and WordPro are trademarks of International Business Machines Corporation and Lotus Development Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

---

## Terms and conditions for downloading and printing publications

Permissions for the use of the publications you have selected for download are granted subject to the following terms and conditions and your indication of acceptance thereof.

**Personal Use:** You may reproduce these Publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these Publications, or any portion thereof, without the express consent of IBM.

**Commercial Use:** You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations. IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

All material copyrighted by IBM Corporation.

By downloading or printing a publication from this site, you have indicated your agreement with these terms and conditions.







Printed in USA