IBM Product Connectivity Scenarios information center
Version 1 Release 0

Connecting WebSphere Application Server to WebSphere MQ; getting started

IBM

Contents

Figures

v

Chapter 1. Scenario: Connecting WebSphere Application Server to WebSphere MQ; getting started

By connecting to WebSphere® MQ, Java™ EE applications on WebSphere Application Server can consume and work with messages from WebSphere MQ. Starting with an existing WebSphere MQ installation, this scenario leads you through the key tasks required to install and connect WebSphere Application Server on the same Windows computer.

This scenario was developed using a sample WebSphere MQ installation, and uses sample applications to demonstrate the use of WebSphere Application Server (full profile) connected to WebSphere MQ. If you want to try the scenario, you can set up a copy of the sample installation as described in the scenario. You can use the sample applications provided with the scenario to verify your progress through each stage.

If you are interested in connecting WebSphere Application Server Liberty profile to WebSphere MQ, see the scenario Connecting WebSphere Application Server Liberty profile to WebSphere MQ.

Notes

This scenario applies to the use of either WebSphere Application Server Version 7 or Version 8 with WebSphere MQ version 7.

> This scenario was developed and tested with WebSphere Application Server Version 7 and WebSphere MQ version 7. If you use WebSphere Application Server Version 8, you will notice some differences like using the IBM Installation Manager instead of the installation wizard described in this scenario.

Optional information to help you learn while implementing the solution

> The scenario contains blocks of optional information marked by Why? or What else?. You do not need to read this information to complete the scenario, but might choose to learn more:

> **Why?** Describes *why* you are instructed to do something. For example:

> > **Why am I doing this?**
> > You use the **Scope** property to set the level at which the activation specification is visible. The cell scope is the highest level, giving the activation specification the greatest visibility.

> **What else?**
> > Describes *what else* you might do, or want to learn about, related to what you are reading in the main window. For example:

1

> **What else might I do or be interested in?**
>
> You can also create activation specifications at other levels.
> For example, if you have multiple servers you might create
> an activation specification for each server, using the server
> scope, so that you can specify different settings to be used for
> each server.

Tip: Some "Why?" and "What else?" information provides links that
would take you to information resources outside the scenario. To complete
a scenario, you do not need to follow such links; they are provided only as
optional aids for your learning.

Related information:

Scenario: Connecting WebSphere Application Server to WebSphere MQ; getting
started on Linux

➦ Scenarios and Patterns information center
This scenario in the Scenarios and Patterns information center

➦ Product web page
WebSphere Application Server product web page

➦ Library page
WebSphere Application Server library page

➦ Product web page
WebSphere MQ product web page

➦ Library page
WebSphere MQ library page

"What's new?"
A summary of what's new in this scenario.

What's new?

This scenario has been updated to add an electronic book (ebook) version of the
scenario in EPUB format.

The EPUB file is linked in the "Related resource" list on the scenario welcome
page.

You can read EPUB files with many eBook readers; for example, the iBooks app for
Apple products, a variety of ebook reader apps for Android devices, and dedicated
eBook reader devices like the Barnes & Noble Nook eReader range and the Sony
Reader. You can also read ePub files in your web browser with EPUBReader
add-on for Firefox and MagicScroll for Chrome. If you want to read an EPUB file
on the Amazon Kindle, you can convert EPUB files to kindle format with a
converter like Calibre.

Related information:

Chapter 1, "Scenario: Connecting WebSphere Application Server to WebSphere MQ;
getting started," on page 1
By connecting to WebSphere MQ, Java EE applications on WebSphere Application
Server can consume and work with messages from WebSphere MQ. Starting with
an existing WebSphere MQ installation, this scenario leads you through the key
tasks required to install and connect WebSphere Application Server on the same

Windows computer.

Chapter 2. Planning the solution

You can connect WebSphere Application Server to WebSphere MQ in a number of different ways. Review the topics in this section to understand what is covered in this scenario, the reasons why a business might want to follow the scenario, the user roles involved, and an overview of the solution proposed by the scenario.

Assumptions

This scenario makes several assumptions about your system, such as the version of the products that you are using.

This scenario assumes the following points:

- You are using the Windows operating system.
- This scenario was developed by using 7.0.0 Fix Pack 13. Instructions are provided for obtaining and installing fix packs for WebSphere Application Server.

> **Why am I doing this?**
> Earlier versions of the products might require configuration that is not covered in this scenario. For example, if you install WebSphere Application Server without any fix packs, you must manually maintain the version of the WebSphere MQ resource adapter within WebSphere Application Server. See Maintaining the WebSphere MQ resource adapter for more details.

Note: This scenario applies to the use of either WebSphere Application Server Version 7 or Version 8 with WebSphere MQ version 7.

- You do not have security configured for WebSphere MQ. This scenario does not describe security configuration. However, if you do have security configured for WebSphere MQ you should still be able to complete the scenario.
- You use graphical user interfaces of WebSphere MQ and WebSphere Application Server, rather than the command-line equivalents.

Business overview

A company wants to add a Java EE application on WebSphere Application Server to consume and work with messages from an existing messaging infrastructure provided by WebSphere MQ.

To date, company A has a business solution that uses WebSphere MQ for its messaging infrastructure. Business users interact with a stand-alone client application, for example to register an order, which sends a message into the infrastructure. Such messages are transported by the infrastructure for processing by some separate WebSphere MQ application, possibly in a different business unit. The client application waits for a reply message, to provide the business user with an appropriate response.

Changes to the business model of the company mean that the processing performed on messages is inadequate, yet the processing is not implemented in a standards-based way that can best be developed for the future. The company

recognizes that the dynamic business needs of today are driving IT departments to implement standards-based computing. The company wants to use standards-based programming to provide the updated processing that must be performed on messages before they are returned to business users. The company decides that they want to use the Java Message Service (JMS), the Java EE standard for messaging, that provides a standard API for applications to use when performing enterprise messaging with application portability. The company also recognize that the Java EE standard streamlines application development, and by using standards they can create reusable, platform-independent modules.

The company decides to add a Java EE application on WebSphere Application Server to consume and work with messages in the solution. WebSphere Application Server is a proven, standards-based platform that offers enhanced support for standards, emerging technology and a choice of development frameworks that simplifies programming models.

With WebSphere Application Server, the company appreciates that as their business demands increase, they can take advantage of the latest standards and programming models that WebSphere Application Server supports or scale up their business capability by adopting higher-performance features or editions of WebSphere Application Server.

Related information:

➦ WebSphere Application Server features and benefits web page

User roles and interactions

Roles used throughout these scenarios, with interactions between roles for scenario tasks.

Although roles to develop the solution are listed, this scenario focuses on deploying a solution that connects WebSphere Application Server to WebSphere MQ.

Develop the solution

Develop the software aspects of the solution.

Software Architect
    The Software Architect is responsible for dividing the required function between the components that make up a software solution. This person works with the specifications and standards used by existing IT systems, and determines where enhancements or new components must be written by the Developer.

Developer
    The Developer is responsible for creating and testing the software components and linking them together. In some cases the Developer might need information from the Administrator, for example if the Developer must use an existing queue name in their software code. After the Developer finishes creating the software components, the Developer gives those components to the Administrator for deployment.

Deploy the solution

Deploy the solution for production use, by installing, configuring, and testing the components that provide the solution.

Administrator

The Administrator installs and configures the components that support the solution, in this case WebSphere MQ and WebSphere Application Server. The business might have a different Administrator for each product, or one Administrator might perform installation and configuration for all components. If the Administrators are different they might need to exchange information. For example, in this scenario the WebSphere MQ administrator must give the name of the queue and queue manager to the WebSphere Application Server administrator. The Administrator also deploys the software components given to them by the Developer.

Test Implementer

The Test Implementer runs the tests to validate the solution and that the solution is ready for production use. For example, can the solution be started, stopped, backed up, and recovered after a system failure as well as be maintained?
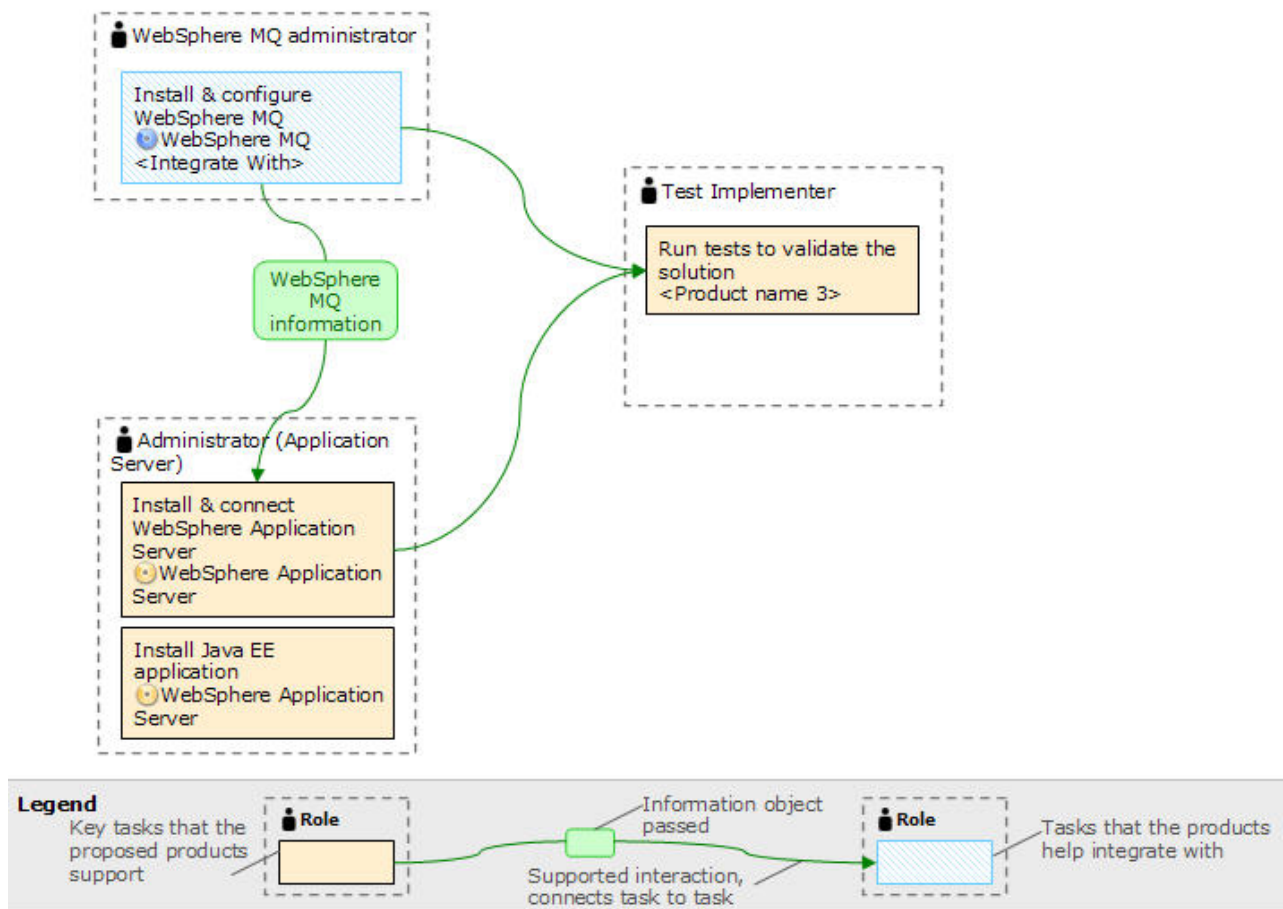

Figure 1. User role interactions to deploy the solution

Technical solution

This scenario describes one way of connecting WebSphere Application Server to a WebSphere MQ system, by using JMS, JCA and JNDI.

Overview: Initial IT configuration

Company A has an existing IT configuration that uses WebSphere MQ for its messaging infrastructure. This scenario describes adding WebSphere Application Server to that initial messaging infrastructure.

The initial IT configuration includes several components that an administrator configures or uses, as shown in Figure 2 on page 10:

JMS application

A stand-alone application that business users interact with, for example to register an order. The application uses the Java Message Service (JMS) for asynchronous messaging.

> **Why am I doing this?**
> - JMS is the Java EE messaging standard that is widely supported. JMS-based applications are therefore portable across many messaging products.
> - JMS provides a level of abstraction from the details of the messaging layer, simplifying the application development process.
> - JMS provides asynchronous communication, enabling applications to run without having to wait for a reply, unlike tightly coupled systems such as remote procedure call (RPC).
> - Applications that use JMS do not directly specify details to access resources. Instead, they look up and use administered JMS objects such as a connection factory and a destination.

> **What else might I do or be interested in?**
> For some situations, other messaging standards might be more suitable than JMS. For example IBM® Message Service Clients for C, C++ and .NET, also known as XMS, are APIs that provide similar benefits to JMS for non-Java applications. XMS is therefore more suitable if you are using the .NET platform, or you want to integrate existing C++ applications with newer Java EE applications.

The application uses point-to-point messaging to send messages to a queue in the infrastructure and processes reply messages, to provide the business user with an appropriate response.

> **Why am I doing this?**
> In this messaging model, an application sends a message to a queue, and another application receives the message from the queue and acknowledges receipt of the message. This model is the simplest form of messaging because it involves only two endpoints. This model is also the most appropriate for the scenario sample application: a single client requests information from a single server.

Queue manager sampleQM
> The WebSphere MQ queue manager that provides the initial messaging infrastructure. It hosts the queue that the JMS application works with.

Q1 [Message queue]
> The WebSphere MQ queue that the JMS application sends messages to.

JNDI namespace
> A Java Naming and Directory Interface (JNDI) namespace is used to hold JMS administered objects, which applications can use to connect to WebSphere MQ and access destinations to send or receive messages.

> Any application or process with access to the JNDI namespace can use the same administered objects. The properties of the administered objects can be changed in JNDI, with all the applications or processes able to benefit from those same changes.

Initial context
> > An initial context defines the root of the JNDI namespace. To use WebSphere MQ Explorer to create and configure administered objects, you first add an initial context that defines the root of the JNDI namespace. Similarly, a JMS application first obtains an initial context, before it can retrieve administered objects from the JNDI namespace.

Connection factory, myCF
> > A JMS connection factory object defines a set of standard configuration properties for connections. An application uses a connection factory to create a connection to WebSphere MQ.

Destination, myQueue
> > A JMS destination can be a topic or a queue. In this scenario, the destination is a queue, and identifies the WebSphere MQ queue

that applications send messages to, or from which an application receives messages, or both. An application looks up the destination in the JNDI namespace to create a connection to the WebSphere MQ queue.
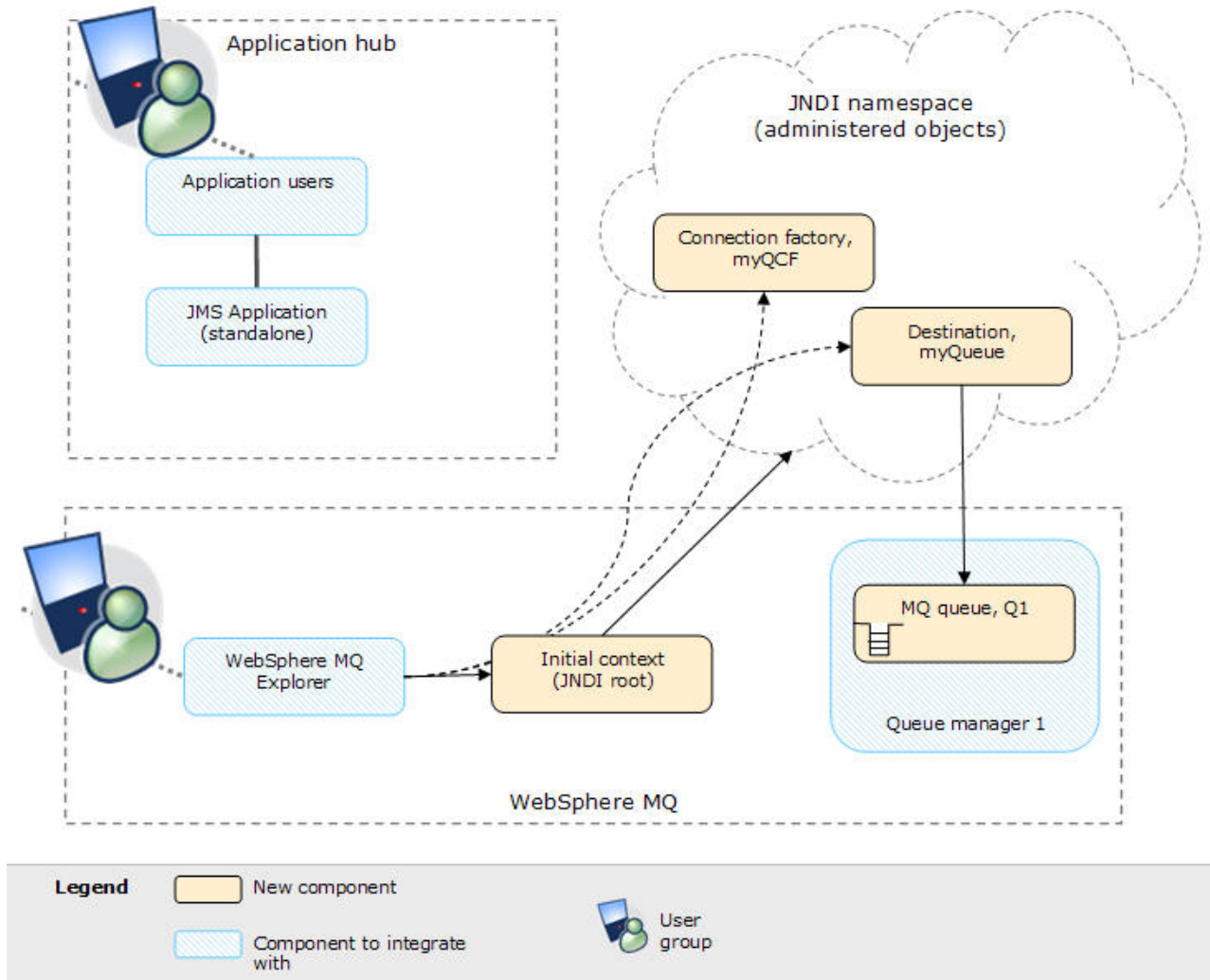
Figure 2. Initial IT configuration. The initial IT configuration includes an initial context, added for the WebSphere MQ Explorer to connect to the root of the JNDI namespace. The JNDI namespace includes a connection factory, added for the sample JMS application to use to connect to WebSphere MQ, and a destination, added for the sample JMS application to connect to the WebSphere MQ queue. That WebSphere MQ queue has also been added into the initial IT configuration for use by the sample JMS application.

Tip: This scenario was developed by using a sample version of the initial IT configuration described. If you want to try out the scenario, you can either follow the instructions to configure WebSphere Application Server to communicate with your own WebSphere MQ, or set up a copy of the sample IT configuration as described in "Creating a sample initial IT configuration" on page 16.

Overview: The delivered logical topology

The company adds a Java EE application on WebSphere Application Server to consume and work with messages from an existing messaging infrastructure provided by WebSphere MQ.

The delivered IT configuration includes several components that an administrator configures or uses, as shown in Figure 3 on page 13.

WebSphere MQ as a *messaging provider* **for WebSphere Application Server**
> The WebSphere MQ messaging provider in WebSphere Application Server makes JMS messaging available to WebSphere Application Server applications by using the existing capabilities in the WebSphere MQ environment.

> ---
> **Why am I doing this?**
>
> WebSphere Application Server applications can interact with WebSphere MQ destinations to send and receive messages in the same way as any JMS application in the WebSphere MQ environment.
> ---

> ---
> **What else might I do or be interested in?**
>
> - You can connect a WebSphere MQ network to a service integration bus within WebSphere Application Server, by using WebSphere MQ links. A WebSphere MQ link provides support for sender-receiver channels between the service integration bus and a WebSphere MQ queue manager or queue-sharing group. This option requires more complex configuration in WebSphere Application Server; you must configure a service integration bus and messaging engines.
> - You can add a WebSphere MQ server (representing a queue manager or queue-sharing group) as a member of a service integration bus within WebSphere Application Server. A WebSphere MQ server provides a direct client or bindings connection between a service integration bus and queues on a WebSphere MQ queue manager. This option also requires you to configure a service integration bus and messaging engines in WebSphere Application Server.
> ---

Java EE application
> The application consumes and works with messages on the WebSphere MQ queue, Q1. This application runs on an *application server* of the WebSphere Application Server product that has been installed and connected with WebSphere MQ.

> The sample application in this scenario provides a *message-driven bean (MDB)* as an asynchronous message consumer. When a message arrives at the queue, the MDB automatically processes the message without the application having to explicitly poll the queue.

> ---
> **Why am I doing this?**
>
> MDBs are activated by the EJB container in WebSphere Application Server on receipt of a message. A typical MDB performs messaging functions, and calls one or more session beans to perform business logic. Because of this separation of function, you can easily change and reuse units of business logic without affecting the messaging function of the application.
> ---

Application server
A server program in WebSphere Application Server that provides the execution environment for Java EE application programs.

WebSphere Application Server JNDI namespace
WebSphere Application Server includes a name server which provides access to the following JMS administered objects through the Java Naming and Directory Interface (JNDI). The use of JNDI, the connection factory, and the destination, are the same as described for the initial IT configuration in "Overview: Initial IT configuration" on page 8.

Activation specification, myActSpec
A JMS activation specification can be associated with one or more MDBs and provides the configuration necessary for them to listen for messages arriving at a destination. Activation specifications process inbound messages to the MDB.

> **Why am I doing this?**
> Activation specifications are part of the Java EE Connector architecture (JCA) 1.5 standard. JCA 1.5 provides a standard way to integrate JMS providers, such as WebSphere MQ, with Java EE application servers such as WebSphere Application Server.

> **What else might I do or be interested in?**
> Use of listener ports is an older configuration method for MDBs to listen for messages arriving at a destination. The use of listener ports is stabilized in WebSphere Application Server Version 7 and later. If you are using WebSphere Application Server for the first time avoid using listener ports.

Connection factory, myCF
A JMS connection factory object defines a set of standard configuration properties for connections. An application uses a connection factory to create a connection to WebSphere MQ.

If your application uses an MDB, as the sample application does, the connection factory is used for outbound messages only; inbound messages are received by the activation specification.

Destination, myQueue
A JMS destination can be a topic or a queue. In this scenario, the destination is a queue, and identifies the WebSphere MQ queue that applications send messages to, or from which an application receives messages, or both. An application looks up the destination in the JNDI namespace to create a connection to the WebSphere MQ queue.

In this scenario, when creating JMS objects in WebSphere Application Server, you specify full details rather than use a *client channel definition table (CCDT)* that has been exported from WebSphere MQ.

Figure 3. Delivered logical topology. The high-level logical topology diagram for new software functions delivered by integrating WebSphere Application Server into the initial WebSphere MQ messaging infrastructure. This delivered logical topology shows the new functions as new or extended logical components, integrated with the software functions and products from the initial logical topology.

Production physical topology and product mapping

The production physical topology specifies the recommended operational environment (machines, operating systems, and software products) for running the applications and services of the solution.

Figure 4. Production physical topology

Node	Operating system and hardware	Software
Host 1 (runtime node)	• Windows Server 2003 or 2008, Standard or Enterprise Edition. You can use Windows 7, or for 32-bit environments, Windows Vista or Windows XP Professional, but these operating systems are not supported for production use of WebSphere Application Server. • Intel EM64T, AMD Opteron, or for 32-bit environments, Intel Pentium processor at 2.33 GHz or faster • 1 GB physical memory • 3 GB free disk space	• WebSphere MQ • WebSphere Application Server Prerequisite software: • A web browser for use by the Launchpad, Windows Explorer, and WebSphere Application Server administrative console: Microsoft Internet Explorer Version 6.0, or Mozilla-based browsers Version 1.7 or higher

Chapter 3. Implementing the solution

Implementing the solution in this scenario involves connecting WebSphere Application Server to WebSphere MQ, which provided the initial messaging infrastructure for a company.

Before you begin

The starting point for this scenario is an existing, verified, WebSphere MQ installation as the initial messaging infrastructure.

This scenario was developed using a sample WebSphere MQ installation, and uses sample applications to demonstrate the use of WebSphere Application Server connected to WebSphere MQ. If you want to try the scenario, you can set up a copy of the sample installation as described in the scenario. You can use the sample applications provided with the scenario to verify your progress through each stage.

About this task

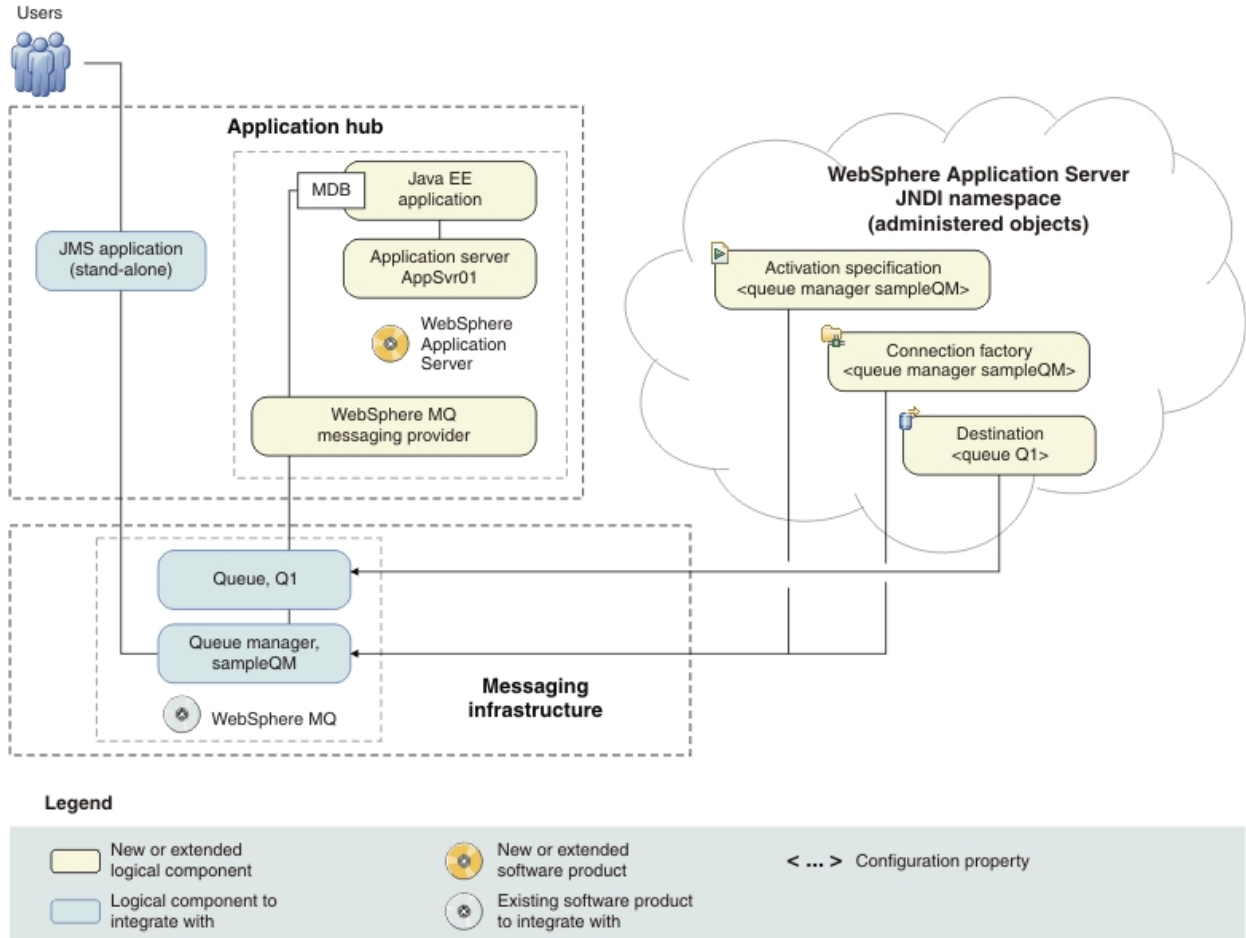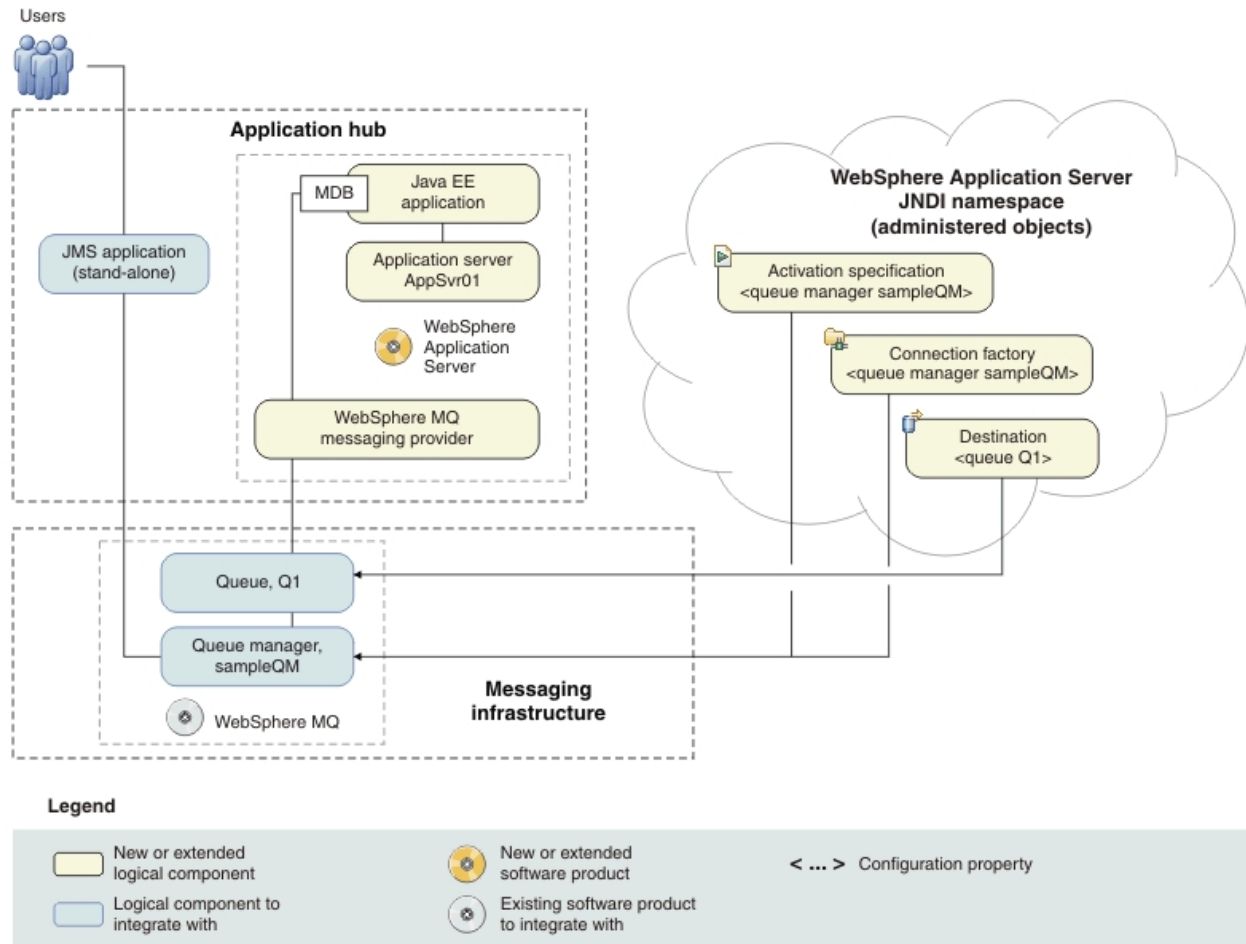*Figure 5. Delivered logical topology, including WebSphere Application Server.* The high-level logical topology diagram for new software functions delivered by integrating WebSphere Application Server into the initial WebSphere MQ messaging infrastructure. This delivered logical topology shows the new functions as new or extended logical components, integrated with the software functions and products from the initial logical topology.

After connecting WebSphere Application Server to WebSphere MQ, you can use the connection by deploying a Java EE application to run on WebSphere Application Server. This scenario describes use of a sample Java EE application that is deployed onto WebSphere Application Server and used to verify the solution.

Creating a sample initial IT configuration

This scenario was developed by using a sample initial IT configuration. Follow the instructions to set up the sample initial IT configuration to try out the scenario in the same way as it was originally developed.

Installing WebSphere MQ

To complete this scenario you must have a functioning WebSphere MQ messaging network. This task provides basic instructions to install WebSphere MQ if you do not already have it installed on your system.

Before you begin

- You must have local administrator authority when you are installing. Define this authority through the Windows facilities.
- Ensure that the machine name does not contain any spaces.
- Ensure that you have sufficient disk space, up to 600 MB, to fully install WebSphere MQ for Windows.
- Determine whether any WebSphere MQ users are defined on Windows 2000 or later domain controllers.

Why am I doing this?

WebSphere MQ checks that only authorized users can access queue managers or queues. When a user attempts such access, WebSphere MQ uses its own local account to query information about the user. However, domain controllers running on Windows 2000 or later can be configured such that WebSphere MQ cannot use local accounts to make these queries. In this situation, you must provide WebSphere MQ with a special account to use. This account is necessary when both of the following conditions apply:

  – Any domain controller on your network is running on Windows 2000 or later

  – Local user accounts are not authorized to query the group membership of the domain user accounts

If you require a special account, refer your domain administrator to the instructions in Creating and setting up domain accounts for WebSphere MQ in the WebSphere MQ information center.

If you want to investigate the requirements for installing WebSphere MQ in more detail, see System Requirements for IBM MQ.

About this task

This task describes the basic steps to install WebSphere MQ on the Windows operating system.

The installation programs contain links to further information if you require it during the installation process. The installation process has four parts:

1. Use the Launchpad to check and install software requirements, specify network information, and start the WebSphere MQ installation wizard.
2. Use the WebSphere MQ installation wizard to install the software, and start the Prepare WebSphere MQ wizard.
3. Use the Prepare MQ wizard to start the WebSphere MQ service and start the Default Configuration wizard.
4. Optionally, use the Default Configuration wizard to create objects for verifying the installation using the Postcard application supplied with WebSphere MQ.

Procedure

1. Insert the WebSphere MQ for Windows Server DVD into the DVD drive. If autorun is enabled, the installation process starts. Otherwise, double-click the Setup icon in the root folder of the DVD to start the installation process.
2. Use the buttons on the left of the Launchpad to review and, if necessary, modify the software requirements and network configuration.

WebSphere Eclipse Platform 3.3 is required for using the WebSphere MQ Explorer and Managed File Transfer Application components. You can use the launchpad to install this software if required: after you choose whether to install from a DVD or your file system, the Launchpad opens a dialog box with the Eclipse `Setup.exe` program selected. Click **Open** to run the WebSphere Studio IES 3.3 installation wizard. After installing, click **Refresh** in the Launchpad to update the status of the software requirements.

3. On the WebSphere MQ Installation page of the Launchpad, click **Launch IBM WebSphere MQ Installer** to start the WebSphere MQ installation wizard.

4. Follow the instructions in the wizard, choosing a typical setup type, to install WebSphere MQ. The following features are installed:

   - WebSphere MQ Server
   - WebSphere MQ Explorer: a graphical interface for administering and monitoring WebSphere MQ resources.
   - WebSphere MQ Development Toolkit
   - Java and .NET Messaging and Web Services

   At the end of the process, the WebSphere MQ Setup window displays the message `Installation Wizard Completed Successfully`. Click **Finish**; the Prepare WebSphere MQ wizard starts automatically.

5. Follow the instructions in the Prepare WebSphere MQ wizard.

   ┌─ **Why am I doing this?** ─────────────────────────────────┐
   │ The Prepare WebSphere MQ wizard helps you to configure WebSphere │
   │ MQ files and a user account for your network. You must run the wizard │
   │ to configure the WebSphere MQ Service before you can start any queue │
   │ managers. │
   └──┘

   a. On the first page of the wizard, click **Next**. The wizard performs some configuration and displays status messages, then displays the WebSphere MQ Network Configuration page. The contents of this page varies according to your use of domain controllers.

   b. Follow the instructions for network configuration then click **Next** to continue through the wizard until the wizard displays the `Status: Starting the WebSphere MQ service` message, followed by the recommended tasks page.

6. Optional: If you want to use the WebSphere MQ Postcard application to verify your installation, click the **Set up the Default Configuration** link to display, after a short delay, the WebSphere MQ Default Configuration window.

   ┌─ **Why am I doing this?** ─────────────────────────────────┐
   │ The default configuration creates objects, such as a queue manager called │
   │ QM_*machine_name*, that you can use to verify the success of your │
   │ installation by using the Postcard application that is supplied as part of │
   │ WebSphere MQ. If you do not create the default configuration, you can │
   │ still use the Postcard application later on, after you have created your │
   │ own objects in WebSphere MQ. │
   └──┘

   Note: You cannot create the default configuration if you have already created other queue managers; you must first delete the other queue managers then run the Default Configuration wizard.

a. The WebSphere MQ Default Configuration window shows an initial summary of the configuration you are about to create. This window remains on screen while you proceed through the pages of the Default Configuration wizard. Click the **Set up Default Configuration...** button to start the Default Configuration wizard.

b. Click **Next** to move through the information pages.

c. On the Select Options page, clear both the **Allow remote administration of the queue manager** and **Join the queue manager to the default cluster** check boxes, then click **Next**.

> **Why am I doing this?**
> This scenario assumes that you have one machine on which you are installing WebSphere Application Server and WebSphere MQ. You therefore do not need to administer the queue from another machine. Clustering is not described in this scenario.

d. On the summary page, click **Finish**. A dialog box is displayed while the default configuration is created.

> **What else might I do or be interested in?**
> You can use this window after installation to change the default configuration. To display this window after installation, use the Windows **Start** menu to start the Prepare WebSphere MQ wizard, then follow the steps.

The existing WebSphere MQ Default Configuration window is updated with the results of the configuration.

e. Click **Close** to close the Default Configuration wizard.

7. In the Prepare WebSphere MQ wizard, click **Next** to display the Completing the Prepare WebSphere MQ Wizard page.

8. Leave the **Launch WebSphere MQ Explorer** check box selected, choose whether to start Notepad to view the release notes, then click **Finish**. The WebSphere MQ Explorer starts.

9. If you created a default configuration, verify your installation by using the Postcard application that is supplied with WebSphere MQ.

a. If the Welcome page is not already displayed, click **Help** > **Welcome** to display it.

b. Click the **Get Started** icon.

c. Click the **Launch the Postcard Application** icon to display the Postcard - Sign On dialog box.

d. Click **Help** in this and other Postcard application dialog boxes to view instructions about running the Postcard application.

If you did not create a default configuration, you can still follow these steps to verify your installation, but you must first configure a queue manager as described in the next task of this scenario.

Note: Running the Postcard application on a non-default configuration automatically creates a queue called `postcard` on your queue manager. You can delete this queue after using the postcard application.

Results

You have installed WebSphere MQ and are ready to configure objects such as queues and queue managers.

If there were problems during installation, use the `.log` files in the `C:\Documents and Settings\`*userID*`\Local Settings\Temp` directory, and the `amqmsccw.txt` and `amqmjpse.txt` files in the installation directory, to investigate. The default installation directory is `C:\Program Files\IBM\WebSphere`.

What to do next

Follow the instructions in "Configuring the JNDI namespace and administered objects" to configure WebSphere MQ for use by the sample application.

Related information:

↪ Launchpad instructions

↪ Configuring WebSphere MQ with the Prepare WebSphere MQ Wizard

↪ Installing the WebSphere MQ Server

Configuring the JNDI namespace and administered objects

Define to WebSphere MQ an initial context for the JNDI namespace, then define in the namespace the administered objects that the sample application can use.

About this task

Before an application can retrieve administered objects from a JNDI namespace, an administrator must first create the administered objects. The administrator can use the WebSphere MQ JMS administration tool or WebSphere MQ Explorer to create and maintain administered objects in a JNDI namespace.

This scenario demonstrates the following aspects:

- Use of a JNDI namespace located in a local file system. A file system is used, because it is the simplest JNDI mechanism for a sample scenario.

> **What else might I do or be interested in?**
> The JNDI namespace can be on a file system, Lightweight Directory Access Protocol (LDAP) server, or on another JNDI implementation. If you want to use a JNDI namespace on an LDAP server or another JNDI implementation, you must configure the JNDI namespace and modify the sample application to reference the JNDI namespace, as required by the implementation.

- Use of the WebSphere MQ Explorer to create administered objects in the JNDI namespace. The JMS application can look up the administered objects to connect to WebSphere MQ and access MQ destinations with which to send or receive messages.

In this task, you create the following objects in WebSphere MQ:

WebSphere MQ

Queue manager
sampleQM

Queue Q1

JMS

Connection
factory
<queue manager
sampleQM>

Destination
<queue Q1>

**WebSphere MQ
JNDI namespace**

Connection
factory
<queue manager
sampleQM>

Destination
<queue Q1>
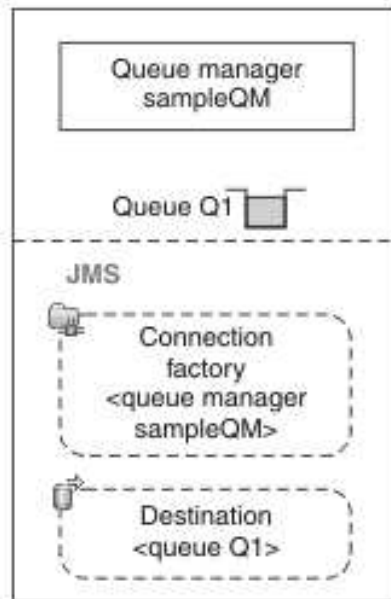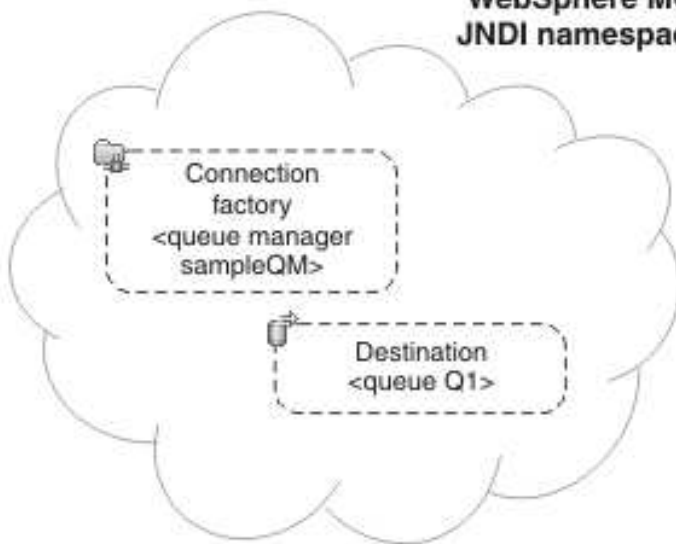
Legend

< ... > Configuration property

Figure 6. Objects created in WebSphere MQ

Procedure

1. Start WebSphere MQ Explorer, if it is not already started, by clicking **Start** >
   All Programs > **IBM WebSphere MQ** > **WebSphere MQ Explorer**. If

WebSphere MQ Explorer is running and displaying the Welcome page, close the Welcome page to start administering WebSphere MQ objects.

2. Create a queue manager to use for the sample application.

   a. Right-click **Queue Managers** and select **New** > **Queue Manager**. The Create Queue Manager wizard starts.

   b. In the **Queue manager name** field type sampleQM.

   What else might I do or be interested in?

   You can choose a different name for the queue manager, but you must remember to use it in later configuration steps, in place of sampleQM.

   Note: The name must have no more 48 characters, from the following set:
   - Uppercase or lowercase A-Z
   - Numerics 0-9
   - Period (.)
   - Forward slash (/)
   - Underscore (_)
   - Percent sign (%

   Names are case-sensitive. Objects of the same type must have different names. For example two queues cannot have the same name, but a queue manager and a queue can.

   ---

   c. In the **Dead-letter queue** field type SYSTEM.DEAD.LETTER.QUEUE. This field is the name of the dead-letter queue that is automatically created when you create the queue manager.

   Why am I doing this?

   A dead-letter queue stores messages that cannot be delivered to their correct destination, for example because the queue is full. All queue managers should have an associated dead-letter queue.

   ---

   d. Leave the other fields empty and click **Finish**, or if that button is disabled, click **Next**. The **Finish** button is disabled if the port number conflicts with an existing queue manager, for example the queue manager that is created as part of the default configuration. You must continue through the wizard to change the default port number.

   e. If you clicked **Next**, continue to accept the defaults and click **Next** on each page until you get to the final page of the wizard, when the **Finish** button becomes available. Change the specified port number, for example to 1415, and click **Finish**.

   WebSphere MQ displays a dialog window while the queue manager is created and started.

3. Add an initial context for the *JNDI namespace* then connect WebSphere MQ Explorer to that context

a. In the MQ Explorer - Navigator pane, right-click **JMS Administered Objects**, then select **Add Initial Context**. This action displays the "Connection details" page.

b. Under "Where is the JNDI namespace located?", select the **File system** check box.

c. In the **Bindings directory** field, type `C:\JNDI-Directory`.

> **Why am I doing this?**
>
> This value matches the JNDI namespace location specified in the sample JMS application. If you must specify a different JNDI directory, you must modify the application to match.

If the directory does not exist on your system the window displays the message `Specified location does not exist or is not readable`. Click **Browse** to open a file system window, navigate to `Local Disk (C:)`, then click **Make New Folder** to create the `JNDI-Directory` folder. Click **OK**.

Click **Next**.

d. On the User preferences page, leave the default settings.

> **Why am I doing this?**
>
> - **Context nickname**: The location of the JNDI namespace will be used as the nickname to display the initial context in WebSphere MQ Explorer.
> - **Connect immediately on finish**: This option connects WebSphere MQ Explorer to the JNDI namespace when you finish creating the initial context, so that you can create administered objects immediately.
> - **Automatically reconnect to context on startup**: This option is not selected, because usually you do not need WebSphere MQ Explorer to automatically reconnect to the initial context every time that you close and reopen WebSphere MQ Explorer.

> **What else might I do or be interested in?**
> If you routinely use WebSphere MQ Explorer to create or manage
> administered objects in the JNDI namespace, you can select the
> **Automatically reconnect to context on startup** check box to cause
> WebSphere MQ Explorer to automatically reconnect to the initial
> context whenever WebSphere MQ Explorer is started. This option
> saves you having to manually connect WebSphere MQ Explorer to the
> initial context.

Click **Finish** to create and display the initial context.

4. Create a connection factory administered object.

> **Why am I doing this?**
> A connection factory administered object defines a set of standard
> configuration properties for connections. An application uses a connection
> factory to create a connection to WebSphere MQ.

a. In the MQ Explorer - Navigator pane, expand **JMS Administered Objects**,
   then expand the initial context, labeled **file:/C:/JNDI-Directory/**.

b. Right-click **Connection Factories**, then select **New** > **Connection Factory**.
   This action displays the New Connection Factory wizard.

c. In the Name field, type `myCF`

> **Why am I doing this?**
> The sample JMS application contains code that looks up a connection
> factory with the name `myCF`. If you must use a different name, you
> must modify the application to match.WebSphere MQ is used for the
> messaging provider, because the sample application uses point-to-point
> messaging

Click **Next**.

d. Leave the type of connection factory as **Connection Factory**, because this
   option is the most flexible for general JMS use.

> **Why am I doing this?**
> A domain-independent connection factory enables JMS applications to
> use both point-to-point messaging and publish/subscribe messaging,
> especially if you want the JMS application to perform both types of
> messaging under the same transaction.

> **What else might I do or be interested in?**
> If a JMS application is intended to use only point-to-point messaging
> or only publish/subscribe messaging, you can select the specific
> messaging domain when you create the connection factory and a
> domain-specific (queue or topic) connection factory is created.

e. Leave the support for XA transactions as cleared.

Click **Next**.

f. Leave the transport as **Bindings**.

Click **Next**, then **Next** again.

g. On the Change properties page, select **Connection** from the menu on the
left, then in the Connection pane select sampleQM as the **Base queue
manager**.

h. Click **Finish**. WebSphere MQ displays a dialog window to show that the
object was created successfully. Click **OK** to close the dialog window.

5. Create a destination administered object.

a. In the MQ Explorer - Navigator pane, expand **JMS Administered Objects**,
then expand the initial context, labeled **file:/C:/JNDI-Directory/**.

b. Right-click **Destinations**, then select **New** > **Destination**. The New
Destination wizard is displayed.

c. In the **Name** field, type myQueue. Leave the **Type** as **Queue**.

> **Why am I doing this?**
> The sample JMS application contains code that looks up a destination
> with the name myQueue. The sample JMS application uses
> point-to-point messaging, so requires a destination of type queue.
> Destinations of type topic are used for publish/subscribe messaging.

d. Select the **Start wizard to create a matching MQ Queue** check box.

> **Why am I doing this?**
> The destination object needs a matching WebSphere MQ queue, and it
> is convenient to use WebSphere MQ Explorer to create both together.
> When you complete the New Destination wizard, the Create an MQ
> Queue wizard opens, with many of the destination details mapped to
> the WebSphere MQ queue.

Click **Next**.

Click **Next** again.

e. On the "Change properties" page, click **Select** next to **Queue manager**.
Select the sampleQM queue manager that you created earlier then click **OK**.

f. Specify Q1 as the name of the WebSphere MQ queue.

> **What else might I do or be interested in?**
> You can choose a different name for the queue, but you must
> remember to use it in later configuration steps, in place of Q1.
>
> **Note:** The name must have no more 48 characters, from the following
> set:
> • Uppercase or lowercase A-Z
> • Numerics 0-9
> • Period (.)
> • Forward slash (/)
> • Underscore (_)
> • Percent sign (%
>
> Names are case-sensitive. Objects of the same type must have different
> names. For example two queues cannot have the same name, but a
> queue manager and a queue can.

g. Click **Finish**. The Create an MQ Queue wizard starts.

If the wizard does not start, you might not have selected the **Start wizard to
create a matching MQ Queue** check box in an earlier step. In the MQ
Explorer - Navigator pane, expand the **sampleQM** queue manager,
right-click **Queues**, then select **New** > **Local Queue...**.

6. Create a matching WebSphere MQ Queue.

> **Why am I doing this?**
> The destination administered object created earlier represents a WebSphere MQ queue. This queue is where the JMS messages are stored.

  a. Click **Next** to accept the `sampleQM` queue manager that you specified earlier.

  b. Click **Next**.

  c. Click **Finish** to create the WebSphere MQ Queue using the information from the destination administered object that you created earlier. WebSphere MQ displays a dialog window with the message that the object was created successfully.

The new queue is now visible in the **Queues** section under the queue manager.

Results

You have now created the WebSphere MQ objects that are required to use the sample JMS application.

What to do next

Follow the steps in "Running the sample JMS application" to verify your configuration.

Related information:

➡ JMS connection factories

➡ Creating a destination

Running the sample JMS application

Run the sample stand-alone JMS application to send and receive messages through WebSphere MQ, and verify that you configured WebSphere MQ correctly for use with the sample application.

Before you begin

Download the sample application package. Click the following link and save the file: sampleJMSApp.zipsampleJMSApp.zip (in online information center), then extract the contents. The package contains a sample JMS application `.jar` file and batch files for running the application.

- The sample `sampleJMSApp.jar` file and the `.cmd` files must be in the same directory.
- The `.cmd` files use environment variables to set the class path for running the JMS application. When running the JMS application, if you see a Java `java.lang.NoClassDefFoundError`, you might need to adjust the class path line in the command file.

About this task

The JMS application comprises a requester client, which sends the initial message, and a responder client, which receives the message and sends a reply. The supplied batch files perform the following actions:

- `runresponder.cmd` opens a command prompt window in which the responder client starts then waits for a message.

- runrequester.cmd opens a separate command prompt window in which the requester client starts then sends a request message and receives a reply.

With two command prompt windows, you can see the actions of the requester and responder separately and more clearly.

Procedure

1. Double-click the runresponder.cmd file. In the command prompt window, labeled Responder window, the responder client starts then waits for a message.

```
> Connection factory located in JNDI.
> Destination located in JNDI.
> Creating connection to QueueManager.
> Created connection.

> Waiting for message.
```

2. Double-click the runrequester.cmd file. In the Requester window, observe the requester messages. In the Responder window, observe the updated responder messages; the message it received (from the requester client) and the reply message that it sent.

Results

In the command prompt window, labeled Requester window, the requester client shows the connection status, the message it sent, then the reply message that it received from the responder client:

```
> Connection factory located in JNDI.
> Destination located in JNDI.
> Creating connection to QueueManager.
> Connection created.

> Sending stock request for 'BakedBeans'
> Sent Message ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002b02

> Received Message ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002902 for 'B
akedBeans - 15 tins in stock'

> Closing connection to QueueManager.
> Closed Connection.
----------------------------------------------------------
In this window, observe the messages sent through WebSphere MQ:
- The request message sent
- The reply message received
-----
When ready, press any key to close this window
Press any key to continue . . .
```

In the Responder window, observe the updated responder messages; the message it received (from the requester client) and the reply message that it sent:

```
> Connection factory located in JNDI.
> Destination located in JNDI.
> Creating connection to QueueManager.
> Created connection.

> Waiting for message.

> Received Message ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002b02 for 'B
akedBeans'

> Sending Reply Message 'BakedBeans - 15 tins in stock'
> Sent Message ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002902
```

```
> Closing connection to QueueManager.
> Closed connection.
----------------------------------------------------------
In this window, observe the updated responder messages
- The request message received (from the requester)
- The reply message sent
-----
When ready, press any key to close this window
Press any key to continue . . .
```

The messages shown in the two command windows verify that the requester and responder clients of the sample application can communicate with each other through WebSphere MQ.

What to do next

Follow the instructions in "Installing WebSphere Application Server" to begin the WebSphere Application Server part of the scenario.

Installing WebSphere Application Server

Install WebSphere Application Server by using the graphical user interfaces. At the end of this task, you are ready to configure WebSphere Application Server to communicate with WebSphere MQ.

About this task

The installation process varies depending on which version of WebSphere Application Server you want to install. Starting from Version 8, you install WebSphere Application Server by using IBM Installation Manager. IBM Installation Manager is a product that you can use to install, uninstall, update, and modify IBM software packages. It helps you to manage IBM software more easily.

Installing WebSphere Application Server Version 6.1 or 7

Install WebSphere Application Server Version 6.1 or 7 and apply the latest fix pack, by using the graphical user interfaces. At the end of this task, you are ready to configure WebSphere Application Server to communicate with WebSphere MQ.

Before you begin

- You must have local administrator authority when you are installing. Define this authority through the Windows operating system facilities. The administrator ID must not contain spaces.
- Ensure that you have sufficient disk space, up to 2 GB for all components.
- Ensure that prerequisites and corequisites are at the required release levels. On the List of supported software for WebSphere Application Server V7.0 page, click the link for your operating system to see information about supported operating systems, browsers, and hardware requirements.
- Download the IBM Update Installer for WebSphere Software package. You use this program to install updates to WebSphere Application Server. See Recommended Updates for WebSphere Application Server, and look under your release of WebSphere Application Server for a link to the Update Installer download page.
- Download the latest WebSphere Application Server fix pack *.pak file from IBM Support site for WebSphere Application Server. This scenario was tested with Fix Pack 13.

Note: If your browser changes the download file extension from `.pak` to `.zip`, change the extension back to `.pak`.

Note: If you are installing WebSphere Application Server Version 8, complete the instructions in Installing the product in the WebSphere Application Server product documentation.

For more information about installation requirements, see IBM Support pages.

About this task

This task describes the basic steps to install WebSphere Application Server to the latest maintenance level on the Windows XP Professional (SP2 or later) operating system.

Procedure

1. Insert the disk that is labeled WebSphere Application Server Network Deployment into the disk drive, or access the root directory of the downloaded Passport Advantage image on your hard disk.
2. Start the installation process by double-clicking the Launchpad icon.
3. On the Welcome page, click **Launch the installation wizard for WebSphere Application Server Network Deployment**. After a short delay, the installation wizard starts.
4. Follow the instructions in the wizard to move through the pages, choosing whether to install sample applications and language packs, and specifying an installation directory, until you reach the WebSphere Application Server environments page.

   Note: Although you can install the sample applications and language packs later, install them now if you might use them in the future. These features can be corrupted if you install them after applying maintenance.
5. On the WebSphere Application Server environments page, select the **Application server** option and click **Next**.

   > **Why am I doing this?**
   > This option creates the simplest profile, for a stand-alone application server.

6. On the next page, choose whether to enable administrative security. If you accept the default option, to enable security, record the user name and password that you choose; without this information you are not able to use the product.

   > **Why am I doing this?**
   > Administrative security activates the settings, such as the authentication of users, the use of Secure Sockets Layer (SSL), and the choice of user account repository, that protects your server from unauthorized users. Some environments, such as development systems, might not require security. However, in most environments enable administrative security to prevent unauthorized users from accessing the administrative console.

7. On the installation summary page, verify your choices and click **Next** to install the product code. The installation wizard creates the uninstallation program and then displays a progress page that shows which components are being installed. This process might take several minutes.

8. The Installation results page is displayed. Examine the information about the page to check that there were no file system or other unusual errors while installing. If there were problems during installation, use the `logs/install/log.txt` file in the installation directory to investigate.

9. Clear the **Launch the First steps console** check box and click **Finish** to close the installation wizard. Close the launchpad. You are now ready to install the fix pack.

    Note: If WebSphere Application Server is running, you cannot install the fix pack. If you must leave the task and continue later, make sure that WebSphere Application Server is stopped before you continue. For example, if you restart the machine WebSphere Application Server starts automatically. To stop WebSphere Application Server, click **Start** > **All Programs** > **IBM WebSphere** > **Application Server Network Deployment v7.0** > **Profiles** > **AppSrv01** > **Stop the server**.

10. Unpackage the IBM Update Installer for WebSphere Software file that you downloaded earlier into a temporary directory, and double-click the `UpdateInstaller\install.exe` file to start the Update Installer installation wizard.

11. Follow the instructions in the wizard to install the software, selecting the **Launch Update Installer for WebSphere Software on exit** option on the final page, which is displayed after installation completes. The Update Installer starts automatically when you click **Finish** to exit the wizard.

12. Follow the instructions in the Update Installer to move through the wizard until you reach the Maintenance Package Directory Selection page.

13. Click **Browse...**, select the directory where you stored the downloaded fix pack `*.pak` file, and click **Open**.

    Note: In this step you specify only the directory containing the fix pack file, not the fix pack file itself.

14. On the next page, Available Maintenance Package to Install, the fix pack `*.pak` file is already selected. Click **Next** to view the summary page, and **Next** again to install the fix pack.

Results

WebSphere Application Server is installed with the latest level of maintenance. If you encountered a problem when installing, see the Troubleshooting installation topic in the WebSphere Application Server information center, to resolve the issue.

What to do next

Follow the instructions in "Configuring WebSphere Application Server" on page 33 to configure WebSphere Application Server to communicate with WebSphere MQ.

Related information:

➡ Installing the Update Installer for WebSphere Software

➡ Installing WebSphere Application Server

Installing WebSphere Application Server Version 8 or later

Install WebSphere Application Server 8 or later by using Installation Manager. At the end of this task, you are ready to configure WebSphere Application Server to communicate with WebSphere MQ.

Before you begin

- You must have local administrator authority when you are installing. Define this authority through the Windows operating system facilities. The administrator ID must not contain spaces.
- Ensure that you have sufficient disk space. You require up to 2 GB for all components.
- Ensure that prerequisites and corequisites are at the required release levels. On the WebSphere Application Server detailed system requirements page, click the link for your operating system and the link for software to see information about supported operating systems and browsers, and hardware requirements.

About this task

This task describes the basic steps to install WebSphere Application Server 8.5 on the Windows 7 (SP1) operating system by using Installation Manager.

Procedure

1. If you haven't installed IBM Installation Manager before, install it.
   a. Extract the Installation Manager archive file.
   b. Switch to Installation Manager archive directory and double-click the `install.exe` file.
   c. In the Install Packages window, select the version number, for example **Version 1.5.2** , and then click **Next**.
   d. Click **I accept the terms in the license agreement**, and then click **Next.**
   e. In **Installation Manager Directory** field, type the file path. The file path is the location where you want to install IBM Installation Manager.
   f. Click **Next**, and then click **Install**. A progress bar is displayed, which shows that the installation is in progress.

   When the installation is complete, a window is displayed, indicating that IBM Installation Manager is installed.
2. Click **Restart Installation Manager** in the lower right of the window. Alternatively, you can click **Start** > **All Programs** > **IBM Installation Manager** > **IBM Installation Manager** to start Installation Manager.
3. Click **Install**.
4. If a message is displayed that states `Installation Manager cannot find any packages to install`, add WebSphere Application Server to the repository.
   a. Click the **Repositories** link in the **Repository connections** field.
   b. Click **Add Repository** in the upper right of the Preference window.
   c. Click **Browse** and find the `repository.config` file for WebSphere Application Server.
   d. Click **OK**.
   e. Click **Test Connections** on the right of the Preference window. If a message is displayed that states `All the selected repositories are connected`, WebSphere Application Server is added to the repository.

f. Click **OK** to close the window and click **OK** again to confirm adding the repository.

5. Select the version number of WebSphere Application Server, for example **Version 8.5.0.0**, and then click **Next**.

6. Click **I accept the terms in the license agreement**, and then click **Next.**

7. Click **Next**, then click **Next** again.

8. Select the languages that you want and click **Next**.

9. Select the components that you want and click **Next**.

10. Click **Install**. A progress bar is displayed, which shows that the installation is in progress.

11. When the installation is complete, a window is displayed, which indicates that the packages are installed. Select the programs that you want to start and click **Finish**. WebSphere Application Server is successfully installed.

12. Create an application server profile

　　　　a. Start Profile Management Tool. For example, click **Start** > **All Programs** > **IBM WebSphere** > **IBM WebSphere Application Server V8.5** > **Tools** > **Profile Management Tool**.

　　　　b. Click **Create**.

　　　　c. In Environment Selection window, expand **WebSphere Application Server**, and click **Application server**, and then click **Next**.

　　　　d. In Profile Creation Options window, leave the option as **Typical profile creation**, and then click **Next**.

　　　　e. Choose whether to enable administrative security and click **Next**. If you choose to enable it, every time you use WebSphere Application Server, you must prove your identity.

　　　　f. Confirm your settings and click **Create**.

　　　　g. When the profile creation is complete, the Profile Creation Complete window is displayed. Clear **Launch the First steps console** and click **Finish**. An application server profile is successfully created.

Results

WebSphere Application Server is installed. If you encountered a problem when installing, see the IBM Installation Manager Information Center. For further information about where to find this information center, see About IBM Installation Manager Information Center.

What to do next

Follow the instructions in "Configuring WebSphere Application Server" to configure WebSphere Application Server to communicate with WebSphere MQ.

Configuring WebSphere Application Server

Create an activation specification, connection factory, and queue for WebSphere Application Server applications to use to communicate with WebSphere MQ.

About this task

To use JMS messaging between WebSphere Application Server and WebSphere MQ, you create the following objects in WebSphere Application Server.

- Activation specification

- Connection factory
- Queue (a type of JMS destination)

These objects are described in "Overview: The delivered logical topology" on page 10.

The following image shows these objects in WebSphere Application Server, and the corresponding set of objects in WebSphere MQ that you created when you configured WebSphere MQ for use by the sample application. When you create one of these objects in WebSphere Application Server, it is stored in the WebSphere Application Server JNDI namespace.
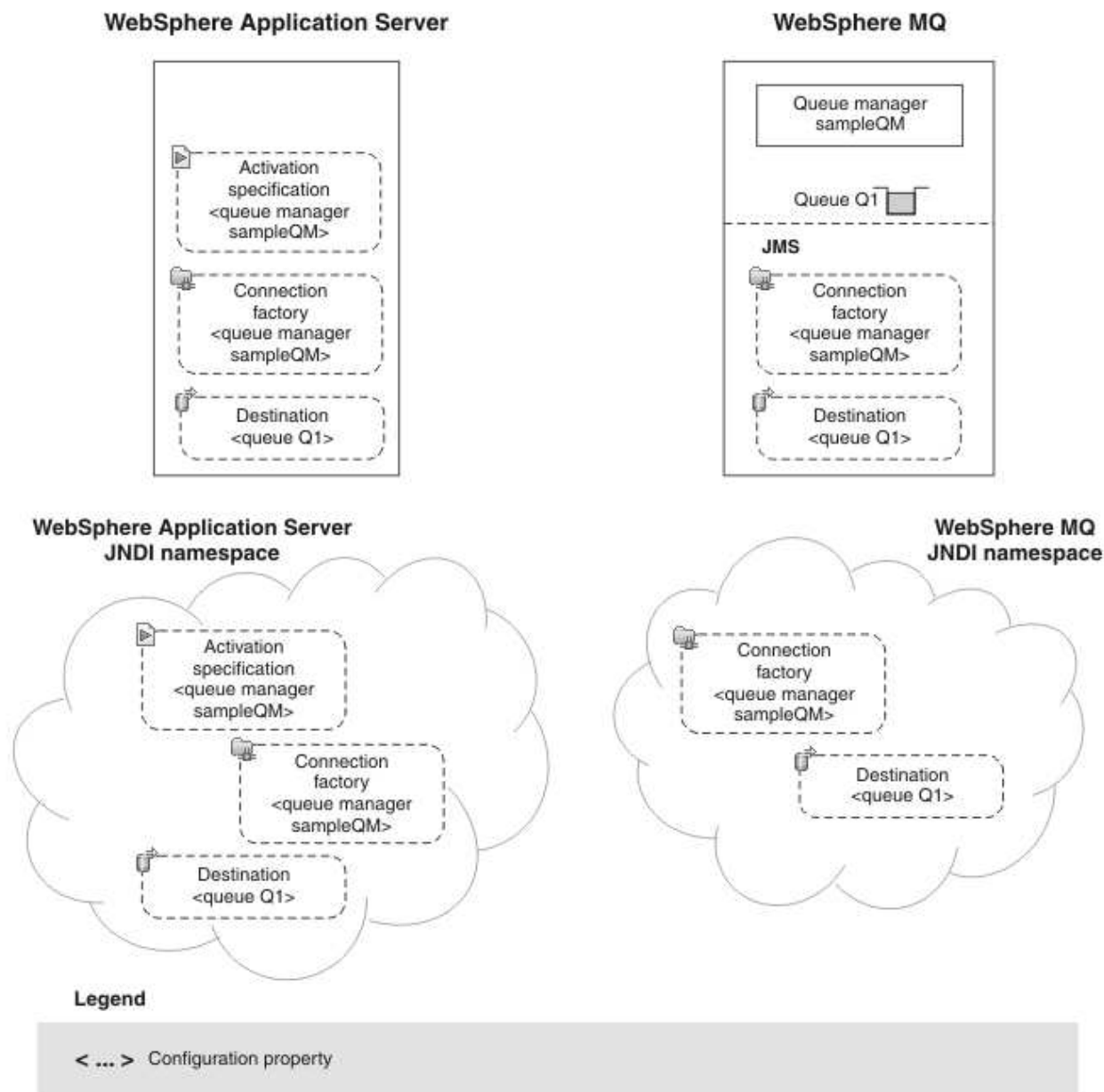
Figure 7. Objects created in WebSphere Application Server, and the corresponding objects in WebSphere MQ

Procedure

1. If WebSphere Application Server is not already started, click **Start** > **All Programs** > **IBM WebSphere** > **Application Server Network Deployment v7.0** > **Profiles** > **AppSrv01** > **Start the server** for example. A command window is displayed until the server is started; this process might take a few minutes.

2. If the administrative console is not already started, click **Start** > **All Programs** > **IBM WebSphere** > **Application Server Network Deployment v7.0** > **Profiles** > **AppSrv01** > **Administrative console** for example.

   Note: You must log on even if you did not enable administrative security when you installed the product. If you did not enable administrative security, the user ID is used only to distinguish changes that you make in the administrative console from changes made by other users.

3. Create an activation specification.

   > **Why am I doing this?**
   > Activation specifications are part of the JCA 1.5 specification. The MDB application uses the activation specification to connect to a WebSphere MQ queue manager for the processing of inbound messages. The activation specification also provides other options, such as security settings.

   > **What else might I do or be interested in?**
   > You can use listener ports instead of activation specifications, however this option is an earlier method which is based on an optional, and now deprecated, part of the JMS specification. Listener ports are stabilized in this version of WebSphere Application Server. For more information, see Message-driven beans, activation specifications, and listener ports and Stabilized features.

   a. In the administrative console, click **Resources** > **JMS** > **Activation specifications**.

   b. Ensure that the scope is set to **Cell=***cell_name*, then click **New**.

   > **Why am I doing this?**
   > You use the **Scope** property to set the level at which the activation specification is visible. The cell scope is the highest level, giving the activation specification the greatest visibility.

   > **What else might I do or be interested in?**
   > You can also create activation specifications at other levels. For example, if you have multiple servers you might create an activation specification for each server, using the server scope, so that you can specify different settings to be used for each server.

   c. Select **WebSphere MQ messaging provider** then click **OK**. The "Create WebSphere MQ JMS resource" wizard starts.

d. On the "Configure basic attributes" page, add a **Name** of your choice, then type jms/PiShopReplyMDB in the **JNDI name** field. Click **Next**.

> **Why am I doing this?**
> The name you choose is used to display the activation specification in the administrative console. The JNDI name is used to reference the object in the JNDI namespace.

e. On the "Specify MDB destination data" page type jms/wasQueue in the **Destination JNDI name** field. The destination type is Queue by default. Click **Next**.

> **Why am I doing this?**
> The sample application uses point-to-point messaging, which is achieved by using a queue destination. Topic destinations are used for publish/subscribe messaging. The JNDI name is used to reference the queue in the JNDI namespace. You use the same JNDI name later on, when you create the queue object.

f. On the "Select connection method" page click **Next**, leaving the default option, **Enter all the required information into this wizard**, selected.

g. On the "Supply queue connection details" page type the name of the WebSphere MQ queue manager, sampleQM in the **Queue manager or queue sharing group name** field, then click **Next**.

h. On the "Enter connection details" page, enter the following information then click **Next**.

Transport mode
Leave as the default setting, **Bindings, then client**.

> **Why am I doing this?**
> In this mode, WebSphere Application Server first attempts to connect in bindings mode, where the WebSphere MQ messaging provider within WebSphere Application Server connects directly to the WebSphere MQ queue manager. If this connection attempt fails, the WebSphere MQ messaging provider then attempts to connect by using client mode, which uses a TCP/IP-based network. Bindings mode usually has better performance, but can be used only when WebSphere Application Server and WebSphere MQ are installed on the same machine. The host name, port, and server connection channel information is required for client mode. Bindings mode requires the configuration of native library path information, which you will add in a later step in this task.

Host name
The name of your machine.

Port The port number of the WebSphere MQ queue manager sampleQM.

> This port number is displayed in WebSphere MQ Explorer under **Queue managers** > *sampleQM* > **Advanced** > **Listeners**, in the Port column in the LISTENER.TCP row.

Server connection channel
> The name of the WebSphere MQ channel to use for communication.
>
> If you do not have a specific channel that you want to use, specify the default channel that is created when you created the queue manager, SYSTEM.DEF.SVRCONN.
>
> **Tip:** To display the default channel in WebSphere MQ Explorer, click the Show System Objects icon at the upper right of the MQ Explorer - Content pane.

  i. On the "Test connection" page, click **Test connection**.

> ┌─ **Why am I doing this?** ─────────────────────────────────
> │ The **Test connection** button verifies that the activation specification contains the necessary details to create a connection to WebSphere MQ. To use this connection, you must create further administrative objects, as described in the rest of the steps.
> └──

> If successful, the wizard displays the message `A connection was successfully made to WebSphere MQ`. If not successful, check that the queue manager name, port, and server channel that you specified are correct, and match the corresponding values in WebSphere MQ. For example, the following error indicates that the channel name was not found on the queue manager that you specified.

```
A connection could not be made to WebSphere MQ for the following reason: JMSCMQ0001: WebSphere MQ call failed with compc
```

> If the following error message is displayed, open WebSphere MQ Explorer and expand **IBM WebSphere MQ** > **Queue Managers**, then right click **sampleQM**, and then click **Properties** > **Communication**. Select **Disabled** in **Channel authentication records** list.

```
A connection could not be made to WebSphere MQ for the following reason: JMSCMQ0001: WebSphere MQ call failed with compc
```

  j. Click **Next**, then on the Summary page click **Finish**.

4. Create a connection factory.

> ┌─ **Why am I doing this?** ─────────────────────────────────
> │ A JMS connection factory contains information about how to create a connection. When an application needs a JMS connection, the factory creates a connection instance. The connection factory requires the same connection information as the activation specification that you created earlier, but is used for outbound messages from the MDB, whereas the activation specification is used for inbound messages.
> └──

> **What else might I do or be interested in?**
> With previous versions of the JMS specification, you had to create either
> a queue connection factory or a topic connection factory, depending on
> your required messaging pattern (point-to-point or publish/subscribe).
> With the *common interfaces* of the JMS 1.1 specification, you can now
> create a generic connection factory, which creates connections to both
> types of messaging destination. You can therefore use both messaging
> models in the same session.

a. In the administrative console, click **Resources** > **JMS** > **Connection
   factories**.
b. Ensure that the scope is set to **Cell=***cell_name*, then click **New**.
c. Select **WebSphere MQ messaging provider**, then click **OK**. The "Create
   WebSphere MQ JMS resource" wizard starts.
d. On the "Configure basic attributes" page enter a **Name** of your choice,
   enter jms/replyCF for the **JNDI name**, then click **Next**.

> **Why am I doing this?**
> The name you choose is used to display the connection factory in the
> administrative console. The JNDI name is used to reference the object
> in the JNDI namespace. The JNDI name is also used in the sample
> MDB application code; if you want to use a different JNDI name you
> must change the MDB to match, then redeploy the application.

e. On the "Select connection method" page click **Next**, leaving the default
   option, **Enter all the required information into this wizard**, selected.
f. On the "Supply queue connection details" page, type sampleQM in the
   Queue manager or queue sharing group name field then click **Next**.
g. On the "Enter connection details" page, enter the following details then
   click **Next**.

Transport mode
> Leave as the default setting, **Bindings, then client**.

> **Why am I doing this?**
> In this mode, WebSphere Application Server first attempts to
> connect in bindings mode, where the WebSphere MQ
> messaging provider within WebSphere Application Server
> connects directly to the WebSphere MQ queue manager. If
> this connection attempt fails, the WebSphere MQ messaging
> provider then attempts to connect by using client mode,
> which uses a TCP/IP-based network. Bindings mode usually
> has better performance, but can be used only when
> WebSphere Application Server and WebSphere MQ are
> installed on the same machine. The host name, port, and
> server connection channel information is required for client
> mode. Bindings mode requires the configuration of native
> library path information, which you will add in a later step
> in this task.

Host name
The name of your machine.

Port The port number of the WebSphere MQ queue manager
sampleQM.

This port number is displayed in WebSphere MQ Explorer under
Queue managers > *sampleQM* > **Advanced** > **Listeners**, in the
Port column in the LISTENER.TCP row.

Server connection channel
The name of the WebSphere MQ channel to use for
communication.

If you do not have a specific channel that you want to use, specify
the default channel that is created when you created the queue
manager, SYSTEM.DEF.SVRCONN.

Tip: To display the default channel in WebSphere MQ Explorer,
click the Show System Objects icon at the upper right of the MQ
Explorer - Content pane.

h. On the "Test connection" page, click **Test connection**.

> **Why am I doing this?**
> The **Test connection** button verifies that the connection factory
> contains the necessary details to create a connection to WebSphere
> MQ. To use this connection, you must create a queue, as described in
> the rest of the steps.

If successful, the wizard displays the message A connection was
successfully made to WebSphere MQ. If not successful, check that the
queue manager name, port, and server channel that you specified are
correct, then match the values in WebSphere MQ.

i. Click **Next**, then on the Summary page click **Finish**.

5. Create a queue.

> **Why am I doing this?**
> A queue is a type of JMS destination, used for point-to-point messaging.
> A destination identifies the messaging target or source.

> **What else might I do or be interested in?**
> In a publish/subscribe scenario, you create a topic instead of a queue.

a. In the administrative console, click **Resources** > **JMS** > **Queues**.

b. Ensure that the scope is set to **Cell=***cell_name*, then click **New**.

c. Select **WebSphere MQ messaging provider** then click **OK**.

d. Under General Properties, enter the following values for the required
fields, then click **OK**.

Name A name of your choice.

This name is used to display the queue in the administrative
console.

JNDI name

> jms/wasQueue
>
> The JNDI name is used to reference this queue in the WebSphere Application Server JNDI namespace. The value matches the destination name that you specified when you created the activation specification.

Queue name

> Q1
>
> This value is the administrative name of the queue in WebSphere MQ that you want to communicate with. This is different from the JNDI name of the queue.

> ---
> **What else might I do or be interested in?**
>
> Specify a **Queue manager or Queue sharing group name** if you want WebSphere Application Server to send messages to a queue that is on a different queue manager from the one connected to by the activation specification or connection factory. The queue managers must be configured to communicate with each other, as described in the What is intercommunication? topic in the WebSphere MQ information center.

6. Configure the WebSphere MQ messaging provider with native libraries information.

> ---
> **Why am I doing this?**
>
> When you created the activation specification and connection factory you specified a transport mode of **Bindings, then client**. To use bindings mode, the WebSphere MQ messaging provider within WebSphere Application Server requires native libraries, for example `.dll` or `.so` files, from WebSphere MQ. You must configure the messaging provider by adding the path to these libraries.

   a. In the administrative console, click **Resources** > **JMS** > **JMS providers**.

   b. Click the **WebSphere MQ messaging provider** whose scope begins **Cell=**.

   c. In the **Native library path** field under General Properties, type the full path of the directory that contains the WebSphere MQ native libraries, then click **OK**. For a default installation, the path is `C:\Program Files\IBM\WebSphere MQ\java\lib` for Windows 32-bit operating system or `C:\Program Files (x86)\IBM\WebSphere MQ\java\lib64` for Windows 64-bit operating system.

7. Click the **Save** link in the Messages section at the top of the pane to save your changes to the master configuration.

8. Stop the application server. Click **Start** > **All Programs** > **IBM WebSphere** > **Application Server Network Deployment v7.0** > **Profiles** > **AppSrv01** > **Stop the server** for example, then wait for the command window to disappear.

9. Open a command window in the `bin` directory of the application server profile, then enter `osgiCfgInit.bat`. By default, the location of the `bin` directory is `C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\bin` for Windows 32-bit operating system or `C:\Program Files (x86)\IBM\WebSphere\AppServer\profiles\AppSrv01\bin` for Windows 64-bit

operating system. The `osgiCfgInit.bat` script clears a cache. If you do not run this script you must restart the application server twice for the changes to take effect.

10. Start the application server. Click **Start** > **All Programs** > **IBM WebSphere** > **Application Server Network Deployment v7.0** > **Profiles** > **AppSrv01** > **Start the server** for example.

Results

You have configured WebSphere Application Server with the necessary objects to enable communication with WebSphere MQ.

What to do next

Follow the instructions in "Deploying the sample application to WebSphere Application Server" to deploy the sample MDB application.

Deploying the sample application to WebSphere Application Server

After you deploy the sample message driven bean (MDB) application, you can use it to verify that WebSphere Application Server is communicating with WebSphere MQ.

Before you begin

Download the sample application. Click the following link and save the file to the machine that hosts WebSphere Application Server: sampleMDB.earsampleMDB.ear (in online information center).

About this task

The sample MDB application, `sampleMDB.ear`, is designed to use the objects that you created earlier in WebSphere Application Server. The MDB application uses these objects to send a message to WebSphere MQ, for receipt by the sample JMS application requester client that you used in "Running the sample JMS application" on page 27.

The following diagram shows a message traveling from the sample JMS client to WebSphere MQ, and then on to WebSphere Application Server, where it is passed to the MDB running within WebSphere Application Server. A response message travels from the MDB to WebSphere MQ, and then on to the JMS client.
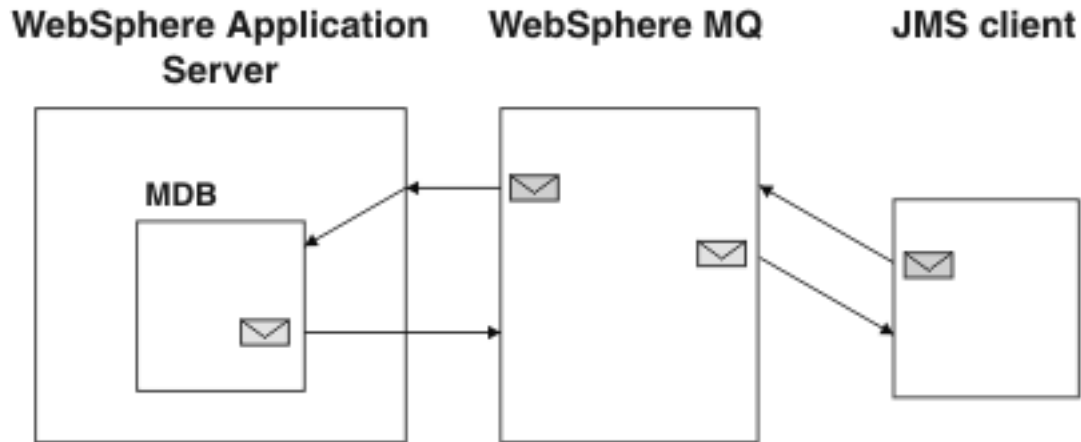
Figure 8. A message traveling from the sample JMS client to WebSphere MQ, then onward to the MDB on WebSphere Application Server

Procedure

1. If WebSphere Application Server is not already started, click **Start** > **All Programs** > **IBM WebSphere** > **Application Server Network Deployment v7.0** > **Profiles** > **AppSrv01** > **Start the server** for example.

2. Click **Start** > **All Programs** > **IBM WebSphere** > **Application Server Network Deployment v7.0** > **Profiles** > **AppSrv01** > **Administrative console** for example to start the administrative console.

3. Click **Applications** > **Application Types** > **WebSphere enterprise applications**. This presents the table of installed WebSphere enterprise applications, and provides a set of buttons for you to act on applications.

> **What else might I do or be interested in?**
> Instead of clicking **Applications** > **Application Types** > **WebSphere enterprise applications** (and next clicking **Install**, you could have clicked **Applications** > **New Application** > **New Enterprise Application**. Using the suggested route described means that when you have finished installing the new application then you are returned to the table of applications.

4. In the Enterprise Applications pane, click **Install**.

5. On the first Preparing for the application installation page, select **Local file system**, then browse to the folder where you downloaded the sample MDB application, sampleMDB.ear.

   a. Select the application then click **Open**.

   b. Leave all other fields unchanged and click **Next**.

6. For "How do you want to install the application?", leave the default option, **Fast path** selected.

> **Why am I doing this?**
>
> By choosing the "fast path" option, you can skip many of the steps on the Install New Application page; you are prompted only when additional information is required. For more information, see the topic Preparing for application installation binding settings in the WebSphere Application Server information center.

Click **Next**.

7. On the Install New Application page, consider whether you want to give a new administrative name to the application.

> **Why am I doing this?**
>
> Unless you specify otherwise, the name of the installed application is derived from the name of the MDB EAR file. For example, in this scenario the default application name "sampleMDB" comes from the EAR file name *sampleMDB*.ear. If you prefer, you can give the application a different name (which in this scenario you would have to use wherever action on the "sampleMDB" application is described). To do this, edit the **Application name** field to specify your chosen name.

8. Click the **Step 3** link to go to the Bind listeners for message-driven beans page. In this step, you specify information needed to bind the MDB to an activation specification in the application server.

> **Why am I doing this?**
>
> Binding information is used to enable an application to find the enterprise beans or resources that it needs to work with in the application server. In this scenario, the MDB must be bound to the activation specification and the JMS queue that you configured in "Configuring WebSphere Application Server" on page 33.

> **What else might I do or be interested in?**
>
> To save having to specify the binding information whenever you deploy the MDB EAR file, you can preset the binding information for the EAR file in other ways, such as using an assembly tool. For more information, see the topic Preparing for application installation binding settings in the WebSphere Application Server information center.

On the Bind listeners for message-driven beans page specify the JNDI name of the activation specification to which the sample MDB application must be bound.

a. Under **Listener Bindings**, select **Activation Specification**.

   Tip: You do not need to select the EJB module, `pishopExampleReplyMDB`, because it is the only EJB module in the EAR file.

b. In the Target Resource JNDI Name field, type the JNDI name, `jms/PiShopReplyMDB`.

The value must match the JNDI name that you specified in "Configuring WebSphere Application Server" on page 33.

You do not need to set other properties, which are provided to allow you to override values configured on the activation specification. For example, in this scenario the MDB uses the JMS destination, jms/wasQueue, that you configured on the activation specification in "Configuring WebSphere Application Server" on page 33. (The optional Destination JNDI name field overrides the destination configured on the activation specification.)

   c. Click **Next**.

9. On the Map resource references to resources page, map the connection factory resource reference defined in the sample MDB application to the JNDI name of the connection factory.

> **Why am I doing this?**
> Each resource reference that is defined in an application must be mapped to an actual resource that has been configured for use by the application. A resource reference allows an application to access a resource by using a logical name rather than the actual name that is defined for the resource in the runtime environment. Resource references eliminate the need to change your application code when you change resource runtime configurations.

   a. Under **Target Resource JNDI Name**, click **Browse** to display a list of resources that are available to be used to bind the resource reference of the MDB.

      Tip: For this scenario, the resource that you need to select is the connection factory PiShopReplyCF that you specified in "Configuring WebSphere Application Server" on page 33.

   b. Select PiShopReplyCF then click **Apply**.

   c. Leave all other fields unchanged and click **Next**.

10. On the Summary page, click **Finish**. The program displays messages to indicate that the application is being installed.

11. Click **Finish**. The program displays messages to indicate that the application is being installed. If the application does not install successfully, examine the messages to identify why the installation failed. Correct the problems as needed and try installing the application again.

12. When the application is installed successfully, click the **Save** link that is displayed at the end of the messages, to save your changes to the master configuration.

Results

You have deployed the MDB application, which is now visible in the Enterprise Applications pane.

What to do next

Follow the instructions in "Verifying the solution" on page 45 to run the sample applications and verify that messages can be passed between the two products.

Verifying the solution

Run the sample JMS and message drive bean (MDB) applications to verify that they can communicate with each other.

Before you begin

You must have configured WebSphere Application Server and WebSphere MQ as described in "Configuring WebSphere Application Server" on page 33 and "Configuring the JNDI namespace and administered objects" on page 20.

About this task

In "Running the sample JMS application" on page 27, you ran the supplied JMS sample application to verify that the requester and responder clients of the application could communicate through WebSphere MQ. In "Deploying the sample application to WebSphere Application Server" on page 41, you installed an MDB application. In this task, you run the requester application as before, but the reply comes from the MDB application instead of the previous responder application, verifying that messages are being passed between WebSphere MQ and WebSphere Application Server.

WebSphere Application Server

MDB

Activation specification
<queue manager sampleQM>

Connection factory
<queue manager smapleQM>

Connection factory
<queue manager sampleQM>

Activation specification
<queue manager smapleQM>

Destination
<queue Q1>

Destination
<queue Q1>

WebSphere MQ

Queue manager sampleQM

Queue Q1

JMS

Connection factory
<queue manager sampleQM>

Destination
<queue Q1>

JMS Client

Q1

Connection factory
<queue manager sampleQM>

Destination
<queue Q1>

WebSphere Application Server JNDI namespace

Activation specification
<queue manager sampleQM>

Connection factory
<queue manager sampleQM>

Destination
<queue Q1>

WebSphere MQ JNDI namespace

Connection factory
<queue manager sampleQM>

Destination
<queue Q1>

Legend

< ... > Configuration property

Procedure

1. If WebSphere Application Server is not already started, click **Start** > **All Programs** > **IBM WebSphere** > **Application Server Network Deployment v7.0** > **Profiles** > **AppSrv01** > **Start the server** for example.

2. Click **Start** > **All Programs** > **IBM WebSphere** > **Application Server Network Deployment v7.0** > **Profiles** > **AppSrv01** > **Administrative console** for example, then log in to the administrative console.

3. Click **Applications** > **Application Types** > **WebSphere enterprise applications** to display a list of installed applications.

4. Select the sampleMDB application and click **Start**. The application status changes to a green arrow. The MDB application is now waiting to receive a message.

   If you see an error message similar to the following message, there might be a mismatch between your WebSphere Application Server configuration and the

information contained in the sample application. Check the WebSphere Application Server error logs in *install_directory*\profiles\AppSrv01\logs\ server1 for more information.

```
ErrorsampleMDB failed to start. Check the logs for server server1 on node mymachineNode01 for details.
ErrorAn error occurred while starting sampleMDB. Check the logs for server server1 on node mymachineNode01 for more info
```

5. Double-click the runrequester.cmd batch file in the sample package. This command file opens a Command Prompt window then runs the sample JMS requester client, which sends a message to WebSphere MQ. The runrequester.cmd file uses environment variables to set the class path for running the JMS application. If you see a Java java.lang.NoClassDefFoundError, you might need to adjust the class path line in the command file.

Results

The sample requester application sends the same request message as before, but the response now comes from the MDB application. The following output, showing sent and received messages, is displayed in the requester application Command Prompt window:

```
> Connection factory located in JNDI.
> Destination located in JNDI.
> Creating connection to QueueManager.
> Connection created.

> Sending stock request for 'BakedBeans'
> Sent Message ID=ID:414d5120514d5f574153494e5354414c423fe14c20010602

> Received Message ID=ID:414d5120514d5f574153494e5354414c423fe14c2000f205 for 'B
akedBeans - 15 tins in stock. MDB has connected to business session bean to find
 expected delivery time: 1 day'

> Closing connection to QueueManager.
> Closed Connection.
```

If you see errors similar to the following error in the WebSphere Application Server logs, check that the name of the queue in WebSphere Application Server matches the administrative name (not the JNDI name) of the queue in WebSphere MQ:

```
WebSphere MQ call failed with compcode '2' ('MQCC_FAILED') reason '2085' ('MQRC_UNKNOWN_OBJECT_NAME')

com.ibm.msg.client.jms.DetailedInvalidDestinationException: JMSWMQ2008: Failed to open MQ queue 'myQueue'. JMS attempted
```

Note: If you update the WebSphere Application Server configuration you must restart the application server for the changes to take effect.

You have finished implementing the solution, and verified that WebSphere Application Server can connect to WebSphere MQ for the successful transmission of messages.

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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
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 106-0032, 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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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.

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

Programming interface information

Programming interface information, if provided, is intended to help you create application software for use with this program.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Important: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol ($^{\circledR}$ or $^{™}$), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at Copyright and trademark information (www.ibm.com/legal/copytrade.shtml).

Sending your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

Use one of the following methods to send us your comments:
- Send an email to ibmkc@us.ibm.com
- Use the form on the web here: www.ibm.com/software/data/rcf/

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

Include the following information:
- Your name and address
- Your email address
- Your telephone or fax number
- The publication title and order number
- The topic and page number related to your comment
- The text of your comment

IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you submit.

Thank you for your participation.

IBM ®

Printed in USA