

IBM Product Connectivity Scenarios information center
Version 1 Release 0

*Using a WebSphere MQ multi-instance
queue manager for high availability
with WebSphere Application Server*

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 57.

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

© **Copyright IBM Corporation 2011, 2019.**

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

Contents

Figures	v	Installing WebSphere MQ.	21
Chapter 1. Scenario: Using a multi-instance queue manager for high availability with WebSphere Application Server	1	Configuring the JNDI namespace and administered objects	25
Chapter 2. Planning the solution	3	Running the sample JMS application	29
Assumptions	3	Installing WebSphere Application Server.	30
Business overview	3	Configuring WebSphere Application Server in the administrative console.	34
User roles and interactions.	4	Deploying the sample application	40
Technical solution.	5	Verifying the solution	43
Overview: Initial IT configuration	5	Making the messaging infrastructure highly available	45
Overview: The delivered logical topology.	9	Making WebSphere MQ highly available	46
Production physical topology and product mapping	12	Enabling WebSphere Application Server client reconnection	53
Chapter 3. Implementing the solution	15	Verifying the solution	55
Creating a sample initial IT configuration	17	Notices	57
Creating a shared file system	17	Programming interface information	59
		Trademarks	59
		Sending your comments to IBM	61

Figures

1. The initial IT configuration. 9
2. Delivered logical topology 12
3. Production physical topology 13
4. Delivered logical topology 16
5. Objects created in WebSphere MQ 26
6. Objects created in WebSphere Application Server, and the corresponding objects in WebSphere MQ 35
7. A message that travels from the sample JMS client to WebSphere MQ, then onward to the MDB on WebSphere Application Server 40

Chapter 1. Scenario: Using a multi-instance queue manager for high availability with WebSphere Application Server

Starting with an existing WebSphere® Application Server and WebSphere MQ installation, this scenario leads you through the key tasks that are required to make WebSphere MQ highly available. This scenario focuses on WebSphere MQ highly availability, and does not provide detail about the additional high availability that can be achieved with WebSphere Application Server.

This scenario was developed with the use of a sample WebSphere Application Server and WebSphere MQ installation. 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 that are provided with the scenario to verify your progress through each stage.

Note: This scenario was completed with the use of WebSphere MQ Version 7.1 and WebSphere Application Server Version 8. The concepts can be used on later versions of WebSphere MQ and WebSphere Application Server.

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:

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:

 [Scenarios and Patterns](#)


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

Chapter 2. Planning the solution

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 a Linux based operating system.
- You already have a Network File System (NFS) running on a separate host from WebSphere MQ. The NFS is mounted on the WebSphere MQ host and is used to store queue manager data. However, instructions are provided for installing a sample configuration NFS.
- You already have WebSphere MQ Version 7.1 running on your system. However, instructions are provided for installing a sample configuration of WebSphere MQ if you do not already have it.
- You already have WebSphere Application Server Version 8 with Fix Pack 3 running on a separate host in your system. However, instructions are provided for installing a sample configuration of WebSphere Application Server if you do not already have it.
- You have root access to all the hosts in the environment.
- You use both the graphical user interfaces and the command-line equivalents of WebSphere MQ and WebSphere Application Server.

Note: To complete this scenario by using WebSphere Application Server Version 8.5, you must use the WebSphere Application Server full profile.

Tip: Complete this scenario with sandbox systems until you have confidence in the setup.

Business overview

A company wants to make its applications highly available, such that if a queue manager goes offline, queue availability is maintained and clients are switched automatically to the new active instance of a multi-instance queue manager.

To date, company A has a business solution that uses a messaging infrastructure that is provided by WebSphere MQ connected to WebSphere Application Server. Through the WebSphere MQ messaging provider in WebSphere Application Server, Java™ EE applications can use the Java Message Service (JMS) with WebSphere MQ as an external provider of JMS messaging resources.

The company wants to make its applications highly available. If a WebSphere MQ queue manager fails, availability is maintained:

- By using a multi-instance queue manager and shared storage
- By switching clients automatically to the new active instance of a multi-instance queue manager

The WebSphere MQ queue manager might fail because of a network error, a server crash, or anything that provides connectivity to WebSphere MQ.

A highly available messaging infrastructure enables company A to initiate controlled failover. The active instance of the queue manager can be taken offline, enabling maintenance to be completed. During this time, the standby queue manager becomes the active instance and the service to the application is unaffected.

The company decides to use a multi-instance queue manager to complete the scenario. The benefits of a multi-instance queue manager include:

- Basic failover support that is integrated into WebSphere MQ
- Faster failover than HA cluster
- Simple configuration and operation
- Integration with WebSphere MQ Explorer

The limitations of using a multi-instance queue manager include:

- Highly available, high performance networked storage required
- More complex network configuration because the queue manager changes IP address when it fails over
- A standby instance must be restarted manually when a failover occurs

User roles and interactions

Roles that are used for consistent description throughout scenario description, with interactions between roles for scenario tasks and phases.

Although roles to develop the solution are listed, this scenario focuses on deploying a solution that uses a multi-instance queue manager to make WebSphere MQ highly available.

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 that are used by existing IT systems, and determines where enhancements or new components must be written by the Developer.

Integration Developer

The Integration Developer is responsible for creating and testing the software components and linking them together. In some cases the Integration Developer might need information from the Administrator, for example if the Integration Developer must use an existing queue name in their software code. After the Integration Developer finishes creating the software components, the Integration 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 the installation and configuration for all components. If the Administrators are different, they might need to exchange information. For example, in this scenario the NFS administrator must give the location of the mount points to the WebSphere MQ administrator. The Administrator also deploys the software components that are given to them by the Developer.

Resilience Engineer

The Resilience Engineer ensures that the solution is available when needed, the Resilience Engineer monitors availability during peak loads, changing business campaigns, and after a failure. The Resilience Engineer must ensure that the NFS can manage with the new load provided by the highly available solution.

Test Implementor

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?

Technical solution

This scenario describes one way of making an existing WebSphere MQ and WebSphere Application Server environment highly available. With a WebSphere MQ multi-instance queue manager on two WebSphere MQ hosts, messaging functions become highly available. With a multi-instance queue manager running on more than one host, if the active instance fails, one of the standby instances becomes active. The connection to the WebSphere MQ queue manager from WebSphere Application Server is maintained.

Overview: Initial IT configuration

Company A uses an existing IT configuration that is provided by WebSphere MQ Version 7.1 with WebSphere Application Server Version 8 on the Linux operating system. This scenario describes making that initial IT configuration highly available.

The initial IT configuration includes several components that an administrator configures or uses, as shown in figure 1 and described after.

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.

Client mode transport between WebSphere Application Server and WebSphere MQ

In this scenario, client mode transport is used for communication between WebSphere Application Server and WebSphere MQ. Client mode is used for communication between WebSphere Application Server and WebSphere MQ installed on different computers. Client mode transport generally offers lower performance than bindings mode, however it provides more isolation between servers. By default, client mode transport uses the WebSphere MQ client classes for JMS, included in WebSphere Application Server.

What else might I do or be interested in?

Bindings mode can be used only when WebSphere Application Server and WebSphere MQ are installed on the same computer and operating system image.

Network File System for WebSphere MQ to store the queue manager logs and data.

In this scenario, the Network File System (NFS) is configured on a separate server to the WebSphere MQ installation. The NFS is used to store the data files and recovery logs of the queue manager remotely, providing highly available system.

The logs and qmgrs directories of the queue manager are exported from the server that hosts the NFS, allowing other servers with a WebSphere MQ installation to mount to these directories and use them to configure a multi-instance queue manager.

The WebSphere MQ resource adapter

The WebSphere MQ resource adapter allows applications that run on an application server to access WebSphere MQ resources. The WebSphere MQ resource adapter implements the JCA 1.5 interfaces, and allows JMS applications and message driven beans (MDBs), running in an application server, to access the resources of a WebSphere MQ queue manager.

Use the WebSphere MQ resource adapter with WebSphere Application Server Version 7.0 or later. Use the WebSphere MQ resource adapter that is shipped as part of WebSphere Application Server. Only apply fixes that are issued against WebSphere Application Server, not service fixes that are issued against WebSphere MQ. For more information about which version of the WebSphere MQ resource adapter is shipped with WebSphere Application Server, see Identifying which WebSphere MQ client is shipped with WebSphere Application Server.

Java EE application

The application consumes and works with messages on the WebSphere MQ queue, *myQueue*. This application runs on an *application server* of the WebSphere Application Server product that is 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 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 environment in which Java EE application programs run.

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).

Activation specification, `SampleActivationSpec`

A JMS activation specification can be associated with one or more message-driven beans (MDBs) and provides the configuration necessary for them to listen for messages that arrive 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 that arrive 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 the use 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; an inbound message is received by the listener port, which passes the message to a new instance of a message-driven bean for processing.

Destination, Q1

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.

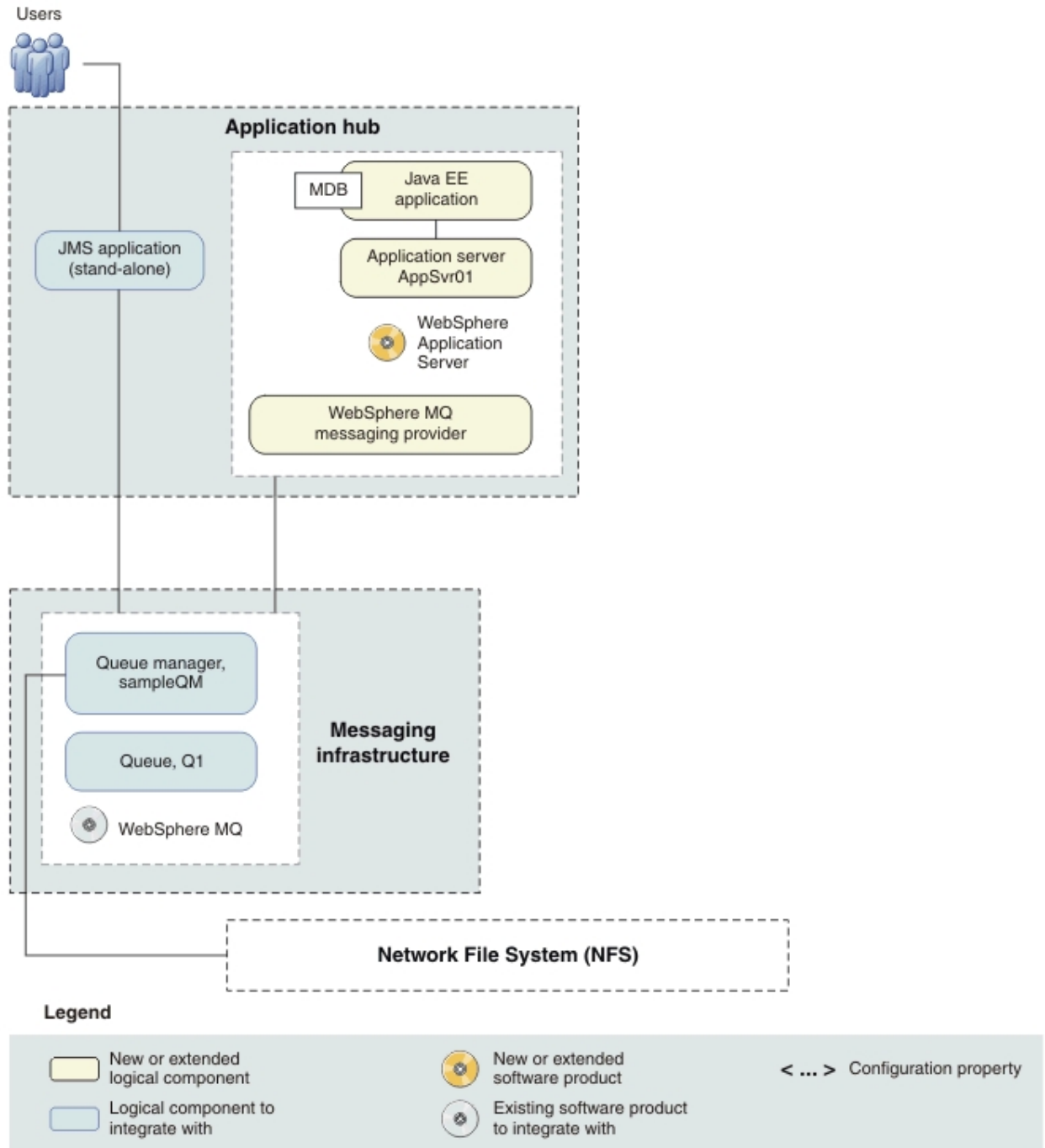


Figure 1. The initial IT configuration. The initial IT configuration includes a WebSphere MQ messaging infrastructure, an Application hub consisting of a stand-alone JMS application and WebSphere Application Server.

Overview: The delivered logical topology

The company creates a highly available messaging infrastructure that is based on the use of a multi-instance queue manager.

The delivered IT configuration includes several components that an administrator configures or uses differently to the initial messaging infrastructure, as shown in Figure 2 on page 12.

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.

Client mode transport between WebSphere Application Server and WebSphere MQ

In this scenario, client mode transport is used for communication between WebSphere Application Server and WebSphere MQ. Client mode transport is used when WebSphere MQ is on a different host from WebSphere Application Server.

Multi-instance queue manager

In this scenario, a multi-instance queue manager is used to provide high availability. A multi-instance queue manager can be started on both hosts simultaneously, one is the active instance and the other the standby. The multi-instance queue manager data is stored on the NFS and accessed by both instances of the queue manager. If the active instance of the queue manager fails, the standby instance becomes active.

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 5.

JNDI administered objects

In this scenario, you create the JMS objects in WebSphere Application Server specifying the full connection details rather than use a *client channel definition table (CCDT)* from WebSphere MQ. A CCDT makes configuring objects in WebSphere Application Server easier because it contains information that is required to connect to WebSphere MQ. If you do not specify a CCDT, you must enter this information yourself. This scenario does not use a CCDT so that you can see more clearly what information is required to make a connection. A CCDT is useful if your client applications connect to a number of alternative queue managers or channels. This scenario does not cover that situation.

WebSphere Application Server and WebSphere MQ high availability

WebSphere Application Server high availability framework enables continued connection with WebSphere MQ queue manager despite failover.

Standby Instance of WebSphere MQ

A second instance of WebSphere MQ is installed on a second host. The second instance of the queue manager is created to identify when the active instance fails, by periodically checking whether the active instance is still running. If the active instance fails or is no longer connected, the standby instance acquires the lock on the queue manager data files and logs. The queue manager restarts and becomes the active instance.

Connection Name List

The connection name list is a comma-separated list that contains the connection details of hosts that are being used to connect to a queue manager.

In this scenario, the connection name list contains entries for the active and standby instances of the multi-instance queue manager and the host that each is on. The host name must be either a DNS name or IP address the port number specifies the WebSphere MQ listener for the queue manager. The WebSphere Application Server JNDI objects try the list in order - once per each reconnection attempt - attempting to connect to a queue manager with the name specified by the **QMANAGER** property and the channel that is listed in the **CHANNEL** property that is listed in the connection factory configuration and the host name and port information that is provided in the Connection Name List. If the queue manager for the first host name was not available, the WebSphere Application Server JNDI objects attempt to connect to the queue manager for the second listed host name. In this scenario, the second host name and port is for the standby instance of WebSphere MQ.

Reconnecting clients

WebSphere Application Server provides its own mechanism to reconnect clients. If the connection fails, the Java EE environment detects the failure and then tries to reconnect the Activation Specification. The Activation Specification uses the information in the connection name list property as before.

After the Activation Specification reconnects to the WebSphere MQ queue manager, the Java EE environment will perform any transactional cleanup, and resume delivering messages.

For more information about client reconnection, see WebSphere MQ automatic client reconnection with the WebSphere MQ classes for JMS.

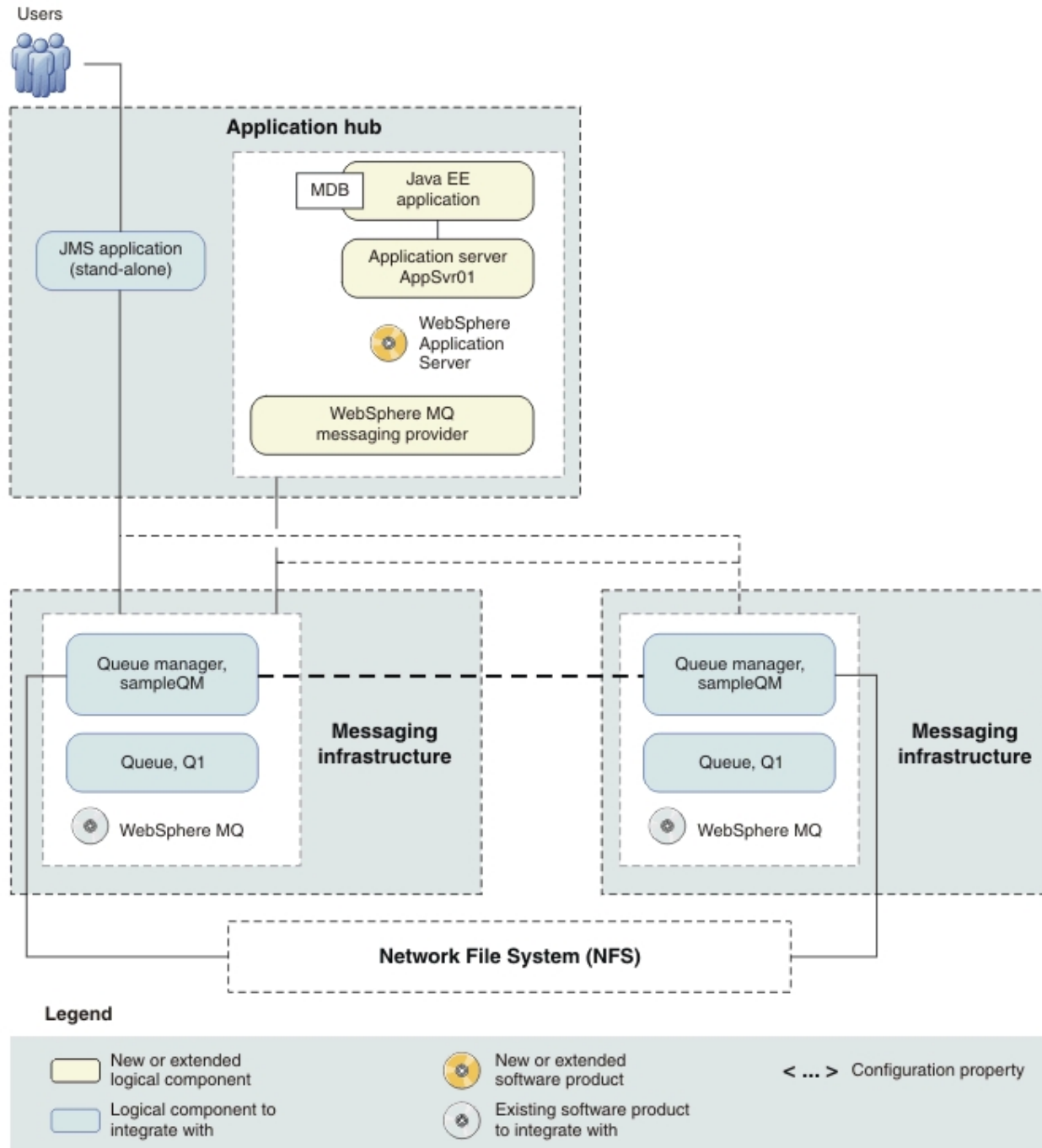


Figure 2. *Delivered logical topology.* The high-level logical topology diagram for software functions that are delivered by making the existing IT configuration highly available. This logical topology shows the new functions as new or extended logical components, integrated with the software functions and products from the initial IT configuration.

Production physical topology and product mapping

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

Figure 3. Production physical topology

Node	Operating system and hardware	Software
Network File System	<ul style="list-style-type: none"> Linux 	<ul style="list-style-type: none"> NFS v4
Host 2 (WebSphere MQ 1)	<ul style="list-style-type: none"> Linux 1 GB physical memory 3 GB free disk space 	<ul style="list-style-type: none"> WebSphere MQ Version 7.1
Host 3 (WebSphere MQ 2)	<ul style="list-style-type: none"> Linux 1-GB physical memory 3-GB free disk space 	<ul style="list-style-type: none"> WebSphere MQ Version 7.1
Host 4 (WebSphere Application Server)	<ul style="list-style-type: none"> Linux 1 GB physical memory 3 GB free disk space 	<ul style="list-style-type: none"> WebSphere Application Server Version 8 Prerequisite software: <ul style="list-style-type: none"> A web browser to use the administrative console.

Chapter 3. Implementing the solution

Implementing the solution in this scenario involves changing the WebSphere MQ configuration to use a multi-instance queue manager for high availability.

Before you begin

The starting point for this scenario is an existing, verified, WebSphere MQ and WebSphere Application Server messaging infrastructure.

If you want to sample the scenario, you can set up a copy of the sample messaging infrastructure as described in “Creating a sample initial IT configuration” on page 17.

About this task

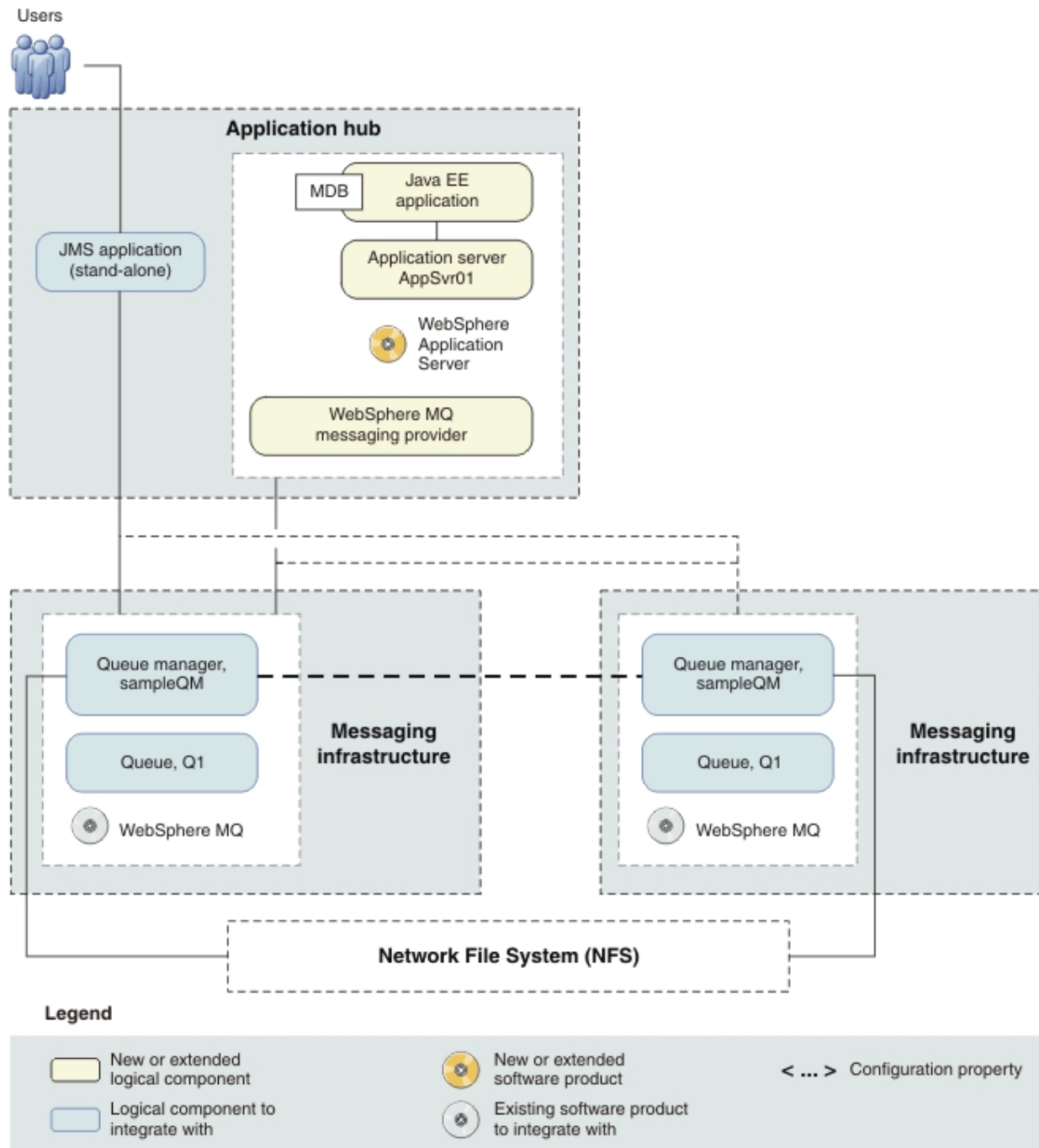


Figure 4. *Delivered logical topology.* The high-level logical topology diagram for software functions that are delivered by making the existing IT configuration highly available. This logical topology shows the new functions as new or extended logical components, integrated with the software functions and products from the initial IT configuration.

After you change the WebSphere MQ configuration to use a multi-instance queue manager, you can test the high availability of the messaging infrastructure. 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 this sample configuration to try out the scenario in the same way as it was originally developed.

Creating a shared file system

Set up and configure the Network File System (NFS), which enables the WebSphere MQ multi-instance queue manager to access the files that are used to store messages over the network. Create the directories and authorities on the NFS server that are exported and to which the multi-instance queue manager mounts.

Before you begin

- Complete this step before the installation of WebSphere MQ.

About this task

The NFS directory is used to share data files and recovery logs for the multi-instance queue manager. It is set up before WebSphere MQ is installed and is used by all installations of WebSphere MQ in the scenario. You create the directories within the NFS to store the shared data, then configure the directories to be accessible from the two WebSphere MQ hosts. The two WebSphere MQ hosts mount to the exported directories to access the shared multi-instance queue manager data.

Procedure

1. On the NFS server, create a user called mqm. In the command line, enter `useradd mqm`.
2. Create a group that is called mqm, enter `groupadd mqm`.
3. Add the mqm user to the mqm group. Enter `usermod -g mqm mqm`.
4. Start the NFS services, enter `chkconfig nfs on`. The NFS services 2, 3, 4, and 5 must be on. Check the status of the services, enter `chkconfig --list nfs`.
5. Create the top-level directory to share. In the command line, enter `mkdir /var/mqm`.
6. Edit the fstab configuration file. In the command line, enter `vi /etc/fstab`.

```
nfs 0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

Why am I doing this?

The fstab file lists file systems that are automatically mounted. To make your configuration highly available, it is important that your file system is automatically mounted to ensure the files that are stored within the NFS are instantly available to the queue manager after an outage.

- a. Add an entry for the new shared directory. At the end of the file, type `/dev/basevg/varmqm /var/mqm ext3 defaults 1 2`. Use the Tab key to match the format of the previous entries.

Why am I doing this?

- /dev/basevg/varmqm lists where the file system can be found
- /var/mqm lists the place in the directory tree where the contents are stored
- ext3 details the type of file system
- defaults is a comma-separated list. For a full list of parameters, use the command **man mount**
- 1 causes the **dump** command to make a backup when the system goes down. If 2 is entered here, no backup is made.
- 2 checks the whole file system for errors when Linux starts. 0 performs no checks and 1 checks the root file system only.

- b. Press the Escape key and enter:wq.
7. Create a logical volume for the /var/mqm file system and specify the logical volume size, enter `lvcreate -L 3G -n varmqm basevg`.

Why am I doing this?

You must create a logical volume to enable the NFS to be exported from NFS01 and mounted onto any other host that has an instance of the multi-instance queue manager. Creating a logical volume creates a partition within the existing volume group on the server, taking a portion, in this case 3 GB, of memory to be allocated to this file system, /var/mqm.

What else might I do or be interested in?

The mount directory becomes the root directory for the /var/mqm file system. It is important for this directory to be new because after it is mounted, any files that are already saved within it, are not accessible.

A confirmation message is displayed:

```
Logical volume "varmqm" created
```

8. Create a mount point for the /var/mqm file system. Enter `mkdir -m 777 /var/mqm`.

Why am I doing this?

You must create a directory for the file system to be mounted on. The `-m` parameter sets the permissions for the mount point.

9. Create the file system device, enter `mkfs.ext3 /dev/basevg/varmqm`.

Why am I doing this?

The file system must use the device that is specified, as detailed in the `fstab` file, /dev/basevg/varmqm. It creates a journal to store all of the changes that are made to the file system. Providing quick recovery of the file system if it fails, with a smaller likelihood of it becoming corrupted.

After running the command, the following message is displayed:

```
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
393216 inodes, 786432 blocks
39321 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=805306368
24 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912
```

```
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

This filesystem will be automatically checked every 33 mounts or 180 days, whichever comes first. Use tune2fs -c or -i to override.

10. Mount the `/var/mqm/` file system device onto the logical volume. Enter `mount /var/mqm`.
11. Create the `/var/mqm/logs` directory to share. In the command line, enter `mkdir /var/mqm/logs`.
12. Edit the `fstab` configuration file. In the command line, enter `vi /etc/fstab`.
 - a. Add an entry for the new shared directory. At the end of the file, type `/dev/basevg/mqmlogs /var/mqm/logs ext3 defaults 1 2`. Use the Tab key to match the format of the previous entries.
 - b. Press the Escape key and enter: `wq`.
13. Create a logical volume for the `/var/mqm/logs` file system and specify the logical volume size, enter `lvcreate -L 3G -n mqmlogs basevg`. A confirmation message is displayed:

```
Logical volume "mqmlogs" created
```

14. Create a mount point for the `/var/mqm/logs` file system. Enter `mkdir -m 777 /var/mqm/logs`.

Why am I doing this?

You must create a directory for the file system to be mounted on. The `-m` parameter sets the permissions for the mount point.

15. Create the file system device, enter `mkfs.ext3 /dev/basevg/mqmlogs`. After running the command, the following message is displayed:

```
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
393216 inodes, 786432 blocks
39321 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=805306368
24 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912
```

Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 33 mounts or 180 days, whichever comes first. Use tune2fs -c or -i to override.

16. Mount the /var/mqm/logs file system device onto the logical volume. Enter `mount /var/mqm/logs`.
17. Create the /var/mqm/qmgrs directory to share. In the command line, enter `mkdir /var/mqm/qmgrs`.
18. Edit the fstab configuration file. In the command line, enter `vi /etc/fstab`.
 - a. Add an entry for the new shared directory. At the end of the file, type `/dev/basevg/mqm/qmgrs /var/mqm/qmgrs ext3 defaults 1 2`. Use the Tab key to match the format of the previous entries.
 - b. Press the Escape key and enter: `wq`.
19. Create a logical volume for the /var/mqm/qmgrs file system and specify the logical volume size, enter `lvcreate -L 3G -n mqmqmgrs basevg`. A confirmation message is displayed:

Logical volume "mqmqmgrs" created

20. Create a mount point for the /var/mqm/qmgrs file system. Enter `mkdir -m 777 /var/mqm/qmgrs`.

Why am I doing this?

You must create a directory for the file system to be mounted on. The `-m` parameter sets the permissions for the mount point.

21. Create the file system device, enter `mkfs.ext3 /dev/basevg/mqm/qmgrs`. After running the command, the following message is displayed:

```
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
393216 inodes, 786432 blocks
39321 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=805306368
24 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912
```

Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 33 mounts or 180 days, whichever comes first. Use tune2fs -c or -i to override.

22. Mount the /var/mqm/qmgrs file system device onto the logical volume. Enter `mount /var/mqm/qmgrs`.
23. Export the NFS mount points.
 - a. Open the /etc/exports file for editing. In the command line, enter `vi /etc/exports`. Elevated privileges are required to edit this file.
 - b. Add the directories to be exported to the end of the file.

```
/var/mqm MQ01Host(rw,no_root_squash,fsid=0) MQ02Host(rw,no_root_squash,fsid=0)
/var/mqm/logs MQ01Host(rw,no_root_squash,nohide) MQ02Host(rw,no_root_squash,nohide)
/var/mqm/qmgrs MQ01Host(rw,no_root_squash,nohide) MQ02Host(rw,no_root_squash,nohide)
```

- c. In the command line, type `/usr/sbin/exportnfs -a`. The changes to the `/etc/exports` file are applied.

24. Run the NFS daemon, in the command line enter `/etc/init.d/nfs start`. The following message is displayed:

```
Starting NFS services:           [ OK ]
Starting NFS quotas:           [ OK ]
Starting NFS daemon:           [ OK ]
Starting NFS mountd:           [ OK ]
```

Results

The NFS for WebSphere MQ is ready to be used as the data store for the multi-instance queue manager.

What to do next

- Verify the shared file system behavior. Check that the shared file system meets the requirements for storing the queue manager data of a multi-instance queue manager. Complete the instructions as described in *Verifying shared file system behavior* in the WebSphere MQ information center.
- Complete the instructions in “Installing WebSphere MQ.”

Installing WebSphere MQ

The subtopics in this task describe the steps to; prepare for the installation of WebSphere MQ on the Linux operating system, install WebSphere MQ on the first WebSphere MQ host, and verify the installation. After WebSphere MQ is installed, check that it is able to use the NFS. Completing this check allows you to share data between the different instances of the WebSphere MQ multi-instance queue manager.

Before you begin

- You must have local root access to install WebSphere MQ.
- You must have permission to create a user and group on the host.
- Ensure that you have sufficient disk space, up to 600 MB to install WebSphere MQ for Linux.

About this task

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

Preparing to install WebSphere MQ

Prepare the first WebSphere MQ host for the installation of WebSphere MQ.

Before you begin

- You must have an NFS set up as described in “Creating a shared file system” on page 17.

About this task

Create a user and group to install WebSphere MQ with and then mount the NFS to the first WebSphere MQ host so the shared directories are accessible.

Procedure

1. Log in to the first WebSphere MQ host as root and create a user and group called `mqm`.
 - a. Create the `mqm` user, in the command prompt enter `useradd mqm`.
 - b. Add the `mqm` user to the `mqm` group, enter `usermod -g mqm mqm`.
2. Check the `mqm` *UID* and *GID* are the same as on the NFS server.

Why am I doing this?

A user and group of the name `mqm` is required because it owns directories and files that contain the resources that are associated with WebSphere MQ.

- a. Check that the *UID* and *GID* are the same, enter `id mqm`. An example output is:

```
uid=11968(mqm) gid=11968(mqm) groups=11968(mqm) context=root:system_r:unconfined_t:SystemLow-SystemHigh
```

- b. If they are different on each host, set the *UID* and *GID* of `mqm` on the host. Type `groupmod -g 11968 mqm usermod -u 11968 -g 11968 mqm`.

3. Mount the NFS on the WebSphere MQ host.

- a. On the WebSphere MQ host, create three directories. In the command prompt type: `mkdir /var/mnt/mqm, mkdir /var/mnt/mqm/logs, mkdir /var/mnt/mqm/qmgrs`.
- b. Change the ownership and permissions of these directories. Enter `chown -R mqm:mqm /var/mnt, chmod -R ug+rw /var/mnt`. Navigate to `/var/mnt` and check the ownership and permissions that are updated correctly, enter `ls -l`. If the updates were not successful, enter the following commands:

```
chown mqm:mqm /var/mnt/mqm
chown mqm:mqm /var/mnt/mqm/logs
chown mqm:mqm /var/mnt/mqm/qmgrs
chmod ug+rw /var/mnt/mqm
chmod ug+rw /var/mnt/mqm/logs
chmod ug+rw /var/mnt/mqm/qmgrs
```

The directory must be owned by the user and group `mqm`, and the access permissions are set to read, write, and execute for the user and group.

- c. Edit `/etc/fstab` to make the system mount the NFS automatically upon starting. In a command prompt, enter `vi /etc/fstab`. Add the following to the file:

```
NFSHostName:/ /var/mnt/mqm nfs defaults,rw,hard,intr 0 0
NFSHostName:/logs /var/mnt/mqm/logs nfs defaults,rw,hard,intr 0 0
NFSHostName:/qmgrs /var/mnt/mqm/qmgrs nfs defaults,rw,hard,intr 0 0
```

Why am I doing this?

The mount is *hard* because the client continually attempts to reach the server until it is able to reach it and then continue from where it left off; preserving data transfers in the event of server failure. A soft mount returns an error. If the server goes offline, all the data is lost. *intr* allows the user to unmount the client manually if the server loses connection.

Enabling the communication and sharing of the queue manager logs and data to provide a highly available infrastructure.

- d. Mount the files that are listed, in `/etc/fstab` from the server. Enter `mount -a`.
- e. Navigate to the `/usr/sbin` directory.
- f. Enter `./rpc.mountd` and `./rpc.nfsd`. The `mountd` and `nfsd` daemons are started.

Results

Your Linux system is ready for you to install WebSphere MQ.

What to do next

Complete the instructions in “Installing WebSphere MQ” on page 47 to install WebSphere MQ.

Installing WebSphere MQ

Install WebSphere MQ on the Linux operating system by using the command line.

Before you begin

- You must have local administrator authority when you are installing. Define this authority through the Linux facilities.
- Ensure that the host name does not contain any spaces.
- Ensure that you have sufficient disk space, up to 600 MB, to fully install WebSphere MQ for Linux.

If you want to investigate the requirements for installing WebSphere MQ in more detail, see Checking hardware and software requirements in the WebSphere MQ information center.

About this task

To install the WebSphere MQ packages to the default location on the first WebSphere MQ host, use the `rpm` command as described in this task. You can either install all components, or specify individual components to install. However, note that the minimum components that you must install are:

- `MQSeriesRuntime`
- `MQSeriesServer`
- `MQSeriesSamples`

You must install the Samples to verify the installation and use the sample applications provided.

Procedure

1. Log in to the WebSphere MQ host as root.
2. Insert the WebSphere MQ for Linux Server DVD into the DVD drive.
3. Accept the WebSphere MQ license. In a terminal, navigate to the `/` directory then enter `./mqlicense.sh -accept`. The following message is displayed:

Licensed Materials - Property of IBM

5724-H72

(C) Copyright IBM Corporation 1994, 2019 All rights reserved.

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

Agreement accepted: Proceed with install.

4. Install the WebSphere MQ packages. Enter `rpm -ivh MQSeries*.rpm` to install WebSphere MQ to the `/opt/mqm` directory. The following is displayed throughout the installation process:

```
Preparing... ##### [100%]
 1:MQSeriesRuntime ##### [ 4%]
 2:MQSeriesClient ##### [ 9%]
 3:MQSeriesJava ##### [13%]
 4:MQSeriesJRE ##### [17%]
 5:MQSeriesGSKit ##### [22%]
 6:MQSeriesMan ##### [26%]
 7:MQSeriesMsg_cs ##### [30%]
 8:MQSeriesMsg_de ##### [35%]
 9:MQSeriesMsg_es ##### [39%]
10:MQSeriesMsg_fr ##### [43%]
11:MQSeriesMsg_hu ##### [48%]
12:MQSeriesMsg_it ##### [52%]
13:MQSeriesMsg_ja ##### [57%]
14:MQSeriesMsg_ko ##### [61%]
15:MQSeriesMsg_pl ##### [65%]
16:MQSeriesMsg_pt ##### [70%]
17:MQSeriesMsg_ru ##### [74%]
18:MQSeriesMsg_Zh_CN ##### [78%]
19:MQSeriesMsg_Zh_TW ##### [83%]
20:MQSeriesSamples ##### [87%]
21:MQSeriesSDK ##### [91%]
22:MQSeriesServer ##### [96%]
23:MQSeriesTXClient ##### [100%]
```

The minimum components that you must install are:

- MQSeriesRuntime
- MQSeriesServer
- MQSeriesSamples

To install individual components to the default location, such as runtime and server components, use the following command:

```
rpm -ivh MQSeriesRuntime-*.rpm MQSeriesServer-*.rpm MQSeriesSamples-*.rpm
```

What else might I do or be interested in?

If you want to install to a non-default location, use the `rpm --prefix` command. The installation path must be to an empty directory, the root of an unused file system, or a path that does not exist. The length of the path is limited to 256 bytes and must not contain spaces. For example, to install the runtime and server components to `/opt/customLocation`, use the following command:

```
rpm --prefix /opt/customLocation -ivh MQSeriesRuntime-*.rpm MQSeriesSamples-*.rpm
MQSeriesServer-*.rpm
```

Results

WebSphere MQ is now installed and you are now ready to verify the installation to ensure that the queue manager and queue communicate.

What to do next

To configure the JNDI administered objects, complete the instructions in “Configuring the JNDI namespace and administered objects.”

Configuring the JNDI namespace and administered objects

Configure the JNDI namespace to administer objects that the sample application can use; a queue, two channels and a listener.

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 in a local file system. A file system is used because it is the simplest JNDI mechanism for a sample scenario.

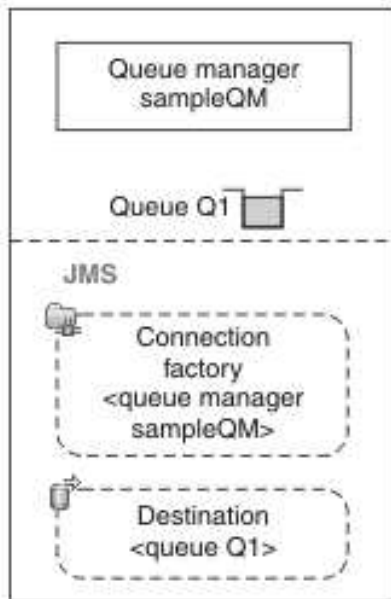
What else might I do or be interested in?

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.

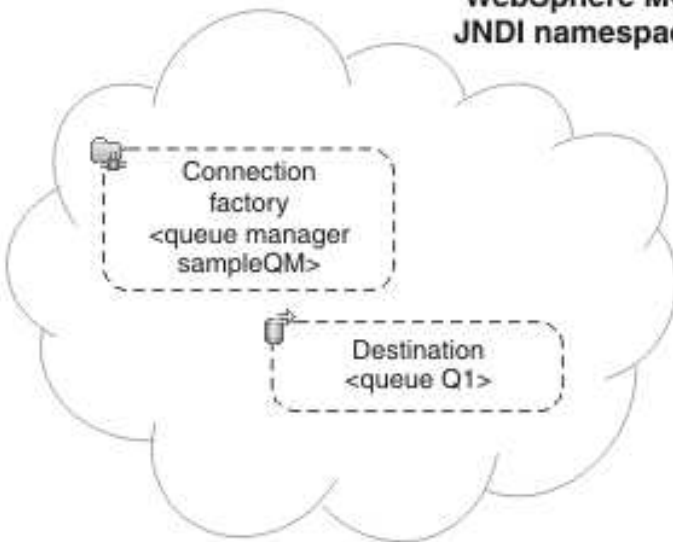
- Use of the WebSphere MQ MQSC commands to create administered objects in the JNDI namespace. The JMS application can look up the administered objects to connect to WebSphere MQ and to access WebSphere MQ destinations with which to send or receive messages.

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

WebSphere MQ



WebSphere MQ JNDI namespace



Legend

< ... > Configuration property

Figure 5. Objects created in WebSphere MQ

Procedure

1. Log in as the root user.

2. Create a JNDI-Directory. Navigate to the / directory, enter mkdir JNDI-Directory.
3. Log in as the mqm user.
4. Create a queue manager.
 - a. Navigate to the /opt/mqm/bin directory.
 - b. Enter `./crtmqm -ld /var/mnt/mqm/logs -md /var/mnt/mqm/qmgrs sampleQM`.

Why am I doing this?

The `-ld` and `-md` parameters are set to use the directories that are mounted on the NFS, providing a highly available set up. The paths must be set up because they are not the default paths.

The following message is displayed:

```
WebSphere MQ queue manager created.
Directory '/var/mnt/mqm/qmgrs/sampleQM' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'sampleQM'.
Default objects statistics : 71 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

5. Start the queue manager, enter `./strmqm -x sampleQM`.

Why am I doing this?

If you omit the `-x`, the queue manager starts, but does not permit a standby instance; it is not a multi-instance queue manager.

The following message is displayed:

```
WebSphere MQ queue manager 'sampleQM' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'sampleQM' during the log replay phase.
Log replay for queue manager 'sampleQM' complete.
Transaction manager state recovered for queue manager 'sampleQM'.
WebSphere MQ queue manager 'sampleQM' started using V7.1.0.0.
```

6. Create a configuration file to connect WebSphere MQ to the initial context of the *JNDI namespace*
 - a. Create a file called `MyAdmin.config`. Navigate to the `/opt/mqm/java/bin` directory. Enter `vi MyAdmin.config`.

Why am I doing this?

This file creates a separate configuration specifically for JMS administration of the administered objects in the JNDI Namespace.

- b. Type the following into the file:

```
INITIAL_CONTEXT_FACTORY=com.sun.jndi.fscontext.RefFSCContextFactory
PROVIDER_URL=file:///JNDI-Directory
SECURITY_AUTHENTICATION=none
export classpath="/opt/mqm/java/lib/com.ibm.mq.jar:/opt/mqm/java/lib/com.ibm.mq.jms.Nojndi.jar:/opt/mqm/java/lib/com.ibm
```

- c. Press Escape then type `:wq`.
- d. On the command line, enter `chmod u+x MyAdmin.config`. The file now has execute permissions.

7. Create the queue manager administered objects.
 - a. Start the MQSC commands on sampleQM. Enter `runmqsc sampleQM`.
 - b. Create a local queue, enter `DEFINE QLOCAL('myQueue')`.
 - c. Create a server-connection channel, enter `DEFINE CHANNEL(JMS.CHANNEL) CHLTYPE(SVRCONN) TRPTYPE(TCP) MCAUSER('mqm')`.
 - d. Create a client-connection channel, enter `DEFINE CHANNEL(JMS.CHANNEL) CHLTYPE(CLNTCONN) TRPTYPE(TCP) CONNAME(hostNameOfMQ) QMNAME('sampleQM')`.
 - e. Create a listener object, enter `DEFINE LISTENER('listener1') TRPTYPE(TCP) CONTROL(QMGR)`.
 - f. Enter `end`. The following message is displayed:

```
2 : end
Four MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

8. Set the JMS environment.
 - a. Log out of the mqm user and the login as root.
 - b. Set the Java path. Enter `export PATH=$PATH:/opt/mqm/java/jre64/jre/bin`.
 - c. Navigate to the `/opt/mqm/java/bin` directory. Enter `./setjmsenv64` to set the file paths that are associated with Java. The following message is displayed:

```
MQ_JAVA_INSTALL_PATH is /opt/mqm/java
MQ_JAVA_DATA_PATH is /var/mqm
MQ_JAVA_LIB_PATH is /opt/mqm/java/lib64
CLASSPATH is /opt/mqm/java/lib/com.ibm.mq.jar:/opt/mqm/java/lib/com.ibm.mqjms.jar:/opt/mqm/samp/wmqjava/samples:/opt/mqm/sa
```

Note: If there is a Java error, repeat steps b and c again.

9. Start the JMSAdmin. Enter `./JMSAdmin -cfg MyAdmin.config`. The **InitCtx>** prompt is displayed.
10. Define a JMS context and change to it.
 - a. Type `define ctx(JMS)`.
 - b. Type `change ctx(JMS)`. The prompt changes to **InitCtx/JMS>**.
11. Create a connection factory that is called myCF, enter `define cf(myCF) tran(CLIENT) host(hostNameOfMQ) chan(JMS.CHANNEL) qmgr(sampleQM)`.
12. Create a queue called myQueue. Enter `define q(Q1) queue(myQueue) qmgr(sampleQM)`
13. Verify the initial context. Enter `display ctx`. The output provides the following information:

```
JMSADM4089 InitCtx/JMS

      .bindings                java.io.File
a myCF                        com.ibm.mq.jms.MQConnectionFactory
a myQueue                    com.ibm.mq.jms.MQQueue

3 Object(s)
0 Context(s)
3 Binding(s), 2 Administered
```

Type `end` to exit the JMSAdmin.

Results

The WebSphere MQ objects that are required to use the sample JMS application are configured.

What to do next

Complete the instructions in “Running the sample JMS application” to verify WebSphere MQ communication.

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 to the WebSphere MQ host: `sampleJMSAppLinuxHA.zip` (in online information center), then extract the contents. The package contains a sample JMS application `.jar` file and shell script files for running the application.

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

- In one terminal run the `runresponder.sh`, responder client then waits for a message.
- In another terminal, run the `runrequester.sh`, the requester client starts and sends a request message and receives a reply.

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

Procedure

1. Log in as the `mqm` user.
2. Run the responder client.
 - a. On the command line, navigate to the directory that contains the `sampleJMSApp.jar` and the `runresponder.sh` files.
 - b. Enter `./runresponder.sh`. The responder client waits for a message.

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

```
> Waiting for message.
```

3. Run the requestor client.
 - a. On the command line, navigate to the directory that contains the `sampleJMSApp.jar` and the `runrequestor.sh` files.
 - b. Enter `./runrequestor.sh`.

Results

The requestor client terminal, shows the connection status, the message it sent, and 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 . . .
```

The responder client terminal, shows 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 that are shown in the two terminals verify that the requester and responder clients of the sample application can communicate with each other through WebSphere MQ.

What to do next

Complete the instructions in “Installing WebSphere Application Server” to begin the WebSphere Application Server section of the scenario.

Installing WebSphere Application Server

Install WebSphere Application Server on your Linux system.

Preparing your Linux system for installation

Prepare your Linux system for the installation of WebSphere Application Server. Check the hardware and software requirements and set permissions for the files and directories that are created by the installation of WebSphere Application Server.

Before you begin

- Ensure that the default shell for your Linux operating system is `/bin/bash`. Use the following command `readlink /bin/sh` to ensure that your default shell is bash not dash.
- Check for any post release technical notes available for your operating system at the product support site [Support home - WebSphere Application Server](#).
- Make sure that your Linux system has the following items:
 - Kernel and C runtime library
 - Current[®] and all compatibility versions of the C++ runtime library
 - X Window System libraries
 - GTK runtime libraries
- Ensure that there is adequate disk space. Installing all features and products requires approximately 2 GB of disk space. This estimate includes the following products, components, and features:
 - Main application server product installation
 - Profiles
 - Sample applications
 - IBM[®] HTTP Server
 - Web Server plug-ins
 - Application Client for WebSphere Application Server
- Verify that prerequisites and corequisites are at the required release levels, which can be reviewed on the [Support hardware and software website](#).
- Complete any distribution-specific setup requirements as described in [Preparing Linux system for installation](#).

About this task

This task checks that the default settings of your Linux operating system meet the requirements for the installation of WebSphere Application Server Version 8.0.

Procedure

1. Log on to the WebSphere Application Server host as root and set the suitable authorities for the installation.
 - a. Verify the **umask** setting. In the command line, type `umask`.
 - b. Set the **umask** setting to `022`, enter `umask 022`. Setting the **umask** to `022` allows only the user to write data, but other users can read the data.

Why am I doing this?

The `umask` sets the permission modes for files and directories that are created by the installation. Preventing WebSphere Application Server from creating its own permissions for files or directories that you might not want to be written by other users of the group.

2. Increase the `ulimit` setting in the bash command shell profile to prevent `addNode` and `importWasprofile` problems. Set the `ulimit` on your Linux command shells by the `.bashrc` file. The shell profile script is found under your home directory. Open `.bashrc` in a text editor and enter `ulimit -n 8192`.

Results

Your Linux operating system is now ready for WebSphere Application Server to be installed.

What to do next

Follow the instructions in “Installing IBM Installation Manager” to install IBM Installation Manager, which is required to install WebSphere Application Server.

Installing IBM Installation Manager

Install IBM Installation Manager, which is required to install WebSphere Application Server Version 8.0.

Before you begin

Obtain the files for IBM Installation Manager in one of three ways:

- Use the copy of IBM Installation Manager and the product files from your product disk.
- Download and unpack the appropriate compressed file from the Passport Advantage® site that contains the pre-configured IBM Installation Manager and necessary product repositories.
- Download and unpack the appropriate compressed file for your operating system from IBM Installation Manager Downloads. In this scenario Version 1.5.3 is used.

Note: For more information about installing IBM Installation Manager, see Managing installations with Installation Manager in the IBM Installation Manager product documentation.

About this task

IBM Installation Manager is a common installer for many IBM software products. WebSphere Application Server Version 8 and later are installed with IBM Installation Manager. This task explains how to install IBM Installation Manager as an administrator on the WebSphere Application Server host. You must have execute permissions on IBM Installation Manager directories, package directories, and parent directories. Without the permissions, unexpected behavior or failures can occur. For information about installing as a non-administrator or group, see Installing as an administrator, non-administrator, or group. in the IBM Installation Manager product documentation.

Note: For information about installing earlier versions of WebSphere Application Server that do not require IBM Installation Manager, see Installing the Network Deployment product in the WebSphere Application Server product documentation.

Procedure

1. Log in to the WebSphere Application Server host as the root user.
2. Install IBM Installation Manager as an administrator.

- a. Navigate to the directory you unpacked the installation files to.
- b. In the command line, enter `./installc -c`.
- c. Press N then A to accept the license agreement.
- d. Press I to install.

Results

IBM Installation Manager is now installed.

What to do next

To use IBM Installation Manager to install WebSphere Application Server, complete the instructions in “Installing WebSphere Application Server.”

Installing WebSphere Application Server

Install WebSphere Application Server on a Linux operating system by using a response file and the IBM Installation Manager.

Before you begin

Before you start this task, follow the instructions in “Preparing your Linux system for installation” on page 31 to prepare for installing WebSphere Application Server and the latest fix pack on a Linux operating system.

- Make sure that you either have available the installation disk that is labeled WebSphere Application Server Network Deployment, or know where to access the root directory of the downloaded Passport Advantage image on your hard disk.
- Set your rollback preference in the installation manager by following the instructions on the Rollback preference page.
- Make sure that you have a valid response file. Download the response file, `WASResponseFile.zip` (in online information center). Edit the response file in a text editor, change the first instance of the *repository location* tag to the location of your `repository.config` file. For more information about response files, see Sample response file: Installing IBM WebSphere Application Server Network Deployment in the WebSphere Application Server product documentation.

About this task

Install WebSphere Application Server, to the latest maintenance level, on the Linux operating system by using a response file to create the single-machine installation of a stand-alone application server.

Procedure

1. Using the IBM Installation Manager, start the installation of the response file.
 - a. Navigate to the `/opt/IBM/InstallationManager/eclipse/tools` subdirectory in the directory where you installed IBM Installation Manager.
 - b. Enter `./imcl -acceptLicense -showProgress input response_file_path_and_name -log log_file_path_and_name`.

Why am I doing this?

The **-acceptLicense** command must be included to indicate acceptance of all license agreements. The **-showProgress** command shows progress of the installation when it is run in silent mode.

If you would like to record a log of the installation, use the

What else might I do or be interested in?

`-log`

parameter and then list the file path where you want to save the log.

2. Create a WebSphere Application Server profile.
 - a. Navigate to `/opt/IBM/WebSphere/AppServer/bin/`
 - b. In the command line, enter `./manageprofiles -create -templatePath /opt/IBM/WebSphere/AppServer/profileTemplates/default.`
 - c. Start the server. Navigate to `/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/bin` and enter `./startServer.sh server1.`

Results

WebSphere Application Server Version is installed with the latest level of maintenance.

What to do next

Complete the instructions in “Configuring WebSphere Application Server in the administrative console” to create the required WebSphere Application Server objects.

Configuring WebSphere Application Server in the administrative console

Configure WebSphere Application Server resources to enable communication with the first WebSphere MQ host.

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: Initial IT configuration” on page 5.

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.

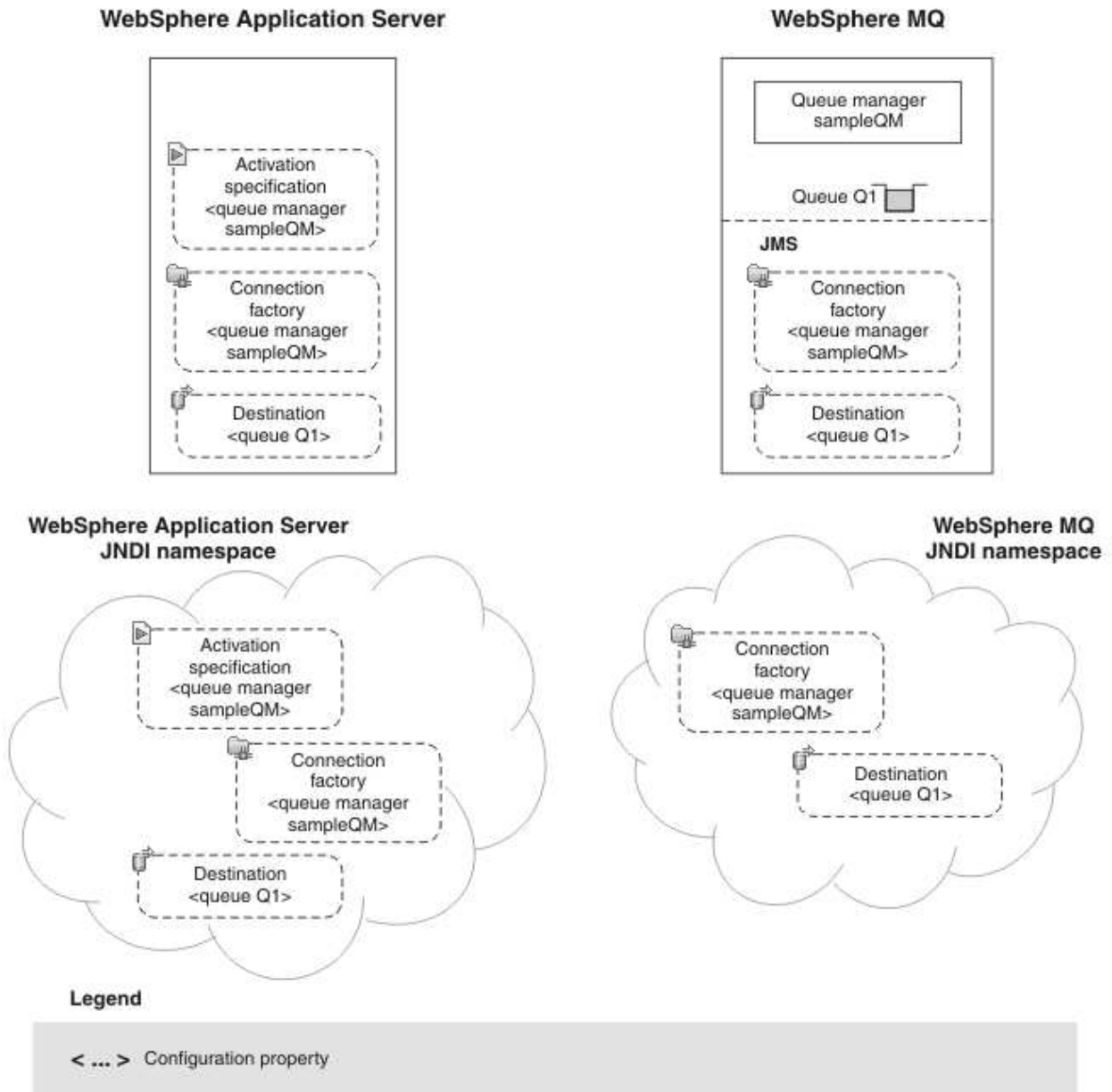


Figure 6. Objects created in WebSphere Application Server, and the corresponding objects in WebSphere MQ

Procedure

1. If WebSphere Application Server is not already started, navigate to `/opt/IBM/WebSphere/AppServer/bin` and enter `./startServer.sh server1`. An example of the message that is displayed is:

```

ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1/startServer.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process ID is 26590
    
```

2. Start a web browser, in the address field enter `host_name:9060/ibm/console`. The `host_name` is the name of your WebSphere Application Server host.

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 that are made by other users.

3. Create an activation specification.

Why am I doing this?

If you want to use message-driven beans with a messaging provider that does not have a JCA 1.5 resource adapter (for example the WebSphere MQ messaging provider), you cannot use activation specifications and therefore you must configure your beans against a listener port.

- a. In the administrative console, click **Resources > JMS > Activation specifications**.
- b. Ensure that the scope is set to **Cell=cell_name**, then click **New**.
- c. Select **WebSphere MQ messaging provider**, then click **OK**. The Configure basic attributes page is displayed.
- d. On the Configure basic attributes page, complete the following fields as follows:

Name A name of your choice; for example, SampleActivationSpec

JNDI name

Enter jms/PiShopReplyMDB

- e. Click **Next**.
- f. On the Specify MDB destination data page, complete the following field:

Destination JNDI name

The JNDI name for the destination to be used by the activation specification: jms/wasQueue

Tip: This name must match what you specify later when you configure the queue.

- g. Click **Next**. On the Select connection method page, leave as default, Enter all the required information into this wizard and then click **Next**.
- h. On the Supply queue manager details page complete, the following field:

Queue manager or queue-sharing group name

Specify sampleQM.

Tip: The queue manager name must be the same as the queue manager you created in the WebSphere MQ configuration.

- i. Click **Next** and then on the Enter connection details page, change the following fields:

Transport

Change to **CLIENT**.

Why am I doing this?

In this mode, WebSphere Application Server connects in client mode, where the WebSphere MQ messaging provider within WebSphere Application Server is on a different host to the WebSphere MQ queue manager. Client transport uses TCP/IP to communicate between the two.

Select **Enter host and port information in the form of separate host name and port values**.

Hostname

The host name of the first WebSphere MQ host.

Port Default value is **1414**, if you specified a port number during the WebSphere MQ configuration, enter this value here.

Server connection channel

Specify `JMS.CHANNEL`

Why am I doing this?

Defines the name of the channel a client can use to connect to the queue manager, `sampleQM`.

- j. Click **Next** and then **Next** again.
 - k. On the Summary page, check the summary and 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 is used for outbound messages from the MDB, whereas the listener port is used for inbound messages.

What else might I do or be interested in?

Before the JMS 1.1 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 Connection Factory Settings page is displayed.

- d. On the Configure basic attributes page, complete the required fields as follows:

Why am I doing this?

The name that you choose for the connection factory 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.

Name Specify a name of your choice; for example: myCF

JNDI name

Specify jms/replyCF

- e. Click **Next** and leave the default selection of **Enter all the required information into this wizard** and then click **Next**.
- f. On the Supply the queue manager details page, complete the required field as follows:

Queue manager or queue-sharing group name

Specify sampleQM

Tip: The queue manager name must be the same as the queue manager you created in the WebSphere MQ configuration.

- g. Click **Next** and on the Enter connections details page, complete the required fields as follows:

Transport

Change to **CLIENT**.

Why am I doing this?

In this mode, WebSphere Application Server connects in client mode, where the WebSphere MQ messaging provider within WebSphere Application Server is on a different host to the WebSphere MQ queue manager. Client transport uses TCP/IP to communicate between the two.

Select **Enter host and port information in the form of separate host name and port values**.

Hostname

The host name of the first WebSphere MQ host.

Port Specify the port number that is used by the first WebSphere MQ host.

Server connection channel

Specify JMS.CHANNEL

Why am I doing this?

Defines the name of the channel a client can use to connect to the queue manager, sampleQM.

- h. Click **Next** and then **Next** again.
 - i. On the Summary page, check the summary and click **Finish**.
5. Create a queue.

Why am I doing this?

A queue is a type of JMS destination, which is 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.

Name A name of your choice; for example: myQueue

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.

Tip: This name must match what you specified earlier when you configured the listener port.

Base queue name

myQueue

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

- e. Click **OK**.
6. Click the **Save** link in the Messages section at the top of the pane to save your changes to the master configuration.
7. Stop the application server.
- a. Log in to the WebSphere Application Server host as the root user.
 - b. Navigate to `/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/bin`.
 - c. In the command line, enter `./stopServer.sh server1`. The following message is displayed:

```
ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1/stopServer.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: server1
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server server1 stop completed.
```

8. Start the application server. In the command line, enter `./startServer.sh server1`. An example of the message that is displayed is:

ADMU0116I: Tool information is being logged in file
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1/startServer.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process ID is 26590

Results

WebSphere Application Server is now configured with the necessary objects to enable communication with WebSphere MQ.

What to do next

Complete the instructions in “Deploying the sample application” to deploy the sample MDB application.

Deploying the sample application

Deploy the sample message driven bean (MDB) application to WebSphere Application Server.

Before you begin

- Download the sample application. Click the following link and save to the host of WebSphere Application Server: [sampleMDB.ear](#).

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 29.

The following diagram shows a message that travels 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.

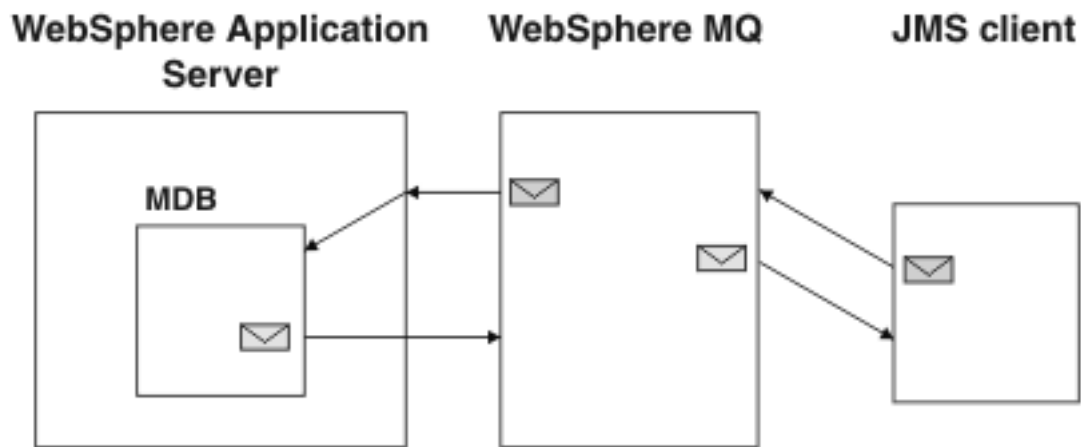


Figure 7. A message that travels from the sample JMS client to WebSphere MQ, then onward to the MDB on WebSphere Application Server

Procedure

1. If the WebSphere Application Server is not already started, navigate to `/opt/IBM/WebSphere/AppServer/profiles/Appsrv01/bin` and enter `./startServer.sh server1`.
2. Start a web browser, in the address field enter `host_name:9060/ibm/console`. The `host_name` is the name of your host.
3. Click **Applications > Application Types > WebSphere enterprise 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 more 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 must use wherever action on the `sampleMDB` application is described). To do change the name, edit the **Application name** field to specify your chosen name. Select **Deploy enterprise beans** and click **Next**.

8. Click the **Step 3** link to go to the Bind listeners for message-driven beans page. In this step, you specify the information that is required 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 must 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 in the administrative console" on page 34.

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 must 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.

The other properties are optional, they are provided so that you can override values that are 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. (The optional

Destination JNDI name field overrides the destination that is configured on the activation specification.)

- c. Click **Next**.
9. On the Map resource references to resources page, map the connection factory resource reference that is 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 is 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 in the runtime environment. Resource references eliminate you having to change the 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 must select is the connection factory `myCF` that you specified in.

- b. Select **replyCF** 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 installs successfully, click the **Save** link that is displayed at the end of the messages to save your changes to the master configuration.

What to do next

To verify the solution, complete the instructions in “Verifying the solution.”

Verifying the solution

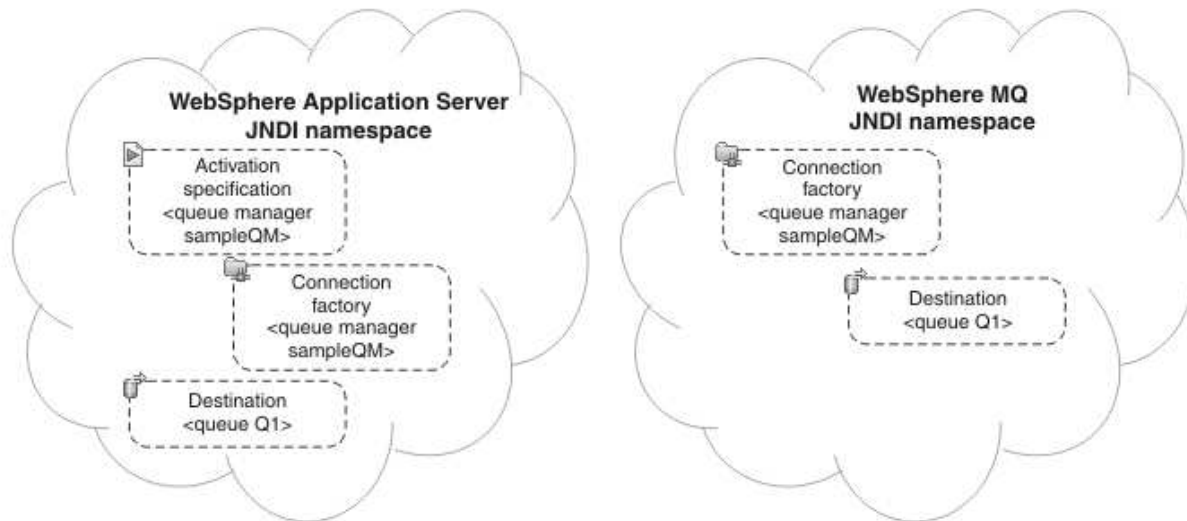
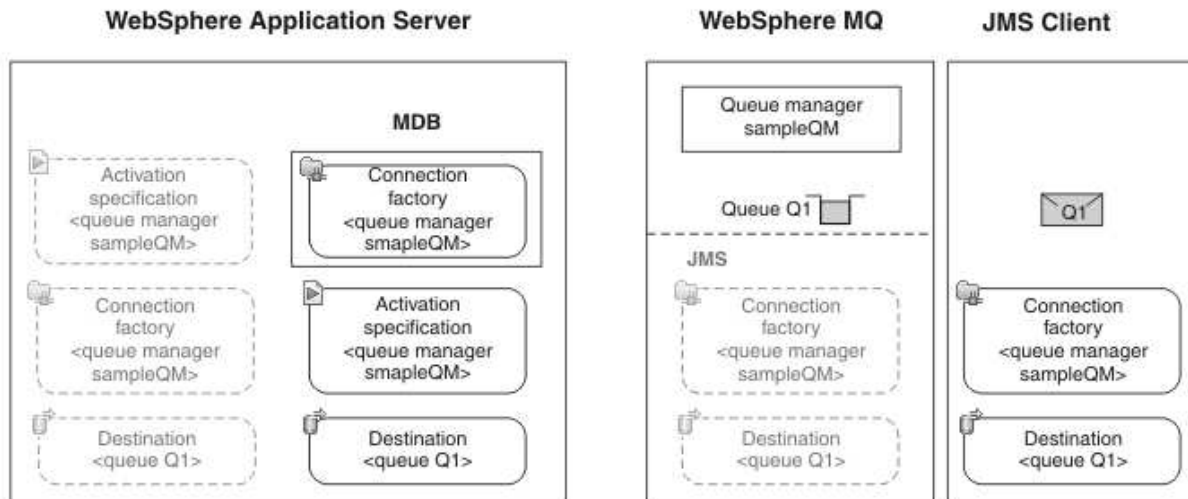
Run the sample JMS and message drive bean (MDB) applications to verify that they can communicate with each other.

Before you begin

You must have WebSphere Application Server and WebSphere MQ configured as described in “Configuring WebSphere Application Server in the administrative console” on page 34 and “Configuring the JNDI namespace and administered objects” on page 25.

About this task

In Verifying the installation of WebSphere MQ, you ran the supplied JMS sample application to verify that the requester and responder clients of the application can communicate through WebSphere MQ. In “Deploying the sample application” on page 40, 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 responder application. Verifying that messages are being passed between WebSphere MQ and WebSphere Application Server.



Legend

< ... > Configuration property

Procedure

1. If WebSphere Application Server is not already started, enter `./startServer server1`.
2. Start a web browser, in the address field enter `host_name:9060/ibm/console`. The `host_name` is the name of your host.
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 that is contained in the sample application. Check in the

WebSphere Application Server error logs *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 information.

5. Log in to the WebSphere MQ host as the mqm user.
6. Run the requestor client.
 - a. On the command line, navigate to the directory that contains the sampleJMSApp.jar and the runrequestor.sh files.
 - b. Enter ./runrequestor.sh.

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 'BakedBeans - 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.

The sample It configuration is implemented, and verified. WebSphere Application Server can connect to WebSphere MQ for the successful transmission of messages.

What to do next

To make the messaging structure highly available, complete the instructions in “Making the messaging infrastructure highly available.”

Making the messaging infrastructure highly available

Install WebSphere MQ on a second Linux host to enable the creation of a multi-instance queue manager. Alter the JNDI objects to use a connection list to connect to both WebSphere MQ hosts. Alter WebSphere Application Server to attempt to reconnect to the queue manager independent of its host.

Before you begin

- You must have the sample IT configuration as described in “Creating a sample initial IT configuration” on page 17.
- You must have a second host to install WebSphere MQ on, the host must meet the requirements as described in “Production physical topology and product mapping” on page 12.

Making WebSphere MQ highly available

To make the WebSphere MQ configuration highly available, you add a second WebSphere MQ host to the sample configuration. You then create an active instance of a multi-instance queue manager on the first WebSphere MQ host, and a standby instance on the second WebSphere MQ host.

About this task

To make the WebSphere MQ configuration highly available create a multi-instance queue manager. You create an instance of the queue manager on both WebSphere MQ hosts. One is active and the other is standby, ready to become active if the one in use fails.

Preparing to install WebSphere MQ on a second Linux host

Prepare the second WebSphere MQ host for the installation of WebSphere MQ.

About this task

Create a user and group to install WebSphere MQ and then mount the NFS to the first WebSphere MQ host so the shared directories are accessible.

Procedure

1. Log in to the second WebSphere MQ host as root and create a user and group called `mqm`.
 - a. Create the `mqm` user, in the command prompt enter `useradd mqm`.
 - b. Add the `mqm` user to the `mqm` group, enter `usermod -g mqm mqm`.
2. Check the `mqm` *UID* and *GID* are the same as the NFS server and the first WebSphere MQ host.

Why am I doing this?

A user and group of the name `mqm` is required because it owns directories and files that contain the resources that are associated with WebSphere MQ.

- a. Check that the *UID* and *GID* are the same, enter `id mqm`. An example output is:

```
uid=11968(mqm) gid=11968(mqm) groups=11968(mqm) context=root:system_r:unconfined_t:SystemLow-SystemHigh
```

- b. If they are different on each host, set the *UID* and *GID* of `mqm` on the host. Type `groupmod -g 11968 mqm` `usermod -u 11968 -g 11968 mqm`.
3. Mount the NFS on the WebSphere MQ host.
 - a. On the WebSphere MQ host, create three directories. In the command prompt type: `mkdir /var/mnt/mqm`, `mkdir /var/mnt/mqm/logs`, `mkdir /var/mnt/mqm/qmgrs`.
 - b. Change the ownership and permissions of these directories. Enter `chown -R mqm:mqm /var/mnt`, `chmod -R ug+rwx /var/mnt`. Navigate to `/var/mnt` and

then check the ownership and permissions are updated, enter `ls -l`. If the updates were not successful, enter the following commands:

```
chown mqm:mqm /var/mnt/mqm
chown mqm:mqm /var/mnt/mqm/logs
chown mqm:mqm /var/mnt/mqm/qmgrs
chmod ug+rwx /var/mnt/mqm
chmod ug+rwx /var/mnt/mqm/logs
chmod ug+rwx /var/mnt/mqm/qmgrs
```

The directory must be owned by the user and group `mqm`, and the access permissions are set to read, write, and execute for the user and group.

- c. Edit `/etc/fstab` to make the system mount the NFS automatically upon starting. In a command prompt, enter `vi /etc/fstab`. Add the following to the file:

```
NFSHostName:/ /var/mnt/mqm nfs defaults,rw,hard,intr 0 0
NFSHostName:/logs /var/mnt/mqm/logs nfs defaults,rw,hard,intr 0 0
NFSHostName:/qmgrs /var/mnt/mqm/qmgrs nfs defaults,rw,hard,intr 0 0
```

Why am I doing this?

The mount is *hard* because the client continually attempts to reach the server until it is able to reach it and then continue from where it left off; preserving data transfers in the event of server failure. A soft mount returns an error, if the server goes offline all data is lost. *intr* allows the user to unmount the client manually if the server loses connection.

Enabling the communication and sharing of the queue manager logs and data to provide a highly available infrastructure.

- d. Mount the files that are listed, in `/etc/fstab` from the server. Enter `mount -a`.
- e. Navigate to the `/usr/sbin` directory.
- f. Enter `./rpc.mountd` and `./rpc.nfsd`. The *mountd* and *nfsd* daemons are started.

Results

Your Linux system is ready for you to install WebSphere MQ.

What to do next

Complete the instructions in “Installing WebSphere MQ” to install WebSphere MQ.

Installing WebSphere MQ

Install WebSphere MQ on the second WebSphere MQ host with the command line.

Before you begin

- You must have local administrator authority when you are installing. Define this authority through the Linux facilities.
- Ensure that the host name does not contain any spaces.
- Ensure that you have sufficient disk space, up to 600 MB, to fully install WebSphere MQ for Linux.

If you want to investigate the requirements for installing WebSphere MQ in more detail, see Checking hardware and software requirements in the WebSphere MQ information center.

About this task

Install the WebSphere MQ packages on the second WebSphere MQ host by using the command line.

Procedure

1. Log in to the second WebSphere MQ host as root.
2. Insert the WebSphere MQ for Linux Server DVD into the DVD drive.
3. Accept the WebSphere MQ license.
 - a. In a terminal, navigate to the / directory. Enter `./mqlicense.sh -accept`.
The following message is displayed:

```
Licensed Materials - Property of IBM
```

```
5724-H72
```

```
(C) Copyright IBM Corporation 1994, 2019 All rights reserved.
```

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

```
Agreement accepted: Proceed with install.
```

- a. Enter `rpm -ivh MQSeries*.rpm` to install WebSphere MQ to the `/opt/mqm` directory. The following is displayed throughout the installation process:

```
Preparing... ##### [100%]  
 1:MQSeriesRuntime ##### [ 4%]  
 2:MQSeriesClient ##### [ 9%]  
 3:MQSeriesJava ##### [13%]  
 4:MQSeriesJRE ##### [17%]  
 5:MQSeriesGSKit ##### [22%]  
 6:MQSeriesMan ##### [26%]  
 7:MQSeriesMsg_cs ##### [30%]  
 8:MQSeriesMsg_de ##### [35%]  
 9:MQSeriesMsg_es ##### [39%]  
10:MQSeriesMsg_fr ##### [43%]  
11:MQSeriesMsg_hu ##### [48%]  
12:MQSeriesMsg_it ##### [52%]  
13:MQSeriesMsg_ja ##### [57%]  
14:MQSeriesMsg_ko ##### [61%]  
15:MQSeriesMsg_pl ##### [65%]  
16:MQSeriesMsg_pt ##### [70%]  
17:MQSeriesMsg_ru ##### [74%]  
18:MQSeriesMsg_Zh_CN ##### [78%]  
19:MQSeriesMsg_Zh_TW ##### [83%]  
20:MQSeriesSamples ##### [87%]  
21:MQSeriesSDK ##### [91%]  
22:MQSeriesServer ##### [96%]  
23:MQSeriesTXClient ##### [100%]
```

The minimum components that you must install are:

- MQSeriesRuntime
- MQSeriesServer
- MQSeriesSamples

To install individual components to the default location, such as runtime and server components, use the following command:

```
rpm -ivh MQSeriesRuntime-*.rpm MQSeriesServer-*.rpm MQSeriesSamples-*.rpm
```

Why am I doing this?

You must install the Samples to verify the installation and use the sample applications provided.

What else might I do or be interested in?

If you want to install to a non-default location, use the **rpm --prefix** command. The installation path must be to an empty directory, the root of an unused file system, or a path that does not exist. The length of the path is limited to 256 bytes and must not contain spaces. For example, to install the runtime and server components to `/opt/customLocation`, use the following command:

```
rpm --prefix /opt/customLocation -ivh MQSeriesRuntime-*.rpm MQSeriesSamples-*.rpm  
MQSeriesServer-*.rpm
```

Results

WebSphere MQ is now installed and you are now ready to configure the multi-instance queue manager.

What to do next

Complete the instructions in “Creating a multi-instance queue manager” to configure the WebSphere MQ multi-instance queue manager.

Creating a multi-instance queue manager

Create a queue manager that has multiple instances on different servers. An active instance and a standby to create a highly available messaging system.

Before you begin

- Ensure that the shared files system to be used by the WebSphere MQ data files and recovery logs from the multi-instance queue manager is exported and then mounted onto the second WebSphere MQ host.
- Check the user and group ID for the user `mqm` is the same in all systems that are involved with the multi-instance queue manager. You can check this using the `id mqm` command.

About this task

Multi-instance queue managers are instances of the same queue manager that is configured on two or more hosts. One is the active instance and the other is the standby. If the active fails, the standby instance on the second host automatically takes over. Multi-instance queue managers can be used to configure your own highly available messaging systems that are based on WebSphere MQ, without requiring a cluster technology.

Procedure

1. Stop the queue manager on the first WebSphere MQ host. In the command line, navigate to `/opt/mqm/bin` and enter `./endmqm -w sampleQM`.

Why am I doing this?

Including the `-w` ensures that the shutdown of the queue manager is orderly, it waits for the queue manager to stop, rather than forcing an immediate shutdown.

2. Copy the queue manager information locations.
 - a. In the command line, enter `./dspmqinf -o command sampleQM`. The output from this command is the command, and arguments that are required to be entered on the second WebSphere MQ host. An example output is:

```
addmqinf -s QueueManager -v Name=sampleQM -v Directory=sampleQM -v Prefix=/var/mqm -v DataPath=/var/mnt/mqm/qmgrs/sampleQM
```

3. Create the queue manager on the second WebSphere MQ host.
 - a. In the command line, navigate to `/opt/mqm/bin` and enter `./addmqinf -s QueueManager -v Name=sampleQM -v Directory=sampleQM -v Prefix=/var/mqm -v DataPath=/var/mnt/mqm/qmgrs/sampleQM`. The following message is displayed:

WebSphere MQ configuration information added.

4. Add the JNDI bindings file to the second WebSphere MQ host. Use FTP to copy the `JNDI-Directory` directory from the first host to the `/` directory on the second host.

Results

The `sampleQM` queue manager can be started on both hosts, one will be the active instance, and one the standby.

What to do next

Complete the instructions in “Testing the multi-instance queue manager” to test the queue manager instances.

Testing the multi-instance queue manager

Test that starting two instances of the queue manager initiates an active and a standby instance by starting both instances of the queue manager.

About this task

Start an instance of the multi-instance queue manager on both of the WebSphere MQ hosts. Use the `dspsmq` command to display the status of the instance of the queue manager.

Procedure

1. Start the queue manager on the first WebSphere MQ host, on the command line, enter `./strmqm -x sampleQM`.

Why am I doing this?

The `-x` parameter is used to start a multi-instance queue manager, if there is not another instance, the queue manager starts and becomes the active. If there is an instance of this queue manager on a separate server, this instance becomes standby.

The following message is displayed:

```
WebSphere MQ queue manager 'sampleQM' starting.  
5 log records accessed on queue manager 'sampleQM' during the log replay phase.  
Log replay for queue manager 'sampleQM' complete.  
Transaction manager state recovered for queue manager 'sampleQM'.  
WebSphere MQ queue manager 'sampleQM' started using V7.1.0.0.
```

2. Start the queue manager on the second WebSphere MQ host, on the command line, enter `./strmqm -x sampleQM`. The following message is displayed:

```
WebSphere MQ queue manager 'sampleQM' starting.  
A standby instance of queue manager 'sampleQM' has been started. The active instance is running elsewhere.
```

3. Verify the status of the instances. On either host, enter `./dspmq -x`. The following message is displayed:

```
QMNAME(sampleQM) STATUS(Running) STANDBY(Permitted)  
INSTANCE(hostName) MODE(Active)  
INSTANCE(hostName) MODE(Standby)
```

Results

An active instance of the multi-instance queue manager is now running on the first host and a standby instance on the second host.

Troubleshooting:

- You must check that the full path name of the WebSphere MQ data files and the WebSphere MQ recovery logs are identical.

When you start the instances without the `-x` parameter, an error or warning message is displayed.

Note: flag.

- Not using the `-x` flag to start the active or standby instance.

Active:

```
strmqm sampleQM
```

, the following message is displayed: WebSphere MQ queue manager 'sampleQM' started.

Standby:

```
strmqm sampleQM
```

the following message is displayed: Active:

- Not using the `-x` flag to start the active and using `-x` on the standby instance.

Active:

```
strmqm sampleQM
```

no warning or error messages are displayed.

Standby:

```
strmqm -x sampleQM
```

the following message is displayed: WebSphere MQ queue manager 'sampleQM' starting. WebSphere MQ queue manager 'sampleQM' is already running elsewhere. It does not permit standby instances.

- Using **-x** on the active and not on the standby instance.

Active:

```
strmqm -x sampleQM
```

no warning or error messages are displayed.

Standby:

```
strmqm sampleQM
```

WebSphere MQ queue manager 'sampleQM' starting. WebSphere MQ queue manager 'sampleQM' is already running elsewhere. It permits standby instances.

- Using **-x** on both the active and standby instances.

Active:

```
strmqm -x sampleQM
```

no warning or error messages are displayed.

Standby:

```
strmqm -x sampleQM
```

WebSphere MQ queue manager 'sampleQM' starting. A standby instance of queue manager 'sample QM' has been started. The active instance is running elsewhere.

What to do next

Complete the instructions in “Configuring the JNDI administered objects” to run the sample application with the multi-instance queue manager.

Configuring the JNDI administered objects

Configure the JNDI administered objects for the stand-alone application to use a connection list, enabling connection from both WebSphere MQ hosts.

Before you begin

- You must have the JNDI objects as described in “Configuring the JNDI namespace and administered objects” on page 25.

About this task

The instructions in this task describe configuring the JNDI administered objects to use a connection name list. The connection name list stores the host name and port value of both of the WebSphere MQ hosts. Copy the configured bindings file to the second WebSphere MQ host to enable the sample JMS application to use the JNDI objects.

Procedure

1. Configure the client-connection channel.
 - a. Log in to the first WebSphere MQ host as the mqm user.
 - b. Navigate to the /opt/mqm/bin directory.

- c. In the command line, enter `./runmqsc sampleQM`.
- d. Alter the client-connection channel, enter `ALTER CHANNEL(JMS.CHANNEL) CHLTYPE(CLNTCONN) TRPTYPE(TCP) CONNAME('hostNameOfMQ1,hostNameOfMQ2') QMNAME(sampleQM)`.

What else might I do or be interested in?

The `CONNAME` value is altered to include the host name of the second WebSphere MQ host. The other values are kept the same.

- e. Enter `DISPLAY CHANNEL(JMS.CHANNEL)` to confirm the changes.
- f. Type end to exit.
2. Configure the connection factory.
 - a. Log in to the first WebSphere MQ host as the root user.
 - b. Navigate to the `/opt/mqm/java/bin` directory.
 - c. In the command line, enter `./JMSAdmin -cfg MyAdmin.config`.
 - d. Type `change ctx(JMS)`. The prompt changes to **InitCtx/JMS>**.
 - e. In the **InitCtx/JMS>** prompt, type `alter cf(myCF) CONNECTIONNAMELIST(hostNameOfMQ1,hostNameOfMQ2)`.
 - f. Enter `DISPLAY cf(myCF)` to confirm the changes.
 - g. Type end to exit.
3. Copy the JNDI objects to the second WebSphere MQ host.
 - a. Use FTP to transfer the `/JNDI-Directory` from the first WebSphere MQ host to the root (`/`) directory on the second host.

Results

The JNDI objects use a connection name list to communicate with WebSphere MQ. If the first host is unresponsive, the object connects to the second host.

Enabling WebSphere Application Server client reconnection

To make the solution highly available, configure the JMS resources in WebSphere Application Server to use a connection name list to connect to multiple WebSphere MQ hosts.

Before you begin

- You must have the JMS resources as described in “Configuring WebSphere Application Server in the administrative console” on page 34.

Note: For information about high availability in WebSphere Application Server, see *Establishing high availability in the WebSphere Application Server product documentation*.

Note: For more information about client reconnection in WebSphere Application Server and the `CONNECTIONNAMELIST` property, see *WebSphere MQ automatic client reconnection with the WebSphere MQ classes for JMS*.

Configuring the JMS resources

Configure the WebSphere Application Server connection factory and activation specification to accept connections from multiple hosts.

About this task

The instructions in this task describe configuring the JMS resources to use a connection name list with the details of both of the WebSphere MQ hosts. The connection factory client reconnect option is changed to RECONNECT so that a connection attempt is made when one of the multi-instance queue manager instances is taken offline.

Procedure

1. If WebSphere Application Server is not already started, navigate to `/opt/IBM/WebSphere/AppServer/bin` and enter `./startServer.sh server1`. An example of the message that is displayed is:

```
ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1/startServer.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process ID is 26590
```

2. Start a web browser, in the address field enter `host_name:9060/ibm/console`. The `host_name` is the name of your WebSphere Application Server host.

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 that are made by other users.

3. Edit the connection factory to use a connection name list.

Note: To make the changes with WebSphere Application Server Version 7, complete the instructions in Configuring multi-instance queue manager connections with WebSphere MQ messaging provider custom properties in the WebSphere Application Server product documentation.

- a. In the administrative console, click **Resources > JMS > Connection factories**.
 - b. Click **myCF**.
 - c. In the Connection panel, select **Enter host and port information in the form of a connection name list**.
 - d. In the **Connection name list** field, enter `hostNameOfMQ1(Port),hostNameOfMQ2(Port)`.
4. Edit the connection factory advanced properties.
 - a. In the **Additional Properties** pane, click **Advanced properties**.
 - b. From the **client reconnect options** menu, click **RECONNECT**.

Why am I doing this?

The sample JMS application is running in the client container, automatic client reconnection with a **CONNECTIONNAMELIST** is supported. For more information about client reconnection, see WebSphere MQ automatic client reconnection with the WebSphere MQ classes for JMS.

- c. Click **OK**, then click **Save**.
5. Edit the activation specification to use a connection name list.
 - a. In the administrative console, click **Resources > JMS > Activation specifications**.

- b. Click **myActivationSpec**.
 - c. In the Connection panel, select **Enter host and port information in the form of a connection name list**.
 - d. In the **Connection name list** field, enter `hostNameOfMQ1(Port),hostNameOfMQ2(Port)`.
 - e. Click **OK**, then click **Save**.
6. Stop the application server
 - a. Log in to the WebSphere Application Server host as the root user.
 - b. Navigate to `/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/bin`.
 - c. In the command line, enter `./stopServer.sh server1`. The following message is displayed:

```

ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1/stopServer.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: server1
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server server1 stop completed.

```

7. Start the application server. In the command line, enter `./startServer.sh server1`. An example of the message that is displayed is:

```

ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1/startServer.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process ID is 26590

```

What to do next

Complete the instructions in “Verifying the solution” to verify the solution.

Verifying the solution

To verify that communication can continue if there is a queue manager failure, run the sample JMS application. Take down the active instance of the queue manager and run the sample JMS application again.

Before you begin

- You must have WebSphere Application Server and WebSphere MQ configured as described in “Configuring WebSphere Application Server in the administrative console” on page 34 and “Configuring the JNDI namespace and administered objects” on page 25.
- You must have a multi-instance queue manager on two WebSphere MQ hosts as described in “Creating a multi-instance queue manager” on page 49.

About this task

In this task you run the sample application by using the queue manager on the first WebSphere MQ host to verify that the application is functioning. The first WebSphere MQ host is taken offline to simulate a queue manager failure, the second instance of the queue manager becomes active and the MDB reconnects to the new instance. The application continues to function.

Procedure

1. If WebSphere Application Server is not already started, enter `./startServer server1`.
2. Start a web browser, in the address field enter `host_name:9060/ibm/console`. The `host_name` is the name of your host.
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 that is contained in the sample application. Check in the WebSphere Application Server error logs `install_directory/profiles/AppSrv01/logs/server1` for more information.

Error sampleMDB failed to start. Check the logs for server server1 on node mymachineNode01 for details.

Error An error occurred while starting sampleMDB. Check the logs for server server1 on node mymachineNode01 for more information.

5. Log in to the first WebSphere MQ host as the mqm user.
6. Run the requestor client.
 - a. On the command line, navigate to the directory that contains the `sampleJMSApp.jar` and the `runrequestor.sh` files.
 - b. Enter `./runrequestor.sh`.The application sends the message to the MDB and receives a response.
7. Verify that the active instance changes when there is a queue manager failure.
 - a. Stop the first WebSphere MQ host.
 - b. On the second WebSphere MQ host, in the command line, enter `./dspmq -x`. The message that is displayed is:

```
QMNAME(sampleQM)                               STATUS(Running)
INSTANCE(MQHost2Name) MODE(Active)
```

This message shows that the standby instance has now become the active instance.

8. Run the requestor client on second host.
 - a. Log in to the second host as the mqm user.
 - b. On the command line, navigate to the directory that contains the `sampleJMSApp.jar` and the `runrequestor.sh` files.
 - c. Enter `./runrequestor.sh`. The application sends the message to the MDB and receives a response.

Results

The standby queue manager automatically becomes the active instance after the first host is taken offline. The sample application continues to work.

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 ([®] 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.



Printed in USA