BEA MessageQ

Installation and Configuration Guide for Windows NT

BEA MessageQ for Windows NT Version 5.0
Document Edition 4.0
October 1998

Copyright

Restricted Rights Legend

Trademarks or Service Marks

MessageQ Installation and Configuration Guide for Windows NT

Document Edition	Part Number	Date	Software Version
4.0	850-001010-003	October 1998	MessageQ for Windows NT, Version 5.0

Contents:

Preface

Purpose of This Document vii
Who Should Read This Document vii
How This Document Is Organized vii
How to Use This Document viii
Opening the Document in a Web Browser viii
Printing from a Web Browser ix
Documentation Conventions x
Related Documentation xii
MessageQ Documentation xii
Contact Information xiii
Documentation Support xiii
Customer Support xiii

1. Installing MessageQ

Preparing to Install the MessageQ Software 1-1
Checking the Software Distribution Kit 1-1
Media 1-2
Hardcopy and Online Documentation 1-2
Complying with the Installation Prerequisites 1-3
Backing up the System Disk 1-3
Installing the MessageQ Software on Windows NT Systems 1-3
Choosing Which Subsets to Install 1-4
Installing MessageQ on Systems Already Running MessageQ 1-4
Installing MessageQ for Windows NT 1-5
Recovering from Installation Errors 1-13
Using the Online User Documentation 1-13

Performing Postinstallation Tasks 1-14
Configuring the Network 1-15
Starting the MessageQ Service for Windows NT 1-15
Adding the MessageQ Drive and Directory to Your Path 1-16
Converting Your Group Initialization File to Version 5.0 1-17

2. Configuring MessageQ

Overview of MessageQ System Configuration 2-1
Message Queuing Bus 2-2
Message Queuing Groups 2-3
Message Queues 2-3
Using Naming 2-5
Cross-Group Connections 2-6
Server Processes 2-6
Configuring a Message Queuing Group 2-8
Defining Parameters in the Group Initialization File 2-10
Default Values for Parameters 2-11
Identifying the Group Initialization File Version 2-11
Setting Group Characteristics in the Profile Section 2-12
Setting Message Recovery System Parameters in the MRS Section 2-15
Configuring the Client Library Server in the CLS Section 2-19
Enabling Network Connections in the Cross-Group Section 2-21
Defining the Message Routing Database in the Route Section 2-23
Route Discovery 2-26
Visit Counts 2-27
Client/Server Message Routing 2-28
Defining Queues in the Queue Configuration Section 2-29
Defining Name-to-Queue Translations in the Group Name Section 2-36
Configuring Global Naming 2-37
Configure Groups to Run or Use the Naming Agent 2-37
Configure the Namespace 2-38
Configure a Default Namespace Path for Each Group 2-40
Define the Queue Names in the Group Initialization File 2-40
Using the Test Utility to Test Your Group Configuration 2-42

3. Managing MessageQ

Using the Monitor Utility 3-1
Connecting to the MessageQ Environment 3-2
Connecting to a Group from the Command Line 3-2
Connecting to a Running Group From the Monitor Utility 3-3
Starting a Message Queuing Group 3-5
Starting a Group from the Command Line 3-5
Starting a Group From the Monitor Utility 3-6
Creating a Custom Group Startup Icon on the Desktop 3-7
Starting a Group from the Startup Program Group 3-7
Shutting Down a Running Group 3-8
Shutting Down a Group from the Command Line 3-8
Shutting Down a Group From the Monitor Utility 3-9
Shutting Down a Group From the Group Control Process System Menu . 3-9
Starting a Cross-Group Connection 3-10
Stopping a Cross-Group Connection 3-10
Starting a Queue 3-11
Stopping a Queue 3-11
Running the Client Library Server 3-12
Starting CLS 3-12
Stopping the CLS 3-13
Restricting Remote Access to CLS 3-13
Managing Message Recovery Services 3-14
Replaying Journal Messages 3-15
Printing Journal Files 3-16
Changing Group Characteristics at Runtime 3-18
Enabling or Disabling Quotas 3-19
Setting Quotas for Dynamic Queues 3-20
Enabling Queue Quotas for Selective Queues 3-20
Defining Timeout Intervals for Link Drivers 3-21

4. Monitoring MessageQ

Invoking the Monitor Utility 4-1
Enabling Statistics 4-2
Enabling Quotas 4-2

Viewing Link and Queue Activity 4-3
Viewing Link Traffic Counts 4-3
Viewing Link Traffic Rates 4-4
Viewing Link Traffic Detail 4-5
Viewing Link Detail 4-6
Viewing Queue Traffic Counts 4-7
Viewing Queue Traffic Rates 4-8
Viewing Queue Traffic Detail 4-10
Viewing Queue Detail 4-10

5. Troubleshooting

Error Logging 5-1
Tracing 5-2
Enabling Tracing 5-2
Disabling Tracing 5-3

Index

Preface

Purpose of This Document

This document provides information on installing and configuring BEA MessageQ on Windows NT systems.

Who Should Read This Document

This document is intended for the following audiences:

- system installers who will install BEA MessageQ on supported platforms
- system administrators who will configure, manage, and troubleshoot BEA MessageQ on supported platforms

How This Document Is Organized

BEA MessageQ Installation and Configuration Guide for Windows NT is organized as follows:

- Chapter 1, “Installing MessageQ,” preinstallation, installation, and postinstallation tasks.
- Chapter 2, “Configuring MessageQ,” provides instructions on configuring a message queuing group, defining parameters in a group initialization file, and testing the group configuration.

- Chapter 3, “Managing MessageQ,” describes how to perform common management tasks, such as connecting to a MessageQ environment, starting and stopping groups and queues, running the Client Library Server, managing message recovery, changin group characteristics at runtime, and enabling and disabling quotas.
- Chapter 4, “Monitoring MessageQ,” describes how to use the Monitor utility to view link and queue activity.
- Chapter 5, “Troubleshooting,” describes how to use the MessageQ Event Watcher and tracing capabilities to diagnose problems with your MessageQ application.

How to Use This Document

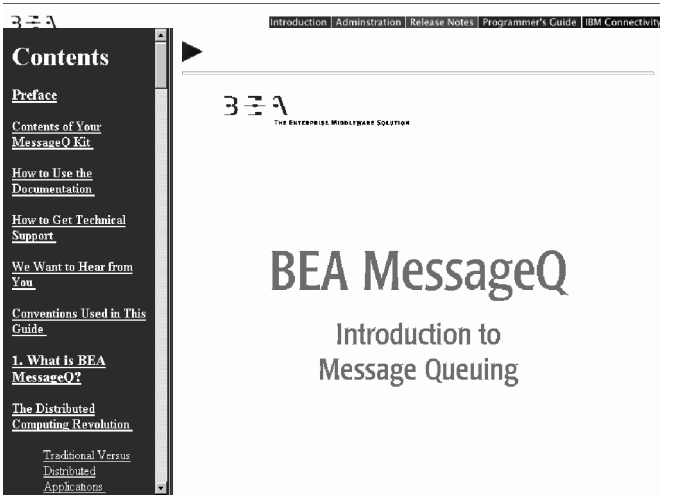
This document is designed primarily as an online, hypertext document. If you are reading this as a paper publication, note that to get full use from this document you should access it as an online document via the BEA MessageQ Online Documentation CD. The following sections explain how to view this document online, and how to print a copy of this document.

Opening the Document in a Web Browser

To access the online version of this document, open the `index.htm` file in the top-level directory of the BEA MessageQ Online Documentation CD. On the main menu, click the Introduction to Message Queuing button. Figure 1 shows the online document with the clickable navigation bar and table of contents.

Note: The online documentation requires a Web browser that supports HTML version 3.0. Netscape Navigator version 3.0 or Microsoft Internet Explorer version 3.0 or later are recommended.

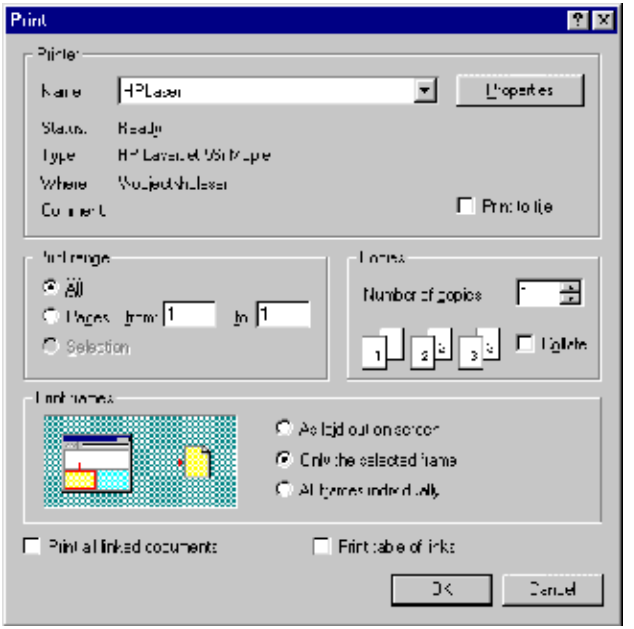
Figure 1 Online Document Displayed in a Netscape Web Browser



Printing from a Web Browser

You can print a copy of this document, one file at a time, from the Web browser. Before you print, make sure that the chapter or appendix you want is displayed and *selected* in your browser.

To select a chapter or appendix, click anywhere inside the chapter or appendix you want to print. If your browser offers a Print Preview feature, you can use the feature to verify which chapter or appendix you are about to print. If your browser offers a Print Frames feature, you can use the feature to select the frame containing the chapter or appendix you want to print. For example:



The BEA MessageQ Online Documentation CD also includes Adobe Acrobat PDF files of all of the online documents. You can use the Adobe Acrobat Reader to print all or a portion of each document. On the CD's main menu, click the Bookshelf button. On the Bookshelf, scroll to the entry for the BEA M3 document you want to print and click the PDF option.

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

Convention	Item

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Example*s: LPT1 SIGNON OR
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

`[ ]`	Indicates optional items in a syntax line. The brackets themselves should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f` *`file-list`*`]...`
	`[-l` *`file-list`*`]...`

`|`	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

Convention	Item

. . .	Indicates one of the following in a command line:
	♦ That an argument can be repeated several times in a command line
	♦ That the statement omits additional optional arguments
	♦ That you can enter additional parameters, values, or other information
	The ellipsis itself should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f file-list]...`
	`[-l file-list]...`

. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

Related Documentation

The following sections list the documentation provided with the MessageQ software, related BEA publications, and other publications related to the technology.

MessageQ Documentation

The MessageQ information set consists of the following documents:

BEA MessageQ Introduction to Message Queuing

BEA MessageQ Programmer's Guide

BEA MessageQ System Messages

BEA MessageQ Client Guide

BEA MessageQ FML Programmer's Guide

BEA MessageQ FML Reference Pages

Note: The BEA MessageQ Online Documentation CD also includes Adobe Acrobat PDF files of all of the online documents. You can use the Adobe Acrobat Reader to print all or a portion of each document.

Contact Information

The following sections provide information about how to obtain support for the documentation and software.

Documentation Support

If you have questions or comments on the documentation, you can contact the BEA Information Engineering Group by e-mail at **docsupport@beasys.com**. (For information about how to contact Customer Support, refer to the following section.)

Customer Support

If you have any questions about this version of ProductName, or if you have problems installing and running ProductName, contact BEA Customer Support through BEA WebSupport at `www.beasys.com`. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

1 Installing MessageQ

This chapter contains the following topics that describe how to install MessageQ software on Windows NT systems:

- Preparing to Install the MessageQ Software
- Installing the MessageQ Software on Windows NT systems
- Performing Postinstallation Tasks

Preparing to Install the MessageQ Software

Before you install MessageQ, there are several installation requirements you should perform to ensure that the installation runs smoothly and the software is installed correctly, as follows:

1. Check the contents of the distribution kit.
2. Comply with the hardware, software, and disk space requirements.
3. Back up your system disk.
4. Choose the type of installation that is applicable for your system (typical, compact or custom).

Checking the Software Distribution Kit

Use the bill of materials (BOM) to check the contents of your software distribution kit. The kit contains the following:

Media

Your kit contains a CD-ROM for industry-standard CD readers, labeled MessageQ for Windows NT, Version 5.0. This CD-ROM contains both Digital Alpha© and Intel® versions.

Hardcopy and Online Documentation

Your kit contains the following hardcopy documentation:

- *Installation and Configuration Guide for Windows NT*
- *Installation and Configuration Guide for UNIX*
- *Release Notes*

The MessageQ for UNIX user documentation set contains the following online manuals:

- *Introduction to Message Queuing*
- *Programmer's Guide*
- *FML Programming Guide*
- *Reference Manual*
- *Installation and Configuration Guide for Windows NT*
- *Installation and Configuration Guide for UNIX*
- *MessageQ Client for Windows User's Guide*
- *MessageQ Client for UNIX User's Guide*
- *System Messages*

As part of your software distribution kit, MessageQ provides this online documentation in HTML format which you can read using any World Wide Web browser. For more information on how to access the online documentation, refer to the section on Using the Online User Documentation.

Complying with the Installation Prerequisites

Before you install MessageQ software you must have access to an account with Administrator privileges on the system where you are installing MessageQ. Also, your system must meet the hardware, software, and disk space requirements listed in this manual.

Backing up the System Disk

We recommend that you back up your system disk before installing any software. For details on performing a system disk backup, see your Windows NT system documentation.

Installing the MessageQ Software on Windows NT Systems

This topic describes the tasks for installing the MessageQ software on Windows NT systems, as follows:

- Choosing Installation Options
- Installing MessageQ on Systems Already Running MessageQ Software
- Installing MessageQ on Systems Not Running MessageQ Software
- Recovering from Installation Errors

The length of a MessageQ installation can vary, depending on the type of media from which you are installing and your system configuration. You can stop the installation procedure any time by clicking Exit or Cancel in the currently active dialog. When you interrupt the installation, files created up to that point are not deleted automatically; you must delete these files manually.

Appendix A lists the files installed on your system during the installation procedure.

Choosing Which Subsets to Install

The MessageQ installation dialogue displays a list of options that let you choose the type of MessageQ installation for your system. If you choose an installation option and then decide you don't want to install it, you can cancel your selection to redisplay the list of available options.

Table 1-1 describes the types of installations available and the subsets installed for each installation type.

Table 1-1 MessageQ Installation Options for Windows NT Systems

Installation Option	Description
Typical	Installs all of the program options. This installation is recommended for most users. The Typical option installs the following subsets: ♦ Server Files ♦ Client Files ♦ Online Documentation
Compact	Installs the minimum required options to run MessageQ. The Compact option installs the following subset: ♦ Client Files
Custom	This installation lets you choose the exact MessageQ options you want installed on your system. You can choose from the following subsets: ♦ Server Files ♦ Client Files ♦ Online Documentation

Installing MessageQ on Systems Already Running MessageQ

We recommend that you install Version 5.0 software in the default installation directory called `C:\Program Files\BEA Systems\BEA MessageQ`. If you have a previous version of MessageQ for Windows NT installed on your system and choose

to install into the same directory, the previous version of the software will be completely overwritten. Regardless of the directory in which the product is installed, the icons for the Program Manager are overwritten to point to the new installation. If a previous version is installed, you will not be able to access it from the Program Manager using the icons.

Note: Before you can run your existing applications under MessageQ Version 5.0, you must convert your group configuration data to use the Version 5.0 format. Refer to the last section in this chapter entitled Performing Postinstallation Tasks for more information on how to convert existing group configuration data.

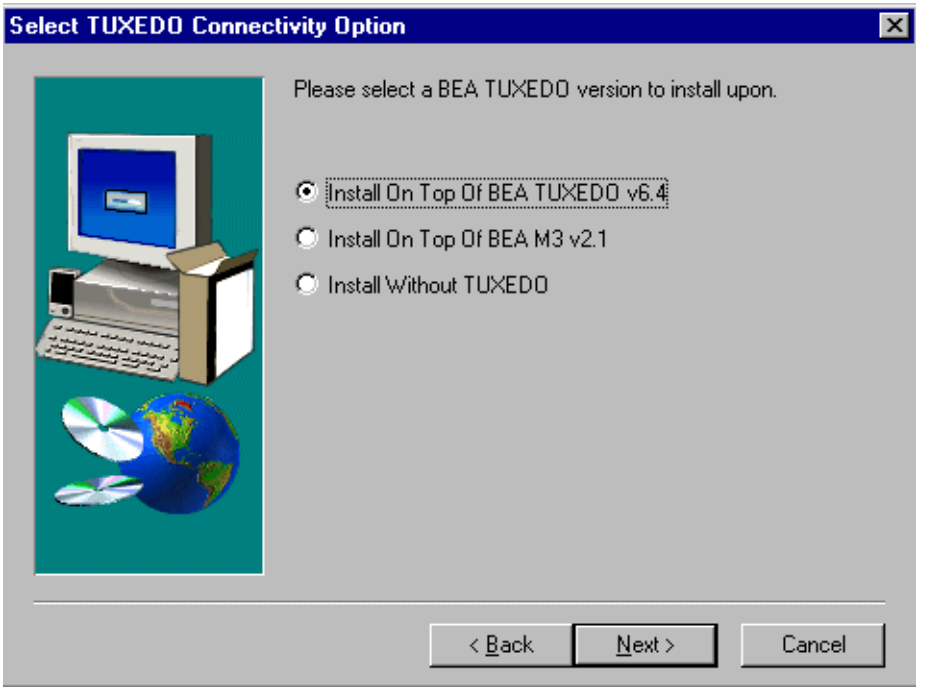
Installing MessageQ for Windows NT

The following procedure describes how to install MessageQ software for Windows NT.

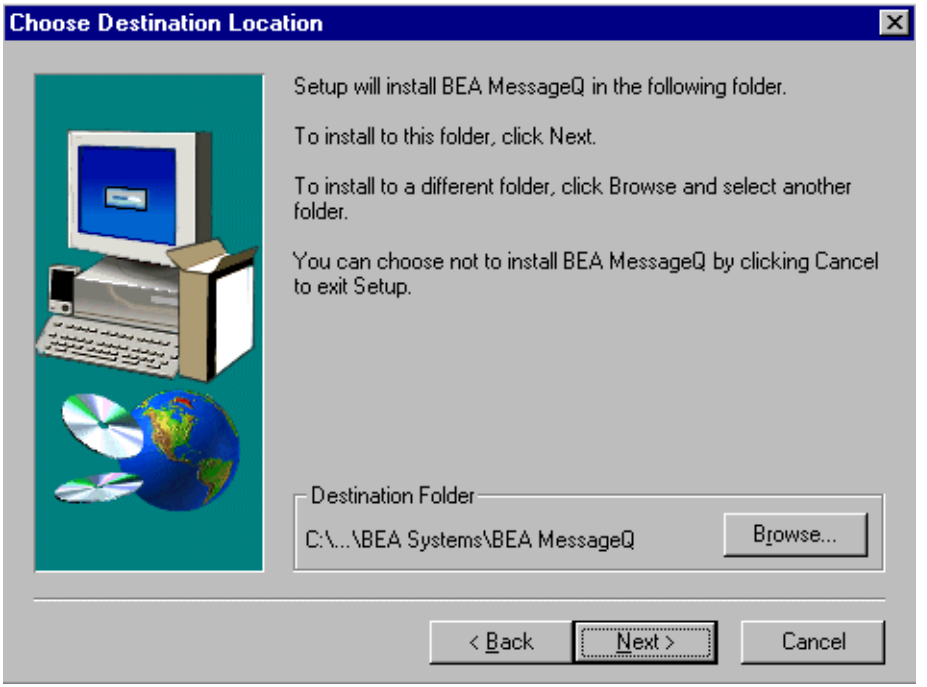
Note: You can stop the installation procedure at any time by clicking Exit in the Installation Options dialog. You can also return to the previous dialog by clicking Back.

1. Log into the Administrator account.
2. Load the MessageQ distribution CD-ROM into the CD reader.
3. Open Windows Explorer or click on the My Computer icon and select the CD drive.
4. Double-click the applicable folder based on your system processor: WinNT for the Intel version and AlphaNT for the Alpha version.
5. Double-click `setup.exe` to start the installation.
6. Click **Next** after the SETUP utility displays the dialog box that welcomes you to the MessageQ software installation.

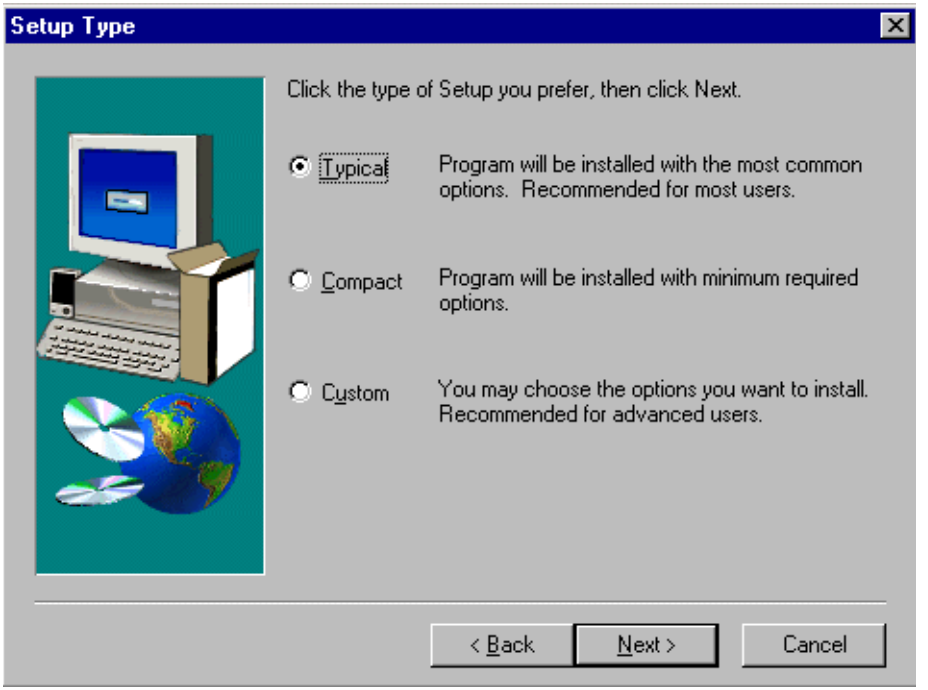
7. If you have BEA TUXEDO or BEA M3 installed on your system, select a BEA TUXEDO or BEA M3 version to install upon and click **Next**. You can also install MessageQ as a standalone product.



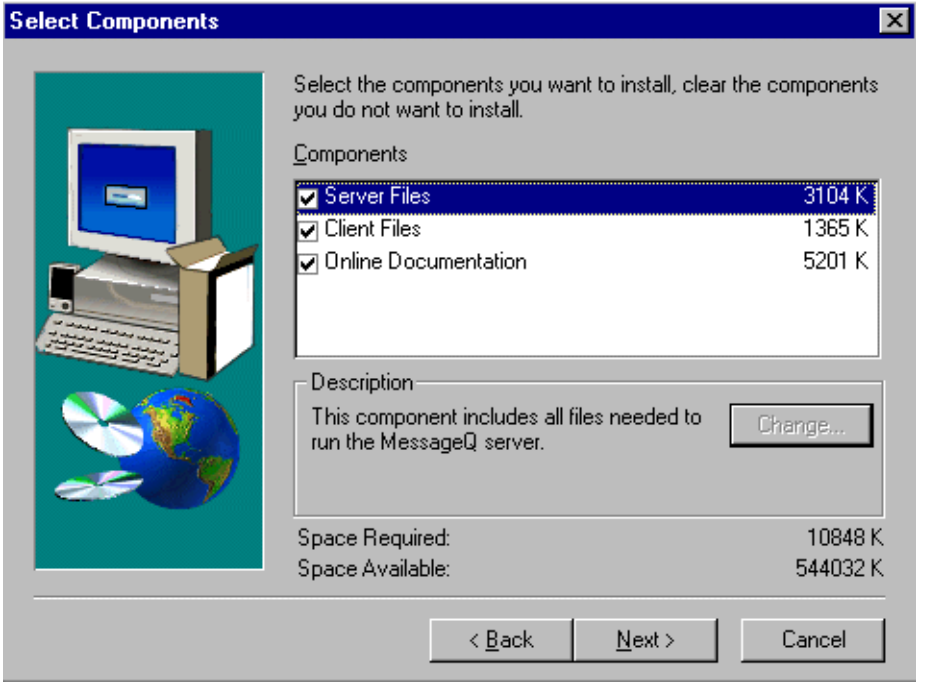
8. Choose the destination location and click **Next**. The default installation for MessageQ is `C:\Program Files\BEA Systems\BEA MessageQ`.



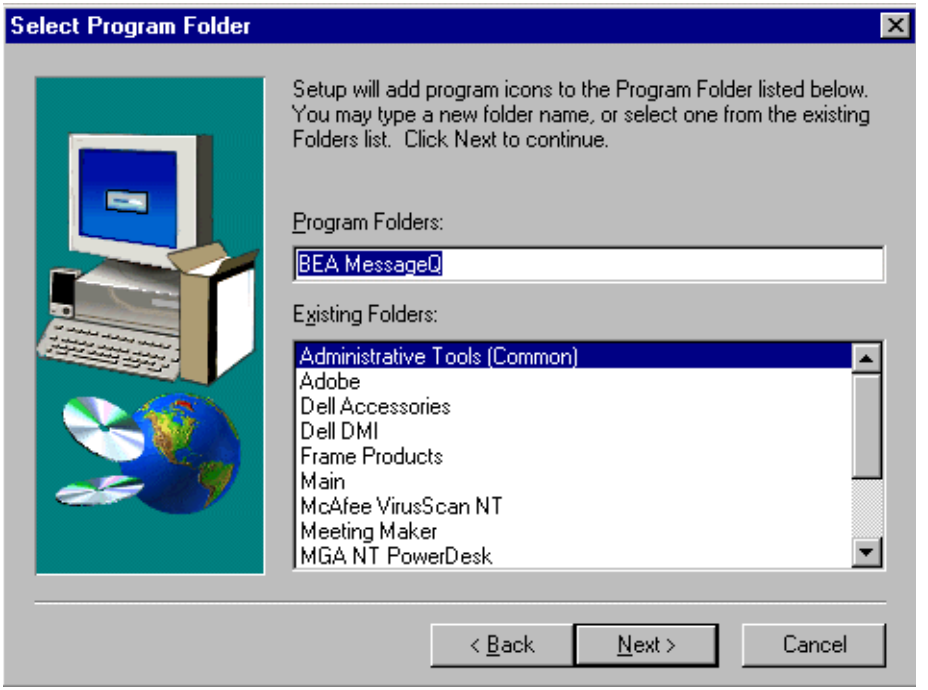
9. Choose between a Typical, Compact, or Custom installation and click **Next**.



10. If you select a Custom installation, you are prompted to select the components you want to install and click **Next**. The Select Components box displays a description of the selected component and shows you information on space required and space available.

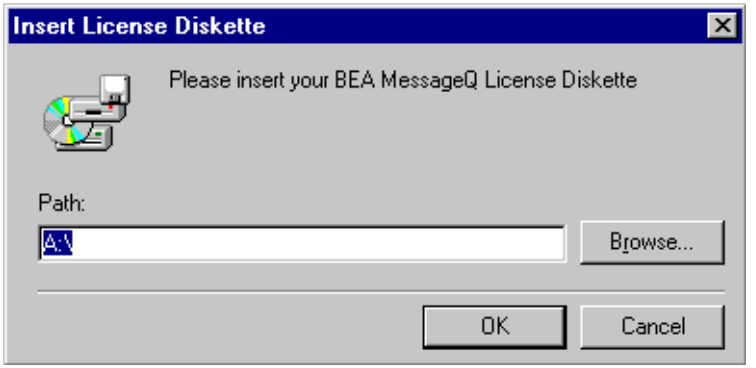


11. Choose the Program Folder name and click **Next**.



12. You are prompted to install the MessageQ license file. You can install the license at a later time, but you can not use MeessageQ until a valid license file has been installed. Click **Yes** to proceed with license installation.

13. Insert the BEA MessageQ License Diskette or browse to a location where the license file has been stored and click **OK**.



14. After the license file has been successfully installed and all files are transferred to your system, click **Finish** to complete the installation

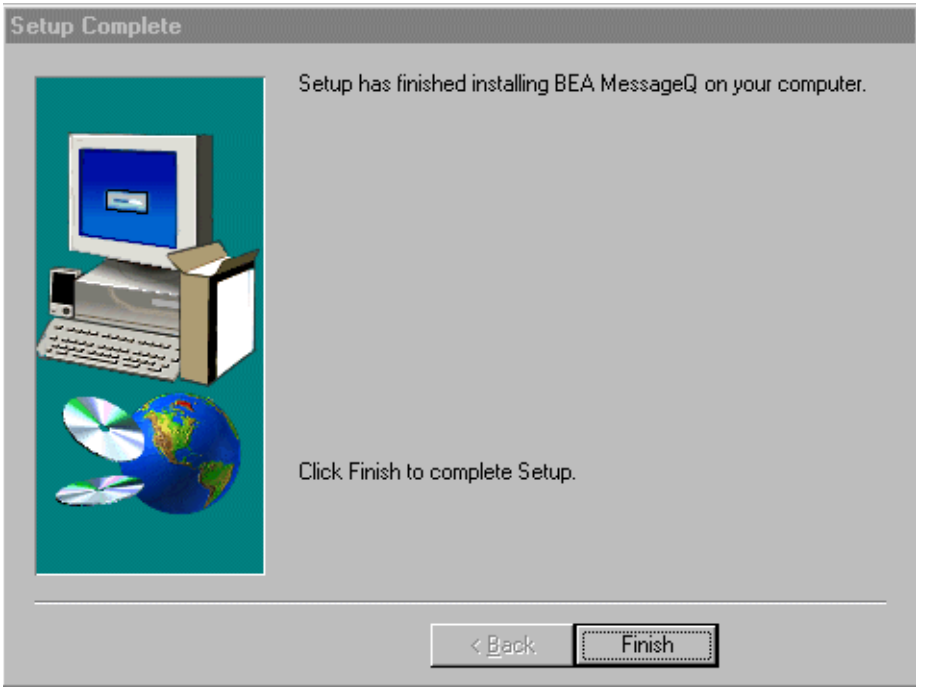


Figure 1-1 shows the MessageQ program group created after a Typical installation is complete.

Figure 1-1 MessageQ Program Group Server Icons

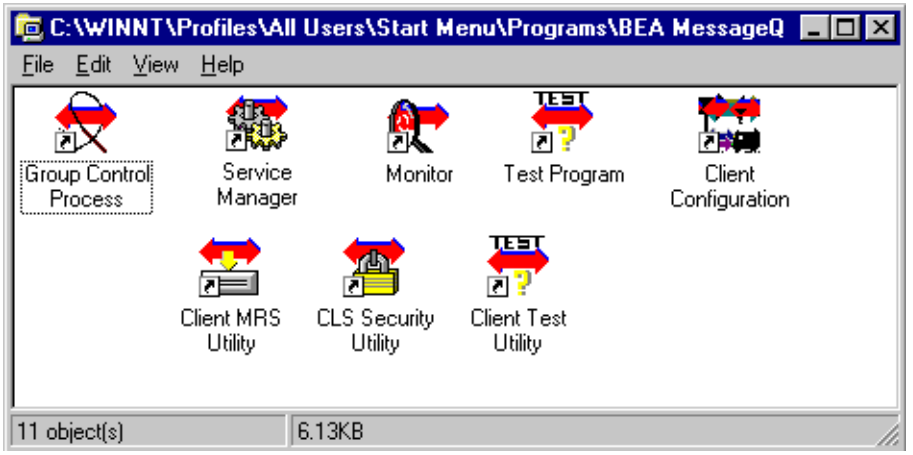


Table 1-2 describes the program group utility icons.

Table 1-2 Utility Group Programs

MessageQ Icon	Description
MessageQ Monitor	Windows-based program that lets you monitor the messaging activity of the MessageQ message queuing buses and groups you have created.
MessageQ Group Control Process	Program that creates a MessageQ for Windows NT message queuing group.
MessageQ Test Program	Windows-based utility that enables you to test message exchanges between the queues you configured. See the *Programmer's Guide* for more information.
MessageQ Loader	Automatically loads the group configuration at group startup, and can be invoked manually to change certain group characteristics while a group is running.

Table 1-2 Utility Group Programs

MessageQ Icon	Description
MessageQ Service Manager	Windows-based utility used to configure the message queuing groups that you would like to be automatically started and stopped by the MessageQ Service. Refer to the section entitled Starting the MessageQ Service for Windows NT for more information.

The CLS Security Utility, Client Configuration, Client MRS Utility, and Client Test Utility are all part of BEA MessageQ Client for Windows and are described in the *BEA MessageQ Client for Windows User's Guide*.

Recovering from Installation Errors

If errors occur during the MessageQ installation procedure for Windows NT, be sure you have met all of the system software and hardware requirements.

If the installation fails, SETUP displays a dialog indicating that the installation was not successful. Installation failures can occur if the following conditions exist:

- The operating system version is not supported by MessageQ software.
- TCP/IP networking software is not installed or configured.
- There is insufficient disk space on the target drive.
- The kit has missing or unreadable files.

If you encounter a problem while using MessageQ and you believe the error is caused by a problem with MessageQ, call BEA technical support. See Customer Support in the Preface for additional information.

Using the Online User Documentation

MessageQ includes online documentation in HTML format that can be read using a World Wide Web browser. To use the HTML-based online documentation, invoke a World Wide Web browser, and use the "Open File" option to open the following file:

drive:*installation directory*\doc\bmq\v5_0\index.htm

Where:

drive:	The letter of the drive (for example, C:, D:, or E:) where you installed MessageQ
installation directory	The directory where you installed MessageQ (for example: \BEA Systems\BEA MessageQ\)

Begin navigating the online documentation using the hyperlinks. Note that each category of information (introduction, administration, programming, etc.) in the online documentation set is accessible on the index.htm page. The first page in each book is the table of contents and the cover page. Hyperlinks can be used to navigate through the information contained within a single book. To view information in a different book, use the navigation bar at the top of the page or click on the BEA Systems logo to jump to the initial page to select another book.

For those sites with a corporate intranet, MessageQ documentation can be copied to a single node and linked into the corporate information base contained on an internal Web site. Copying the online documentation to an intranet server limits the use of disk space to a single system while making it accessible to everyone on the corporate intranet. After the documentation is copied to the corporate Intranet server, users must be given a URL to access the documentation or a description of the page from which the documentation can be accessed.

Performing Postinstallation Tasks

After completing the installation procedure, you can perform the required and optional postinstallation tasks. MessageQ does not require you to perform the postinstallation tasks in a particular order.

The postinstallation tasks are as follows:

Configure the network	Required
Start the MessageQ Service for Windows NT	Optional

Add the MessageQ drive and directory to your Path	Required
Converting Group Configuration Data	Required

Configuring the Network

To configure MessageQ for Windows NT to communicate with remote systems, you must have configured TCP/IP networking and started the TCP/IP service. The remote systems that you intend to communicate with must have their names and Internet Protocol (IP) addresses entered in the `\winnt\system32\drivers\etc\hosts` file. Note that the local system must have an entry in this file as well, if MessageQ is to operate properly. Enter the local system name using uppercase letters.

It is important to note that the node names are case sensitive. Your Windows NT system name, which was defined when Windows NT was installed or when you used the Network applet in the Control Panel, is in uppercase. To use MessageQ for Windows NT to communicate with other networked systems by way of TCP/IP, we recommend that your TCP/IP host name be the same as your Windows NT system name, and should be defined in uppercase letters.

For example, if your Windows NT system is ANTARES, your TCP/IP host name, as defined in the Network applet and the `\winnt\system32\drivers\etc\hosts` file should also be defined as ANTARES. Your Windows NT system name is the name that will be presented to remote MessageQ systems when initiating connections.

Starting the MessageQ Service for Windows NT

MessageQ for Windows NT provides a service that starts buses and groups on demand. The MessageQ Service for Windows NT runs in the background when no one is logged on to the system. By default, the MessageQ Service for Windows NT is not started when it is installed.

To use the MessageQ Service, you must begin by using the MessageQ Service Manager to configure the message queuing groups that you would like to be automatically started and stopped by the MessageQ Service. You can run this utility

by double clicking the MessageQ Service Manager icon in the main MessageQ window. You must also define the environment variable BEADIR as the full path of the MessageQ installation directory.

To start the MessageQ service, select the Services applet in the Control Panel and choosing Start. When the service starts up, it will in turn start all message queuing groups configured for automatic startup. You can also change the default configuration of the MessageQ Service using the Services applet in the Control Panel to enable the MessageQ Service to start automatically when the system boots.

Adding the MessageQ Drive and Directory to Your Path

The MessageQ executable images must be located in the user's default path to operate properly. Add the MessageQ drive and directory to your Path and define the BEADIR environment variable as follows:

1. Open the Control Panel.
2. Double-click the System icon.
3. In the System Properties window, click on the environment tab to edit the Path entry. If no path variable is present, create one by entering "Path" in the Variable field.
4. Click the Value text field and then press the **End** key, or use the right arrow key to scroll to the end of the field.
5. If the Value field is blank, enter the following at the end of the field:

   `drive:\`*`installation directory`*`\bin`

   where:

`drive:`	The letter of the drive (for example, C:, D:, or E:) where you installed MessageQ for Windows NT
`\`*`installation directory`*`\bin`	The directory path where you installed MessageQ for Windows NT. For example: `\Program Files\BEA Systems\BEA MessageQ\bin`

If a path variable is present, do not delete it. Add a semicolon (;) to the end of the existing Path specification and then add the MessageQ Path variable as described above.

6. Click **Set** to set the drive and directory path.
7. If no BEADIR variable is present, create one by entering "BEADIR" in the Variable field.
8. Enter the full installation path for BEA MessageQ in the Value field, for example, `C:\Program Files\BEA Systems\BEA MessageQ\`.
9. Click **OK** to confirm the new settings.
10. Close the Control Panel.
11. Log out of the Windows NT system and then log back in to initialize the new Path and BEADIR environment variable..

Converting Your Group Initialization File to Version 5.0

MessageQ for Windows NT provides a conversion utility that lets you quickly convert your existing group configuration data so that it is compatible with the Version 5.0 software. This section describes how to use the conversion utility called dmqconvert.

MessageQ for Windows NT, Version 5.0 no longer provides the Configuration Editor for configuring message queuing groups. This release now uses an initialization file to store all group configuration data making the Windows NT environment compatible with all other MessageQ Server environments. Due to this change, MessageQ configuration data is no longer stored in the NT Registry. When you convert your existing groups to Version 5.0, the MessageQ conversion utility copies the existing data from the NT Registry to the file that you specify using a common group initialization file format.

Note that the conversion utility *does not* update the group initialization file to include the new Version 5.0 sections and parameters. To update your initialization file to use new Version 5.0 features, refer to the information in this chapter, which describes all the sections and parameters in the group initialization file. Using a text editor, you can add to or change the existing initialization files to incorporate Version 5.0 parameters and their associated settings.

To convert your existing group initialization file, follow these steps:

1. Open an MS-DOS command prompt window.
1. Change directory to the MessageQ installation directory, for example:

   ```
   cd \Program Files\BEA Systems\MessageQ
   ```

2. To convert your initialization file, enter the following command:

   ```
   dmqconvert -b x -g y -f myfile.init
   ```

   Where:

x	The Bus number of the group initialization file you want to convert
y	The Group number of the group initialization file you want to convert
myfile.init	The name you want to give to the new initialization file

2 Configuring MessageQ

This chapter contains the following topics that describe how to configure MessageQ message queuing buses using the Configuration Editor and message queuing groups by editing the group initialization file:

- Overview of MessageQ Configuration
- Starting the Configuration Editor
- Configuring a Message Queuing Bus
- Configuring a Message Queuing Group

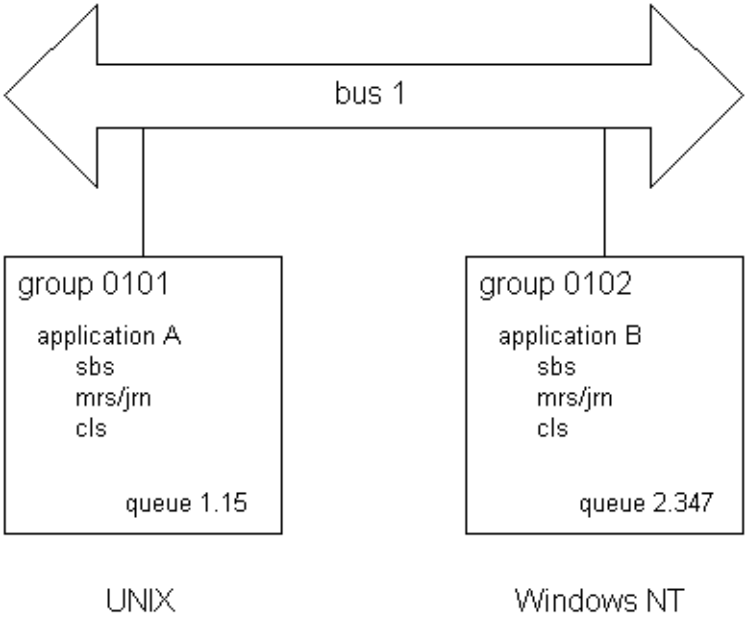
Overview of MessageQ System Configuration

Before applications can exchange information, you must configure the MessageQ environment by defining the default values and parameter settings for the following components:

- Message Queuing Bus
- Message Queuing Groups
- Message Queues
- Cross-group Connections
- Server Processes

Figure 2-1 shows a sample MessageQ configuration. The configuration shows a message queuing bus that contains two message queuing groups, one on a Digital UNIX system and the other on a Windows NT system. Both groups are able to exchange messages using a cross-group connection. Each application is associated with at least one primary message queue and several server processes, such as MRS/JRN, SBS, and CLS.

Figure 2-1 Sample MessageQ Configuration



The following topics describe the standard components of a MessageQ configuration.

Message Queuing Bus

The MessageQ message queuing bus is the backbone of MessageQ because it forms the interprocess communication highway that applications use to send and receive messages. The message queuing bus is made up of one or more message queuing groups that are configured to communicate with each other.

Message Queuing Groups

A MessageQ message queuing group is a collection of related processes that share system resources. A single group must reside on a single system; however, a single system can support multiple groups. Applications that manage common resources or have frequent or high-volume communications with each other should be placed in the same group.

A group connects related applications or components of the same application. Communication between groups on the same system and between applications running in different message queuing groups on different systems requires that a computer network be configured. (See the Configuring the Network topic in Chapter 1 for more information.)

Groups are labeled with a four-digit bus ID and a five-digit group ID. Groups with the same bus ID form a message queuing bus; only groups that share a bus ID can communicate. A Group Control Process automatically creates the resources required for a group using the values and settings in the group initialization file. All MessageQ processes are started by way of the Group Control Process.

Message Queues

A message queue provides an area for an application to store and retrieve messages. Message queues can be thought of as attachment points on the message queuing bus. Each message queue is identified by a unique queue number, which is part of the queue address. Message queues can be temporary or permanent.

If an application needs to use a queue for a short period of time, the queue should be configured as a temporary queue. However, if an application needs to use a queue for an extended period of time (for example, a financial application that tracks the buying and selling of particular stocks), then the queue should be configured as a permanent queue.

A message queue is configured by the application developer and is managed by MessageQ. MessageQ supports three types of message queues: primary, secondary, and multireader. Table 2-1 describes these message queues.

Table 2-1 Types of Message Queues

Queue Type	Description
Primary	Serves as the main mailbox for a MessageQ application. An application is limited to one primary queue, although it may be associated with multiple secondary or multireader queues. When an application reads a message from a primary queue, the message is removed from the queue. Messages are read in first-in/first-out (FIFO) order, unless another order is specified.
Secondary	Serves as an alternate mailbox for an application to receive messages
Multireader	Serves as a central mailbox for multiple applications. This lets more than one application simultaneously attach to and read messages from the same queue.

Primary and secondary queues can be permanent or always writable. A queue that is always writable can store messages when there are no receiver programs attached. A permanent queue cannot store messages unless a receiver program is attached to it. Multireader queues are always writable, and temporary queues are only active when a receiver program is attached.

Applications must be associated with at least one message queue to receive MessageQ messages. For example, a system designer may designate queue 3 in group number 1 to receive temperature readings from a semiconductor furnace. After the bus, group, and queue address have been defined, the sender application must know the queue address to direct messages containing temperature readings. The receiver application must know the queue address from which to read messages about temperature changes in the furnace.

Applications also can send and receive messages from message queues in one or more message queuing groups. The application designer decides how to distribute messaging based on the requirements of the application.

Using Naming

Naming is a MessageQ capability that enables applications to refer to queues by name instead of using their physical address in the MessageQ configuration. Using names separates applications from the specifics of the network environment and enables system managers to make configuration changes without requiring developers to change the applications. Developers use the `pams_locate_q` function to locate the queue address for a queue name at runtime. Currently, the MessageQ naming capability applies only to queue names. There is no ability to assign a name to other MessageQ entities such as groups or buses.

Names can be defined to have a local or global scope. Local names are visible only to applications running in a particular group. Global names are available for use by any application attached to the message queuing bus.

Also, names can be defined using a static or dynamic approach. Static definition means that the definition of the name-to-queue translation is provided in the Global Name Table (%GNT) section (see Defining Name-to-Queue Translations in the Group Name Section for more information) of the MessageQ configuration file for the group. Dynamic definition means that an application has created a name-to-queue translation using the `pams_bind_q` function and that MessageQ must look up the definition that has been created.

The MessageQ process that supports the naming capability is called the Naming Agent. The Naming Agent is responsible for creating entries in the name space that were created by static or dynamic definitions. In addition, the Naming Agent provides the look up capability for name-to-queue translations that are made dynamically at runtime.

When you configure a group, you must decide the message queuing group in which the Naming Agent will run. You can also define a backup Naming Agent group in the event that the first group becomes unavailable. This definition is set in the %NAM section (see “Defining the Naming Agent Process in the Name Section” for more information) of the group initialization file. The group that is assigned to run the Naming Agent starts the Naming Agent process at group startup.

Cross-Group Connections

Communication between groups requires the use of a network protocol. MessageQ for Windows NT supports TCP/IP as a network protocol.

There are two types of cross-group connections: direct and indirect. Groups that share a physical network connection can be configured directly in the %XGROUP section of the group initialization file. Groups that do not share a physical network connection can be configured indirectly in the %ROUTE section of the group initialization file.

See Enabling Network Connections in the Cross-Group Section and Defining the Message Routing Database in the Route Section for instructions on how to configure direct and indirect cross-group connections.

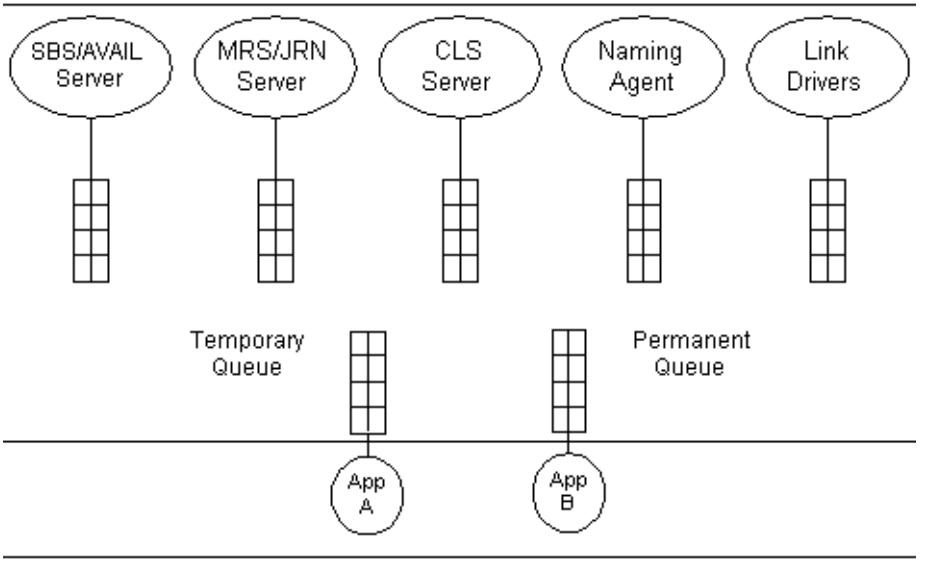
You can also specify timeout intervals for testing cross-group connections. For more information, see the “Defining Timeout Intervals for Link Drivers” topic.

Server Processes

MessageQ for Windows NT provides several server processes that provide key features of MessageQ, such as recoverable messaging, message broadcasting, cross-group communication, and message queuing support for MessageQ Clients.

Figure 2-2 shows the server processes and other components of a MessageQ message queuing group.

Figure 2-2 Components of a MessageQ Message Queuing Group



The server processes can be enabled by setting values in the %PROFILE section of the group initialization file. Table 2-2 describes the server processes that are available to MessageQ for Windows NT.

Table 2-2 MessageQ Server Processes

Server Process	Description
Selective Broadcasting Services (SBS) Server	Controls the broadcasting of data between an application and multiple receiving applications. In addition, the SBS server provides AVAIL/UNAVAIL message-based services.
Message Recovery Services (MRS)/Journal (JRN) Server	Manages the disk storage required to handle recoverable message traffic. The MessageQ message recovery system guarantees message delivery if the system, network, or application fails. Messages designated as recoverable are directed to an MRS server for storage and removed from storage when delivery is confirmed by a user or another MRS server. This server also controls journal management functions such as the handling, opening, closing, and dumping of all journal files.

Table 2-2 MessageQ Server Processes

Server Process	Description
Client Library Server (CLS)	Provides full message queuing support for applications running in a MessageQ Client environment

Configuring a Message Queuing Group

MessageQ for Windows NT Version 5.0 requires users to configure groups by editing a group initialization file using a text editor. This approach is now used by all MessageQ server implementations and makes group configuration data portable among all MessageQ server platforms.

Note: The Configuration Editor utility that previously shipped with the MessageQ for Windows NT product is now obsolete. In Version 5.0 configuration data is no longer stored in the NT Registry. Instead it is maintained in a group initialization file for each message queuing group. If you have existing configuration data that was created using the Configuration Editor utility, you need to convert the data to a MessageQ Version 5.0 group initialization file. Refer to the last section of Chapter 1 entitled Performing Postinstallation Tasks for information on how to use the MessageQ conversion utility.

To create a group initialization file for each group that you need to configure, make a copy of the template file called `group.init` and copy it to the working directory for the group. The working directory is the directory in which your application or applications are designed to execute. You can use the standard settings in the template file as the configuration data for your group, or you can change them as described in this section. You may want to add comments on your specific configuration. Use the exclamation, pound sign, semicolon, or asterisk characters (!, #, ;, or *) to as comment characters.

When you run the `dmqstartup` procedure, MessageQ uses the settings contained in the group initialization file to set the characteristics of the group it creates. The `group.init` template can be found in the following location:

```
Program Files\BEA Systems\BEA MessageQ\TEMPLATES\group.init
```

Table 2-3 gives an overview of the sections of the group initialization file.

Table 2-3 Sections of the Group Initialization File

Section Header	Name	Description
%VERSION	Version	Identifies the group initialization file version. See the Default Values for Parameters topic for more information.
%PROFILE	Profile	Defines the basic parameters of a group. See the Defining Parameters in the Group Initialization File topic for more information.
%XGROUP	Cross-Group Communications	Enables groups on nodes that share a network link to directly exchange messages. This section should have an entry for the node on which the group is running. See the Enabling Network Connections in the Cross-Group Section topic for more information.
%ROUTE	Message Routing	Enables groups on nodes that do not share a network link to indirectly exchange messages. See the Defining the Message Routing Database in the Route Section topic for more information.
%CLS	Client Library Server	Enables the support of multiple MessageQ Client connections to a single NT process using the TCP/IP transport. See the Configuring the Client Library Server in the CLS Section topic for more information.
%BUFFER	Buffer Pool Configuration Table	Applies to OpenVMS systems only.
%QCT	Queue Configuration Table	Defines the permanent queues. See the Defining Queues in the Queue Configuration Section topic for more information.

Table 2-3 Sections of the Group Initialization File

Section Header	Name	Description
%SBS	Selective Broadcast Services (SBS)	On OpenVMS systems, this section of the initialization file controls the operation of the local SBS server. On MessageQ for UNIX and Windows NT, the local SBS server is enabled by setting the parameter value to YES in the %PROFILE section. The %SBS section is empty. See the Setting Group Characteristics in the Profile Section topic for more information.
%MRS	Message Recovery System (MRS)	Controls the operation of the local MRS server. See the Setting Message Recovery System Parameters in the MRS Section topic for more information.
%GNT	Group Name Table	Provides generic queue names (that do not necessarily appear in the %QCT) and their addresses. See the Defining Name-to-Queue Translations in the Group Name Section topic for more information.
%NAM	Naming Agent	Defines the group in which the naming agent process is running. You can define two naming agents for the environment. The naming agent provides the lookup capability for global (bus-wide) names. See the Configuring Global Naming topic for more information.

Defining Parameters in the Group Initialization File

The MessageQ group initialization file is a ASCII text file that can be created and modified using any text editor. You define parameters in the group initialization file by providing numeric values within a specified range or enabling settings to be YES

or NO. To create a group initialization file for each message queuing group, make a copy of the template file, copy it to the group working directory and edit the copy. Do not edit the sample template file that is distributed with the MessageQ kit.

MessageQ also lets you modify a subset of parameters in the group initialization file at runtime. You can use the MessageQ Loader utility to dynamically reload the group initialization file without having to stop and restart MessageQ. Refer to the section Changing Group Characteristics at Runtime in Chapter 3 for a list of the parameters that you can change for a running group. Several of the modifiable parameters can only be changed if certain conditions exist. These conditions are noted in the parameter description.

Note that Windows NT services cannot access mapped drives when using the default System Account. To allow access to mapped drives, modify the service entry using the Control Panel Services applet to specify "This Account" and a "Password". The account and password must have privileges on the mapped drive.

The following topics describe the parameters, values, and settings in the group initialization file.

Default Values for Parameters

The `group.init` file template file contains most of the common parameters that you will need to configure. The following sections of this guide describe all available parameters that you can set using this file. The default values for all parameters are set at group startup, regardless of whether the parameter appears in the group initialization file. If a parameter is not specified, or is specified as a "." or -1, then the paramater takes on the default value.

Some sections of the sample `group.init` file are commented out because they are not used by default. To use these sections, you must uncomment these lines and set the values as appropriate for your environment.

Identifying the Group Initialization File Version

The %VERSION (file version) section consists of a single line that identifies the group initialization file version. (Note that this is the initialization file version, and is not equivalent to the product version number.) MessageQ for NT systems accepts group

initialization files with version number 4.0 only. MessageQ uses the version number to be able to correctly parse configuration information for the message queuing group at startup.

Listing 2-1 shows a sample %VERSION section.

Listing 2-1 Sample %VERSION Section

```
%VERSION 4.0
```

Setting Group Characteristics in the Profile Section

The %PROFILE section configures the primary characteristics (default values and parameter settings) of a group. This section is also enables MessageQ Server processes such as MRS/JRN and SBS. Each server process is dedicated to a specific task or set of tasks.

You can start the following server processes by indicating "YES" for each server entry in the %PROFILE section:

- MRS Server—Enables recoverable messaging.and journaling services for postconfirmation and dead letter journals.
- SBS Server—Enables message broadcasting and AVAIL/UNAVAIL notification.

Listing 2-2 shows a sample %PROFILE section:

Listing 2-2 Sample %PROFILE Section

```
%PROFILE
!
ACCEPT_KILL_COMMAND.......YES.....! accept kill commands from DMQ monitor
ENABLE_XGROUP             NO      ! allow cross-group communications
XGROUP_VERIFY             NO      ! do not verify cross group links
FIRST_TEMP_QUEUE          200     ! minimum is 101, maximum is 850
```

```
XGROUP_TABLE_SIZE          20      ! size of the group link table
GROUP_BYTE_QUOTA           8388608 ! maximum number of bytes on all queues
GROUP_MAX_MESSAGE_SIZE     32000   ! largest message size permitted in this group
ATTACH_TMO                 600     ! timeout in seconds for locates

ENABLE_MRS                 NO      ! start recoverable messaging servers
ENABLE_JRN                 YES     ! enable PCJ journaling when MRS is enabled
ENABLE_SBS                 YES     ! start selective broadcast and AVAIL/UNAVAIL

DEFAULT_NAMESPACE_PATH   /u/mydir ! default namespace path for naming agent

%EOS
```

Table 2-4 describes the parameters of the %PROFILE section.

Table 2-4 Profile Parameters

Parameter	Range	Default	Description
ACCEPT_KILL_COMMAND	YES NO	YES	Controls group control process termination requests from the MessageQ Monitor utility. A NO setting means that any message requests to shut down group control processes are ignored. **Note**: This parameter can be modified at runtime.
ENABLE_XGROUP	YES NO	NO	Enables MessageQ TCP/IP network access. A YES setting causes the appropriate link drivers to start up, to perform cross-group communications.
XGROUP_VERIFY	YES NO	NO	Verifies incoming TCP/IP connections. A YES setting causes all connections to be validated against the cross-group connection table, much like an Access Control List (ACL).

Table 2-4 Profile Parameters

Parameter	Range	Default	Description
FIRST_TEMP_QUEUE	101–3950	200	Selects the starting number of the pool of temporary queue pool. This defines the size of the permanent and temporary queue pools for a group.
XGROUP_TABLE_SIZE	1–32,000	20	Selects maximum number of group entries. This parameter defines the maximum number of group entries that MessageQ tracks, and does not affect the maximum group address.
GROUP_BYTE_QUOTA	1,048,576–Unlimited	8,388,608	Sets the maximum number of message bytes on queues in memory. This includes user queues and all internal queues (such as recoverable messages that have not been written to a journal yet). The primary purpose of this parameter is to prevent the group server from becoming excessively large. **Note:** Although the range for this parameter is unlimited, you should not specify a value that is more than 30% of the virtual memory configured for the system.
GROUP_MAX_USER_QUEUE	101-3999	999	Sets the highest user queue number
GROUP_MAX_MESSAGE_SIZE	1-4194304	32,000	Specifies the largest message size permitted in this group
ATTACH_TMO	>0	600	Selects the maximum elapsed time that an attach queue operation can take before a status of `PAMS_TIMEOUT` is returned to the caller. This number is in units of seconds. **Note:** This parameter can be modified at runtime.
ENABLE_MRS	YES NO	NO	Enables Message Recovery Services

Table 2-4 Profile Parameters

Parameter	Range	Default	Description
ENABLE_JRN	YES NO	YES	Enables PCJ journaling when MRS is enabled. If `ENABLE_JRN` is set to YES, messages are written to the PCJ as a function of the `force_j` argument value in the `pams_confirm_msg` call. If `ENABLE_JRN` is set to NO, messages are not written to the PCJ regardless of the `force_j` argument value in the `pams_confirm_msg` call. The default journaling action is to not write messages to the PCJ.
ENABLE_SBS	YES NO	YES	Enables Selective Broadcast Services and AVAIL/UNAVAIL services
DEFAULT_NAMESPACE_PATH	—	—	Default namespace pathname for naming agent. Refer to the section entitled Configuring Global Naming for more information. **Note:** This parameter can be modified at runtime.

If the group you have defined will use resources such as message recovery services, CLS, group connections, permanent queues, or name-to-queue translations, you must define them next.

Setting Message Recovery System Parameters in the MRS Section

The %MRS section lets you enable MessageQ Message Recovery Services (MRS), which provide a mechanism for guaranteed message delivery by storing messages on disk and automatically attempting redelivery until the message is received by the target system. On MessageQ for NT systems, message recovery is provided by the `dmqjourn` journal process.

The MessageQ startup procedure starts up the MRS Server only if the ENABLE_MRS parameter is set to YES in the Profile section. In addition, setting the ENABLE_JRN parameter in the Profile section to YES enables message recovery services to log messages to the auxiliary journals called the postconfirmation journal and the dead letter journal. For a complete description of MRS, refer to the *Programmer's Guide*.

MessageQ provides the following types of recovery journals:

Journal Type	Description
Store and Forward (SAF)	Used to store recoverable messages targeted at a remote group on the local system. Messages are automatically re-sent after communication with the target group is restored.
Destination Queue File (DQF)	Used to store recoverable messages targeted at the local group on the local system. Messages are automatically re-sent when the queue becomes active.
Postconfirmation Journal (PCJ)	Used to store successfully confirmed recoverable messages. The PCJ forms an audit trial of message exchange that can be read or printed. The default journaling action is to not write messages to the PCJ.
Dead Letter Journal (DLJ)	Used to store messages that could not be stored for automatic recovery by MRS. Undelivered DLJ messages can be re-sent under user or application control.

When configuring the path for journal files on Windows NT systems, note that Windows NT services cannot access mapped drives when using the default System Account. Modify the service entry using the Control Panel Services applet to specify "This Account" and a "Password". The account and password must have privileges on the mapped drive.

Listing 2-3 shows a sample %MRS section and Table 2-5 describes the parameters of the %MRS section.

Listing 2-3 Sample %MRS Section

```
%MRS
```

```
CACHE_PERCENTAGE         90                ! %rcv msg quota for MRS msgs
MRS_JOURNAL_PATH              C:\my\rj_path   ! recovery journal path
MRS_POST_CONFIRMATION_PATH    C:\my\pcj_path  ! post confirmation journal path
MRS_DEAD_LETTER_PATH          C:\my\dlj_path  ! dead-letter journal path
MAX_CACHE_BYTES               4194304       ! maximum number of unwritten bytes
%EOS
```

Table 2-5 %MRS Parameters

Parameter	Range	Default	Description
MRS_JOURNAL_PATH	—	—	If a recoverable message cannot be delivered to the target application, the journal process writes it to the recoverable journal file. Journal files are dynamically created and deleted as needed by the journal process. The size of the journal file is fixed. The journal process creates a new journal file each time the current file fills up and attempts to empty journal files by periodically attempting to reach the corresponding remote queue.

Table 2-5 %MRS Parameters

Parameter	Range	Default	Description
CACHE_PERCENTAGE	1 – 100	50	Specifies the maximum number of recoverable messages that can appear in a queue at one time. We recommend that you always set this parameter to under 100. This parameter operates as follows: ♦ When a message queue is attached, enough recoverable messages are delivered to match this limit. ♦ When the limit is reached, no more recoverable messages are delivered until enough recoverable messages are dequeued by the target to drop below the set limit. ♦ While recoverable message delivery is stopped due to the limit, new recoverable messages are written to the DQF file to be delivered in order of receipt. Non-recoverable messages are delivered as usual.
MRS_POST_CONFIRMATION_PATH	—	—	Recoverable messages that successfully reach the delivery interest point can be written to the postconfirmation journal of the target group. If no path is specified, this parameter defaults to the same setting as MRS_JOURNAL_PATH.
MRS_DEAD_LETTER_PATH	—	—	Recoverable messages that cannot be stored by the message recovery system are written to the dead letter journal of the sender program's group. The DLJ file provides a backup mechanism for recovering messages that could not be stored for automatic recovery. If no path is specified, this parameter defaults to the same setting as MRS_JOURNAL_PATH.

Table 2-5 %MRS Parameters

Parameter	Range	Default	Description
MAX_CACHE_BYTES	1,048,576–Unlimited	4,194,304	Sets the maximum number of bytes that have not yet been written to disk. A large value improves the performance of recoverable messaging, particularly for PCJ and SAF journals. There is no maximum value for this parameter. **Note:** Although the range for this parameter is unlimited, you should not specify a value that is more than 30% of the virtual memory configured for the system.

Configuring the Client Library Server in the CLS Section

The %CLS section lets you configure the MessageQ Client Library Server. The CLS provides MessageQ applications running on a supported MessageQ client system with access to message queuing. Applications can communicate with other distributed applications anywhere in the MessageQ network using a single connection to a CLS process connected to the bus. For detailed information about the CLS, refer to the *MessageQ Client for Windows NT User's Guide*.

Listing 2-4 shows a sample %CLS section.

Listing 2-4 Sample %CLS Section

```
%CLS

!

!                              Max      Security
!   Endpoint     Transport     Clients  File

!    12345        TCPIP         25       C:\my\dmqclsec.txt
!    64365        TCPIP         50

%EOS
```

Table 2-6 describes the parameters of the %CLS section:

Table 2-6 %CLS Parameters

Parameter	Range	Default	Description
ENDPOINT	TCP/IP: 1024 – 65535	—	Identifies the TCP/IP port number that the server uses to accept incoming connections. The same endpoint is used in the configuration of MessageQ Clients to locate the CLS. For TCP/IP transport, port numbers less than 1024 are reserved. There is no restriction on the use of port numbers within the available range by CLS. However, you should select port numbers that do not conflict with port numbers used by MessageQ TCP/IP Link Drivers or other TCP/IP-based applications on your system.
TRANSPORT	TCPIP	—	The name of the network protocol stack to be used for the server. For MessageQ for NT systems, this value is TCPIP.
MAX CLIENTS	1 – 512	32	The maximum number of clients the server will serve at any given time. This parameter applies to OpenVMS systems only. **Note:** This parameter can be modified at runtime, but the CLS must be stopped.
SECURITY FILE	—	—	The full pathname of the security file. If no security file is specified, security is not enabled. The MessageQ kit provides a template file called dmqclsec.txt. To configure security, copy this file to your group working directory and modify it as appropriate for your environment. For more information about the CLS security file, see the Restricting Remote Access to CLS topic in Chapter 3. **Note:** This parameter can be modified at runtime, but no process can be attached to the queue.

Enabling Network Connections in the Cross-Group Section

To enable message queuing between different systems in a network, you must create MessageQ message queuing groups on each system and establish cross-group connections between them. Messages can be exchanged between groups using two methods:

To Exchange Messages...	Between...	Use...
Directly	Groups that share a physical network link	Cross-group connections (%XGROUP section)
Indirectly	Groups that do not share a physical network link	Message routing (%ROUTE section)

Your configuration will probably use both methods, so it is important to understand that if a group is defined in the %XGROUP section, it cannot be designated as a target group in the %ROUTE section (as described in the Defining the Message Routing Database in the Route Section topic). It can, however, be designated as a route-through group.

The %XGROUP section lets you define cross-group communication connections among groups that share a physical network link. Each line in the %XGROUP section defines a single cross-group connection, which creates two new link drivers for each unique group number. The %XGROUP section provides the link drivers with the information needed to connect to other MessageQ message queuing groups through TCP/IP networks.

- All parameters for a given connection must appear on the same line.
- You must provide an entry for the local (current) group.
- Changing the node/host name in the `group.init` file then running the Loader utility results in an additional entry at the front of the %XGROUP table rather than a replacement entry. In this case, there is link configuration information in the running group that does not match the `group.init` file and that is also not accessible from any monitor display.

Listing 2-5 Sample %XGROUP Section

```
%XGROUP

!

!Group Group  Node/ Init Thresh- Buffer Recon-    Window      Trans-  End-
!Name  Number Host       old     Pool   nect   Delay  Size    port    point


GROUP1  1    delius  Y  2000000   .     30     10     25000   TCPIP   10001
GROUP2  2    bizet   N  4000000   .     30     10     25000   TCPIP   10002
GROUP3  3    bartok  N  8000000   .     30     10     25000   TCPIP   10003
GROUP4  4    boehm   N  4000000   .     30     10       250   TCPIP   10004
GROUP4  4    rameau  N  4000000   .     30     10       250   TCPIP   10004


%EOS
```

Table 2-7 describes the parameters of the %XGROUP section.

Table 2-7 %XGROUP Parameters

Parameter	Range	Default	Description
GROUP NAME	—	—	Name by which the remote MessageQ group is known to the local group
GROUP NUMBER	1-32000	—	MessageQ Group Number of the remote group
NODE/HOST	—	—	Name by which the remote node is known to the local node
INITIATE	Y N D (Disable)	—	Logical value indicating whether connections to this node should be initiated or disabled. Y — The Group Control Process initiates a connection to the node. N — The Group Link Table and Group Link Vector Table entries are built for the node; however, a connection is not initiated. D — All connections to the node are disabled.

Table 2-7 %XGROUP Parameters

Parameter	Range	Default	Description
THRESHOLD	>1,048,576	1,048,576	This setting is ignored on MessageQ for UNIX and Windows NT systems. It has been superseded by the GROUP_BYTE_QUOTA parameter in the %PROFILE section.
BUFFER POOL	—	—	This value is ignored on MessageQ for UNIX and Windows NT systems.
RECONNECT	>=1	60	Interval, in seconds, between reconnect attempts when this cross-group link is not connected
WINDOW DELAY	>=0	10	Delay, in seconds, that a sender must wait before using a new window when the receiver is congested
WINDOW SIZE	>0	250	Maximum number of messages a group can send to another group before requesting permission to send more. **Note:** This parameter can be modified at runtime.
TRANSPORT	TCPIP		Network protocol stack used. For MessageQ V5.0 this is TCPIP.
ENDPOINT	TCP/IP: 1024 – 32767	—	Address (port number) of the remote link listener process. For TCP/IP transport, this address is an Internet port number ranging from 1024 to 32767, inclusive.

Defining the Message Routing Database in the Route Section

MessageQ uses routing entries to allow messaging between groups that are not directly attached to the local group by way of a physical network link. MessageQ software allows an arbitrary number of intermediate MessageQ groups between the message source and the target queue. The intermediate groups are specified in the %ROUTE section of the group initialization file. The %ROUTE section specifies the next hop (next group to route through) for a message to reach the target destination queue. Groups defined in the %XGROUP section do not need to be defined in the %ROUTE section, and vice versa.

The %ROUTE section lets you define the MessageQ static routing database. The routing database determines where to send messages that are addressed to non-adjacent nodes; that is, nodes that do not appear in the %XGROUP section. All parameters for a given routing entry must appear on the same line.

Listing 2-6 shows a sample %ROUTE section.

Listing 2-6 Sample %ROUTE Section

```
%ROUTE

!Target     Routethrough
!Group      Group

  7         1
  10        2

%EOS
```

Table 2-8 describes the parameters of the %ROUTE section.

Table 2-8 %ROUTE Parameters

Parameter	Range	Default	Description
TARGET_GROUP	1-32,000	—	Group for which traffic is being routed
ROUTE_THROUGH_GROUP	1-32,000	—	Group to which traffic for the target group will be routed

Note: If a group is defined in the %XGROUP section, it cannot be designated as a target group in the %ROUTE section. It can, however, be designated as a route-through group.

In Listing 2-7, the %XGROUP and %ROUTE sections are correctly defined. Group2 and Group7 are not defined as target groups in the %ROUTE section, and they are correctly defined as route-through groups.

Listing 2-7 Correctly Defined %XGROUP and %ROUTE Sections

```
%XGROUP
!

!Group Group  Node/ Init Thresh- Buffer Recon- Window Window  Trans-  End-
!Name  Number Host       old     Pool   nect   Delay  Size    port    point


GROUP2  2     node2  Y    5       75     10    10     250     TCPIP   10002
GROUP7  7     node7  N    5       75     10    10     250     TCPIP   10007



%EOS

!
!Target    Routethrough

!Group     Group

   10        7
   1         2

%EOS
```

In Listing 2-8, the %ROUTE section is incorrectly defined. Group1 cannot be designated as a target group in the %ROUTE section because it was previously defined as an adjacent group in the %XGROUP section. However, Group2 is correctly designated as a route-through group.

Listing 2-8 Incorrectly Defined %XGROUP and %ROUTE Sections

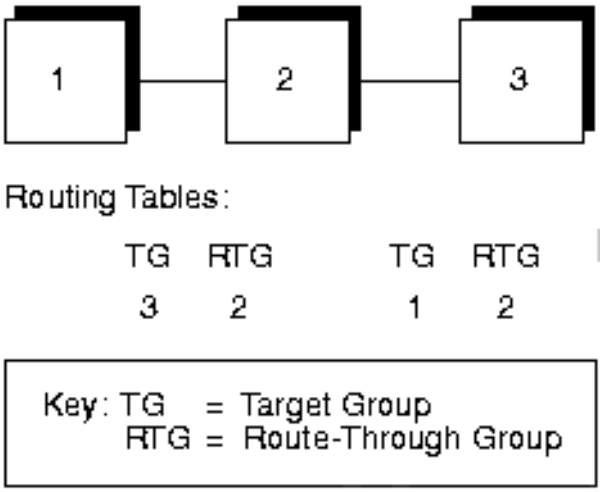
```
%XGROUP

!Group Group  Node/ Init Thresh- Buffer Recon- Window Window  Trans-  End-
!Name  Number Host       old     Pool   nect   Delay  Size    port    point

GROUP1  001   node1  Y    5      75     10     10     250     TCPIP   10001
GROUP2  002   node   N    5      75     10     10     250     TCPIP   10002
%EOS
%ROUTE
!
!Target    Routethrough

!Group     Group

   1         2
   7         2

%EOS
```

Route Discovery

The user-specified %ROUTE section is automatically updated by the process of route discovery. As a message hops along the message route, the MessageQ link drivers automatically load the return address into the routing table so that a message can be returned. Route discovery has the advantage of allowing groups to acquire information about other groups with which they might communicate.

Figure provides an example of how route discovery allows a group to acquire routing information about other groups.

Figure 2-3 Route Discovery



In the example, Group 2 is connected to Groups 1 and 3, and the %ROUTE for Group 1 section was loaded at startup. When Group 1 sends a message to Group 3, the link driver loads the return address of the message into the %ROUTE section for Group 3. Through route discovery, Group 3 is now able to send a message to Group 1.

Visit Counts

Messages with circular routing paths could be endlessly passed from system to system. To prevent this, full message routing uses a visit count, which is a system-determined limit on the number of groups through which a message can hop. The visit count is placed into the message header when the message is created. As a message is written to the link sender, the visit count is decremented. When the visit count reaches zero, the message can no longer be forwarded to another system and the Undeliverable Message Action (UMA) for the message is taken.

Full message routing rules followed by the MessageQ server are as follows:

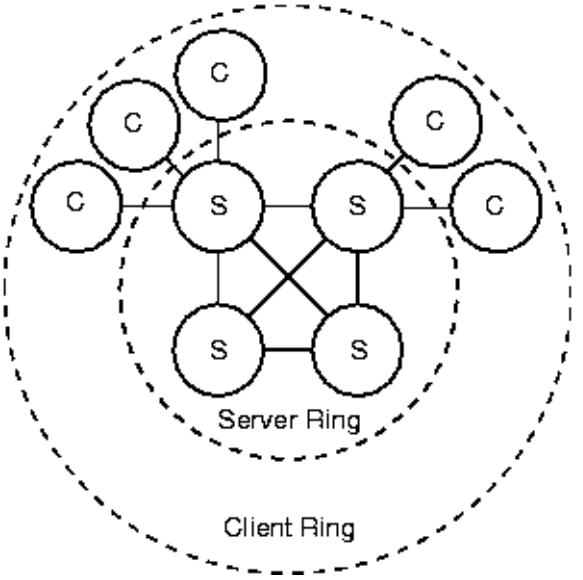
- The initial visit count is 16.
- If the visit count has been exceeded, the UMA is taken.
- If there is a direct connection to the group, the MessageQ server ignores the routing table and uses the direct connection.
- If the group has a known route-through group, the MessageQ server uses it.

- If there is no known route to the target group, the UMA is taken.

Client/Server Message Routing

Full message routing is based on the model of a server group ring surrounded by a client ring, as shown in Figure 2-4. Each server can route messages to other groups without requiring clients to be directly connected to the target group of the message.

Figure 2-4 Client and Server Full Message Routing



In full message routing, a client group has cross-group entries for itself and the adjacent server group. A client group can attach to different server groups, one at a time. The server groups subsequently update the routing table by using route discovery.

A server group has entries in the %XGROUP section of the initialization file for itself and all other server groups. A server group does not have entries for clients; instead, it uses route discovery to learn the return route from server back to client.

Defining Queues in the Queue Configuration Section

The %QCT section lets you define the permanent queues. Each line in the %QCT section defines a single queue. All parameters for a given queue must appear on the same line. Parameters that do not apply must be given a placeholder value.

Note: On VMS systems, queues numbered 90 to 100 and 150 to 199 are reserved exclusively for MessageQ. On NT systems, queues numbered 90 to 100 are reserved exclusively for MessageQ.

Listing 2-9 shows a sample %QCT section.

Listing 2-9 Sample %QCT Section

```
%QCT
!

!Queue  Queue  Byte   Msg   Quota UCB  Queue Owner Conf  Perm   Name   Security
!Name   Number Quota  Quota Enbl  Send Type  Queue Style Active Scope

TEMPLATE  0    262144  .    NONE   .   P      0    II    N      L     N
QUEUE1    1      .     .    NONE   .   P      0    II    N      L     N
QUEUE2    2      .     .    NONE   .   P      0    II    N      L     N
QUEUE3    3      .     .    NONE   .   P      0    II    N      L     N
QUEUE4    4      .     .    ALL    .   M      0    II    Y      L     N
QUEUE5    5      .     .    .      .   S      3    II    Y      L     N
QUEUE6    6      .     .    .      .   S      3    II    Y      L     N
QUEUE7    7      .     .    .      .   S      0    II    N      L     N


! queues that specify ACL security files exist and should be
! checked each time a process attaches one of these queues
QUEUE10    10    .       .   .      .   P    0    II    N      L     Y
QUEUE11    11    .       .   MSG    .   S    10   II    N      L     Y
QUEUE12    12    .       .   BYTE   .   S    10   II    N      L     Y


! queues with differing quotas. Quotas also can be turned on using
! the monitor programs (dmqmonc and dmqmonm)
QUEUE13    13     8192   32  ALL    .   P    0    II    N      L     N
QUEUE14    14   131072   64  MSG    .   P    0    II    N      L     N
QUEUE15    15    32768   .   BYTE   .   .    0    II    N      L     N
```

```
! queues that require explicit confirmation and allow out of
! order confirmation
QUEUE16    16            .   None    .   P    0    EI    N      L     N
QUEUE17    17            .   .       .   P    0    EO    N      L     N


! the example_q_n queues are used by the example programs
! provided in the MessageQ Example Programs subset
example q_1 193          .   .       .   P    0      .   N      L     N

example q_2 194          .   .       .   P    0      .   N      L     N

example q_3 195          .   .       .   P    0      .   N      L     N

example q_4 196          .   .       .   P    0      .   N      L     N

example q_5 197          .   .       .   P    0      .   N      L     N


! The DEAD_LETTER_QUEUE is the target for UMA modes.of
! xx_DLQ A user process may attach this queue to collect un-
! deliverable messages for application-specific processing
DEAD_LETTER_QUEUE 96 64000 500 None .   P    0    II    Y      L     N


%EOS
```

Table 2-9describes the parameters of the %QCT section.

Table 2-9 %QCT Parameters

Parameter	Range	Default	Description
QUEUE NAME	1 – 255 characters	—	MessageQ internal name for each permanent queue. You can create a TEMPLATE QUEUE (queue 0), allowing you to create a set of customized default values for your environment. Specifying a -1 or "." as the value when configuring additional queues enables them to automatically use the TEMPLATE QUEUE settings.

Table 2-9 %QCT Parameters

Parameter	Range	Default	Description
QUEUE NUMBER	1-3950	—	Number of this queue. Queue numbers must be less than the value for the `FIRST_TEMP_QUEUE` in the %PROFILE section. A queue number can appear only once in the %QCT section. Queues 90 - 95 and 97 - 100 are reserved and cannot appear in this section.
BYTE QUOTA	>1	65,536	Maximum number of uncollected bytes that can reside in this queue when quotas are enabled. If the value is -1 or ".", the value in the TEMPLATE QUEUE is used. If no value is specified, the default value of 65,536 is used. **Note:** This parameter can be modified at runtime.
MSG_QUOTA	>1	128	Maximum number of uncollected messages that can reside in this queue when quotas are enabled. If the value is -1 or ".", the value in the TEMPLATE QUEUE is used. If no value is specified, the default value of 128 is used. **Note:** This parameter can be modified at runtime.

Table 2-9 %QCT Parameters

Parameter	Range	Default	Description
QUOTA ENABLED	All None Byte Msg "."	—	Controls the enabling and disabling of queue quotas. Values for this attribute are as follows: All — All queue quotas are ON. None — All queue quotas are OFF. Byte — Only the byte quota is enabled, msg quota is disabled. Msg — Only the msg quota is enabled; the byte quota is disabled. "." — Default value, which is "All" for all queues. **Note:** This parameter can be modified at runtime.
UCB SEND	—	—	This setting is ignored on MessageQ for UNIX and Windows NT systems.
QUEUE TYPE	P S M	P	Designates the queue type as follows: P: Primary queue P: Secondary queue M: Multi-reader queue **Note:** This parameter can be modified at runtime under the following conditions: ♦ The queue must be empty and have no processes attached. ♦ When changing a primary queue to a secondary queue, the primary queue cannot have any secondary queues defined.

Table 2-9 %QCT Parameters

Parameter	Range	Default	Description
OWNER	Any non-secondary queue	0	If QUEUE TYPE = S, specifies the number of the controlling queue. If QUEUE TYPE = P or M, use zero. **Note:** This parameter can be modified at runtime under the condition that the queue must be empty and have no processes attached.
CONF STYLE	II EI EO "."	EO	Determines how recoverable messages are confirmed by the receiver program. Possible values are as follows: II: Implicit, in-order confirmations EO: Explicit, in-order confirmations EO: Explicit, out-of-order confirmations ".": Accepts default, which is "EO". **Note:** This parameter can be modified at runtime.
PERM ACTIVE	Y N	N	Permanently active queue. Setting this value to Y allows processes to insert messages into this queue even if it is not currently attached by a client process. **Note:** This parameter can be modified at runtime.
NAME SCOPE	L (Local) G (Global)	—	Indicates the scope of the queue name. Valid settings are: L: The name is loaded into the Group Name Table, but not into the distributed naming service. The name can be translated by the service `pams_locate_q` function and by other processes in this group. G: The name is loaded into the distributed naming service. The name can be translated by the `pams_locate_q` function by other processes in this network.

Table 2-9 %QCT Parameters

Parameter	Range	Default	Description
SECURITY	Y N	N	Controls security, as follows: N: No security authorization checking for attach. Y: Activates security authorization checking. An Access Control List (ACL) file must be present and able to be opened for read or write in order to attach the queue. Create the queue security file in the following subdirectory of the MessageQ instalation directory: `\udataobj\dmq\b_BBBB\g_GGGGG\acl` where BBBB is a 4 digit bus ID and GGGGG is a 5 digit group ID
MRS_FLUSH_TIMER	-1 — Unlimited	100	Maximum number of milliseconds to wait before flushing the non-journaled message cache. ♦ If the value is -1 or absent, the default value is used. ♦ If the value is 0, no scheduled flushing occurs. However, flushing can occur as a result limits set by other parameters. ♦ When the chosen value is anything other than 0, messages will get flushed sooner than the limit only if the group server is idle. ♦ This parameter must be manually added to the file (see Listing 2-10).

Table 2-9 %QCT Parameters

Parameter	Range	Default	Description
MRS_FLUSH_DEPTH	-1 — Unlimited	0	Maximum number of messages that can be cached. ♦ If the value is -1 or absent, the default value is used. ♦ If the value is 0, the number of messages cached can be unlimited, but may be limited by other parameters such as MAX_CACHE_BYTES. ♦ >0 means to flush when that number of messages are cached. For example, a value of 1 means flush each message immediately; a value of 2 means flush after 2 messages are cached. ♦ This parameter must be manually added to the file (see Listing 2-10).

The Flush Timer and Flush Depth parameters must be manually added to the file as the last two columns in the QCT group. Listing 2-10 shows the addition of these parameters:

Listing 2-10 Sample %QCT Section

```
%QCT
!
!Queue      Queue    Byte...  Name  Security  Flush     Flush
!Name       Number   Quota... Scope           Timer     Depth
TEMPLATE    0     262144...    L       N      1000       10
QUEUE1      1       .  ...     L       N      1000       10
QUEUE2      2       .  ...     L       N      1000       10
```

Defining Name-to-Queue Translations in the Group Name Section

The %GNT section lets you define the generic queue names and their address for queues that do not necessarily appear in the %QCT section. Each line in the %GNT section defines a single name-to-queue translation.

Listing 2-11 shows a sample %GNT section where `INVENTORY_IN` is statically defined in the bus-wide namespace, while `ACCOUNTS_PAY` allows dynamic binding using `pams_bind_q`.

Listing 2-11 Sample %GNT Section

```
%GNT
!
!Name           Group.Queue     Scope

INVENTORY_IN    9.10            G
ACCOUNTS_PAY    0.0             G
%EOS
```

Table 2-10 describes the parameters in the %GNT section.

Table 2-10 %GNT Parameters

Parameter	Range	Default	Description
NAME	—	—	Equivalence name. Queue reference used by applications to refer to the queue by name. **Note:** MessageQ for NT systems is case-sensitive, while MessageQ for VMS is not. Use both upper and lower case for developers porting applications from Open VMS systems.

Table 2-10 %GNT Parameters

Parameter	Range	Default	Description
GROUP.QUEUE	—	—	Group and queue number to be returned to the calling process when this name is translated by the `pams_locate_q` function. Contains a group identifier, a period (.) and a queue number. Specifying zero for the group identifier implies the local group number. Specifying a zero for both the group and queue number means that this name can be dynamically assigned a queue address at runtime using the `pams_bind_q` function.
SCOPE	L (Local) G (Global)	—	Scope of name, as follows: L: Queue name is loaded into the group-wide namespace. Other processes in this group can issue the `pams_locate_q` function to translate the queue name. G: Queue name is loaded into the group-wide namespace and the bus-wide namespace. Any application on the message bus can issue the `pams_locate_q` function to translate the queue name.

Configuring Global Naming

To use local (group-wide) naming, configure queue names in the Queue Configuration Table (%QCT) or the Group Name Table (%GNT) section of the group initialization file. When the group starts up, MessageQ automatically creates the group namespace. It creates the process name space when an application attaches to the message queuing bus.

To enable your applications to use global (bus-wide) naming, you must perform additional configuration steps. First, you must decide the group or groups in which the naming agent will run. MessageQ allows you to specify a main group and an alternate group to run the Naming Agent. To configure a group to run the Naming Agent follow the steps outlined in the topic Configure Groups to Run or Use the Naming Agent.

Configure Groups to Run or Use the Naming Agent

The MessageQ Naming Agent is the MessageQ Server that maintains the namespace for name-to-queue address translations and performs the runtime queue lookup when an application refers to a queue by name. The %NAM section of the group

initialization file enables you define the group or groups in which the Naming Agent process will run. MessageQ allows the definition of two naming agents for each message queuing bus.

When MessageQ starts each group, it looks in this section of the initialization file to decide whether to start a naming agent for the group. For groups that do not run a Naming Agent, MessageQ uses the information in the %NAM section to direct requests to the Naming Agent. Groups must have a cross-group connections to the groups in which the Naming Agent runs.

Listing 2-12 shows a sample %NAM section.

Listing 2-12 Sample %NAM Section

```
%NAM
NA_GROUP    10
NA_GROUP    28
%EOS

%END
```

Valid ranges for the NA_GROUP parameter are 0 to 32,000. The default value is 0, meaning that a naming agent should be started in the local group.

Configure the Namespace

MessageQ supports only the creation of a “lightweight” namespace that is included with MessageQ. It does not support any add-on naming products such as CDS or DECdns. To use global naming, you must create a namespace on the nodes on which the Naming Agents will run. To create the global namespace, MessageQ users use a flat file system by creating the directory in which the MessageQ Naming Agent will maintain the name space.

MessageQ enables users to configure two Naming Agents to support global messaging for the environment. In order to allow the second Naming Agent to form a backup for the first, both Naming Agents must be configured to use the same name space.

Therefore, when you configure your name space for use by two Naming Agents that run on different systems, it must use a shared file system (directory structure) that is accessible to both Naming Agents.

After you create the name space, you must set the DMQNS_DEVICE environment variable to specify a device name for the name space because access to the MessageQ lightweight name space for global naming is system dependent. Therefore, when a naming agent is configured, it must be told what device name to use when it accesses this name space. This is done by setting the environment variable DMQNS_DEVICE as follows:

- For Windows NT, it should be set to a drive letter followed by a colon (for example, `c:`) or a full qualified sharename (e.g. `\\machine\share`)
- For VMS, it should be set to a disk device name or a logical name that translates to such a device (`dua1` or `disk$user1`)
- For UNIX, it should be set to a file system specification (for example, `/` or `/usr` or `/mnt/dmqns`)

Note that this environment variable need only be set for the group or groups in which the Naming Agent is running. Only the Naming Agent process is designed to use this environment variable setting to resolve the location of the name space. For detailed instructions on how to set an environment variable, refer to your system documentation.

For environments which use two naming agents, it is critically important to ensure that the device name set using the DMQNS_DEVICE environment variable on both systems points to the same device that stores the shared file system containing the MessageQ namespace.

In addition to specifying the DMQNS_DEVICE environment variable, MessageQ also provides another environment variable called DMQNS_DEFAULT_PATH that provides path information for the name space. Using this environment variable to specify path information can be used to add a layer of path information to that specified in the group initialization file DEFAULT_NAMESPACE_PATH parameter or by the application.

Configure a Default Namespace Path for Each Group

To use a global name, at least some portion of the path name must be specified. Path information can be supplied by the application, or you can use the DEFAULT_NAMESPACE_PATH parameter in the %PROFILE section of the group initialization file in order to specify path information for global names.

For global naming to function properly, this parameter must be set to the same value for all groups in which applications are designed to access the same name space. The following example illustrates how to set the default namespace path to refer to global names in a namespace created in a directory called `/u/mydir`.

```
DEFAULT_NAMESPACE_PATH /u/mydir
```

For example, for testing purposes, you might set this parameter to look at a copy of the production name space that you store in your own development directory. However, when the application is deployed into production, the application will reference the common name space shared by all production systems.

Define the Queue Names in the Group Initialization File

Use the Queue Configuration Table (%QCT) or the Group Name Table (%GNT) of the group initialization file to create static or dynamic definitions for global names as follows:

- Define global static names in the %QCT or %GNT by providing the name, the queue address and setting the name scope identifier to "G" for global names.
- Define global dynamic names by supplying the name, "0.0" as the address and the "G" identifier for global names. Names defined with a 0.0 address can be dynamically bound to a queue address at runtime using the `pams_bind_q` function.

The following example shows static and dynamic global name definitions in the GNT section of the initialization file.

Listing 2-13 Sample Group Name Table for Global Naming

```
%GNT
!Name           Group.Queue     Scope
```

```
widgets          9.10            G
red_widgets      0.0             G
%EOS
```

When an application refers to a queue by name using the `pams_locate_q` or the `pams_bind_q` functions, it can specify the name as one of the following:

unqualified name—The application uses only the queue name such as "widgets" and does not specify the path. The Naming Agent automatically prefixes the name with the value of the environment variable DMQNS_DEVICE. Further, it prepends the value of the environment variable DMQNS_DEFAULT_PATH unless the path name specified by the DEFAULT_NAMESPACE_PATH begins with a "/". For example, if the DMQNS_DEVICE environment variable is set to "`c:`" and the DEFAULT_NAMESPACE_PATH is set to "`/inventory`", the Naming Agent would search for the name "`widgets`" in:

```
c:/inventory/widgets
```

partially qualified name—The applications specifies the queue name and a portion of the path name. The Naming Agent automatically prefixes the pathname and queue name with the device specified as the DMQNS_DEVICE environment variable and the setting of the DEFAULT_NAMESPACE_PATH parameter or the DMQNS_DEFAULT_PATH environment variable. For example, if the DMQNS_DEVICE environment variable were set to "`c:`" and the DEFAULT_NAMESPACE_PATH were set to "`/inventory`", the Naming Agent would search for the name "`test/widgets`" in:

```
c:/inventory/test/widgets
```

fully qualified name—The application specifies that the name is a fully qualified name using "/" as the first character of the name. When the first character of a name begins with "/", the Naming Agent does not prepend any information to the name other than the device name specified by the DMQNS_DEVICE environment variable. This means that a fully qualified name includes the full path name and queue name. For example, if the DMQNS_DEVICE environment variable is set to "`c:`" and the DEFAULT_NAMESPACE_PATH is set to "`/inventory`", the Naming Agent would search for the name "`/production/test/widgets`" in:

```
c:/production/test/widgets
```

The use of unqualified, partially qualified, and fully qualified names gives application developers significant flexibility in using global name references. Listing 2-14 provides several more examples of how global names are resolved. In this example, the DMQNS_DEVICE environment variable is set to "C:" and the DMQNS_DEFAULT_PATH is set to "/dmq/dmqns".

Listing 2-14 Sample Global Names and Their Resolution

```
Name Used in API  DEFAULT_NAMESPACE_PATH  Name Searched

toto                bus1          C:/dmq/dmqns/bus1/toto
mypath/toto         bus1          C:/dmq/dmqns/bus1/mypath/toto
/anotherpath/toto   bus1          C:/anotherpath/toto
toto/bus1C:         /bus1         /toto
/mypath/toto/bus1C: /bus1         /mypath/toto
/anotherpath/toto   /bus          C:/anotherpath/toto
```

Refer to the *MessageQ Programmer's Guide* for more information on designing applications to use the MessageQ global naming feature.

Using the Test Utility to Test Your Group Configuration

The Test utility allows applications developers to send and receive messages between applications to:

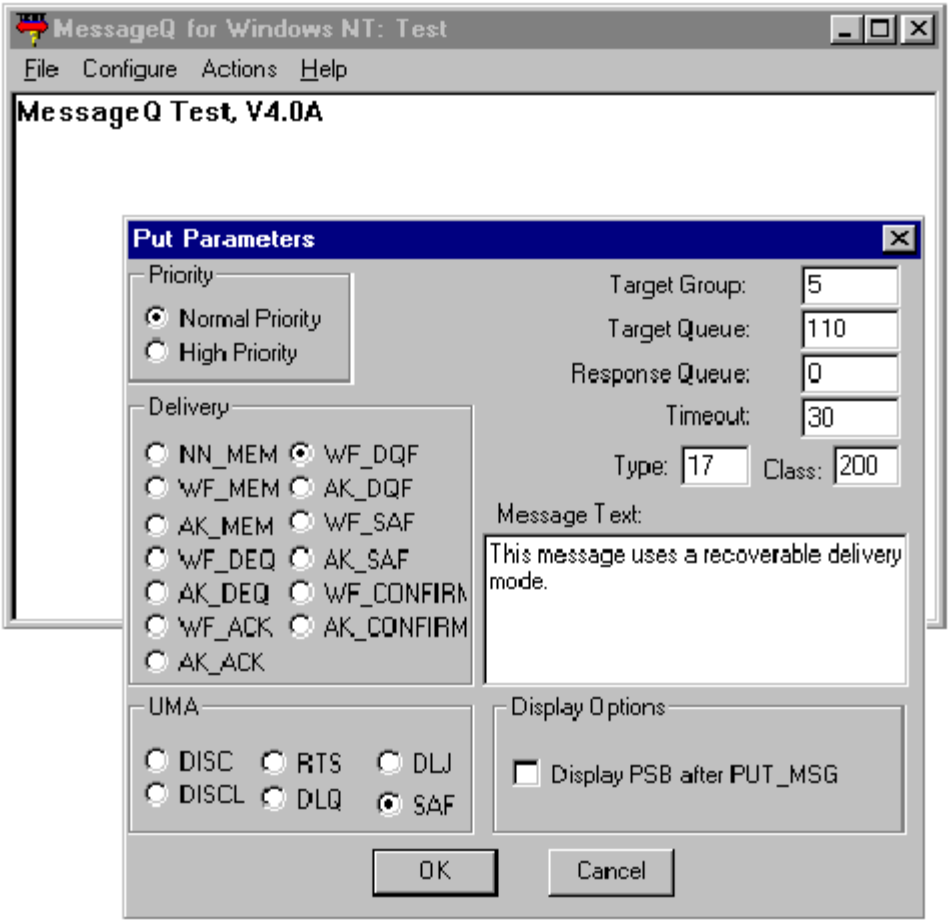
- Build interactive tests of application modules.
- Simulate send and receive messages to any target from any source.

- Exercise the queues in the MessageQ system.

The MessageQ Test utility enables application developers to interactively attach to a permanent or temporary queue, read messages from a script file or available interprocess messages, and pass messages to a defined target queue. To invoke the Test utility, double click on the Test utility icon.

Figure 2-5 illustrates the use of the Test utility to send a message on a Windows NT system.

Figure 2-5 MessageQ Test Utility



3 Managing MessageQ

This chapter describes how to manage MessageQ software on your system by performing the following tasks:

- Using the Monitor Utility
- Connecting to the MessageQ Environment
- Starting a Message Queuing Group
- Shutting Down a Running Group
- Starting a Cross-Group Connection
- Stopping a Cross-Group Connection
- Running the Client Library Server (CLS)
- Restricting Remote Access to the CLS
- Managing Message Recovery Services (MRS)
- Changing Group Characteristics at Runtime

Using the Monitor Utility

The Monitor utility is a MessageQ application that allows you to monitor and control the performance of your MessageQ NT system.

This chapter describes how the Monitor utility lets you manage MessageQ functions such as:

- Connecting to the MessageQ Environment
- Starting and Stopping Queues
- Starting and Stopping Cross-Group Connections
- Managing Message Recovery Services

Connecting to the MessageQ Environment

Before running a program that uses MessageQ, you must set the environment to identify the message queuing bus and the message queuing group with which the program will be associated. Connecting to the MessageQ environment can be accomplished from the command line, or from the Monitor utility.

Connecting to a Group from the Command Line

A MessageQ program associates itself with a specified bus ID and group ID at run time by obtaining the values of the DMQ_BUS_ID and DMQ_GROUP_ID environment variables.

These values can be set from the command line, as follows:

```
SET DMQ_BUS_ID=bus_id
SET DMQ_GROUP_ID=group_id
```

where:

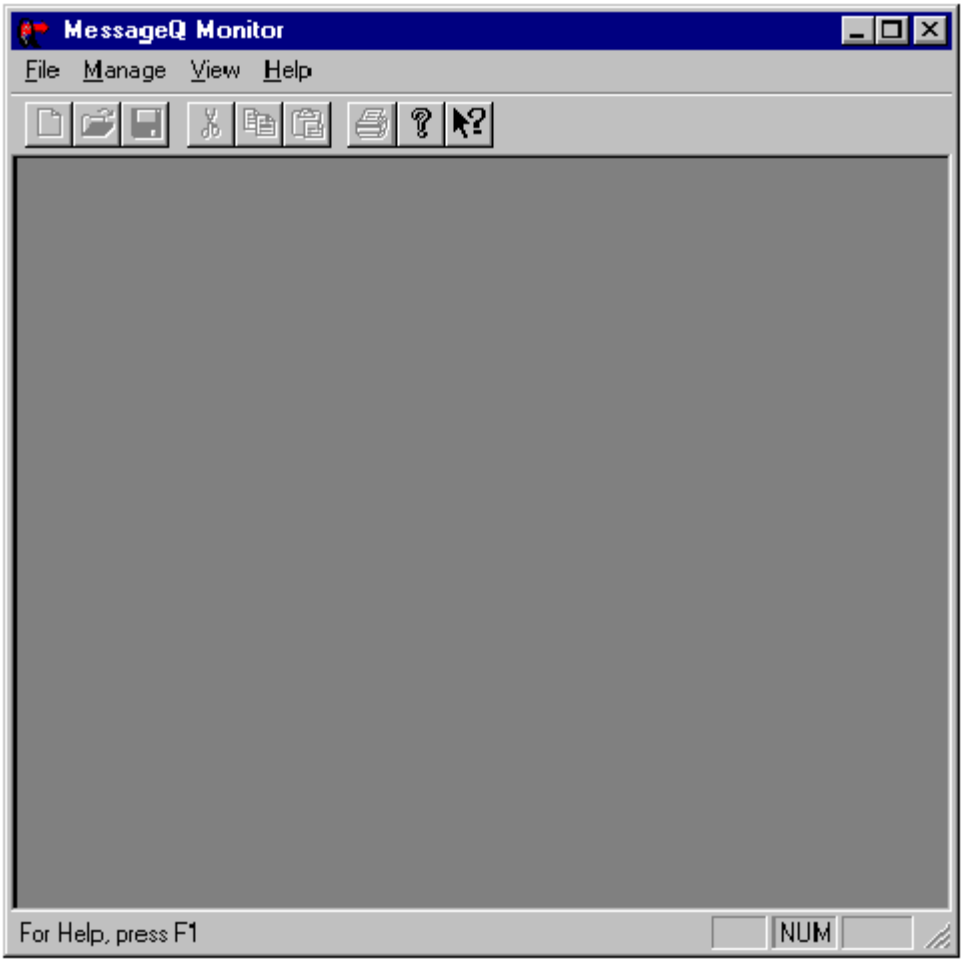
bus_id	Numeric bus ID; 1 to 9999
group_id	Numeric group ID; 1 to 32000

You can establish default values for these symbols by adding these environment variables to your profile with the System applet in the Control Panel. You can also add code to your MessageQ application that sets these environment variables using the WIN32 call *SetEnvironment* variable. See your Windows NT programmer's documentation for more information.

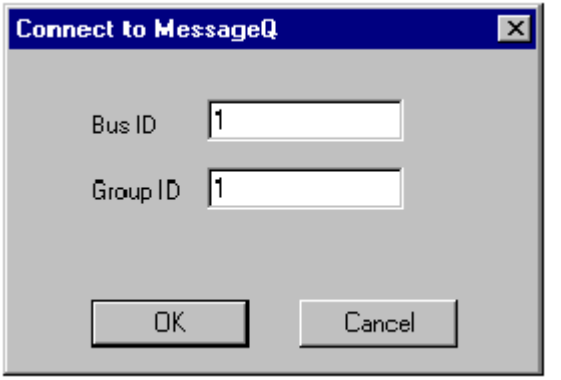
Connecting to a Running Group From the Monitor Utility

To use the Monitor Utility, you must have at least one group running in order to connect to the MessageQ environment. You can connect to a running group from the MessageQ Monitor utility using the following procedure:

1. Select the **File** pull-down menu and choose **Connect**.



2. Enter the Bus ID, Group ID, and click **OK**.



Starting a Message Queuing Group

There are several ways you can start a MessageQ for Windows NT message queuing group. You can start it from:

- The command line
- The Monitor utility
- A Group Control Process (GCP) icon
- The Startup program group

You can also start MessageQ as a Windows NT service. See Chapter , "Starting the MessageQ Service for Windows NT," for more information

Starting a Group from the Command Line

To start MessageQ you must invoke the MessageQ startup procedure (`dmqstartup`) to start each referenced group in the group initialization file.

The MessageQ startup procedure executable image is `dmqstartup`, and is run interactively by entering the following command:

```
dmqstartup -b bus_id -g group_id -f filespec -n group_name [-l logfile_name]
```

Where:

`-b bus_id`	Numeric bus ID; maximum of four digits
`-g group_id`	Numeric group ID; maximum of five digits
`-f filespec`	The pathname and file specification of the group initialization file
`-n group_name`	Descriptive alphanumeric name for this group. No spaces or special characters are allowed.
`-l logfile_name`	Optional log file name. You must specify a pathname and file specification. Note that if you do not use a log file, the output displays to the screen in an "unformatted" manner.

MessageQ for Windows NT uses the bus ID and group ID to locate configuration information in the Windows NT Registry. Note that when you start a group from the command line, the default log file `dmqlog.log` is placed in the current directory.

Starting a Group From the Monitor Utility

You can start a group from the MessageQ Monitor utility using the following procedure:

1. Select the **Manage** pull-down menu, choose the **Group** option and then select **Start**.
2. Enter the Bus Number ID
3. Enter the Group Number ID.
4. Enter the Group Name.
5. Enter the full path to the Group Initialization File.

   For example: C:`\user3\users\jones\bigfile.init`

6. Click **OK**.

Creating a Custom Group Startup Icon on the Desktop

Follow this procedure to create a custom Group Startup icon to start a message queuing group under MessageQ for Windows NT:

1. Position the cursor anywhere on the desktop and click the right mouse button.
2. Select the **New** and then **Shortcut** option to create a desktop shortcut.
3. On the Create Shortcut menu, enter the appropriate syntax to start the MessageQ Group Control Process (`dmqgcp.exe`) supplying the necessary information to startup the selected group. For example:

   ```
   c:\...\bin\dmqgcp.exe -b 1 -g 9 -f init.txt -n group9
   ```

   Click the **Next** button to proceed to the next screen.
4. On the Select a Title for the Program screen enter a name for the desktop shortcut and and press **Finish**. The newly created icon will appear on your desktop.
5. Now you can start this group at any time by doubleclicking the icon.

Starting a Group from the Startup Program Group

Follow this procedure to add the message queuing group to the startup menu to be started automatically when you log in to your Windows NT user account.

1. From the start menu select the **Settings** and then the **Taskbar** option.
2. On the Taskbar Properties menu, select the Start Menu Programs tab and click the **Add** button.
3. On the Create Shortcut menu, enter the appropriate syntax to start the MessageQ Group Control Process (`dmqgcp.exe`) supplying the necessary information to startup the selected group. For example:

   ```
   c:\...\bin\dmqgcp.exe -b 1 -g 9 -f init.txt -n group9
   ```

   Click the Next button to proceed to the next screen.

4. On the Select Program Folder menu, select the Startup folder as the location for the shortcut.
5. On the Select a Title for the Program screen enter a name for the desktop shortcut and press **Finish**. The shortcut will be available in the Startup folder and the group will automatically start each time you log in to your account.

Shutting Down a Running Group

MessageQ offers a shutdown procedure to shut down a running group from the command line, the Monitor Utility, or from the Group Control Process system menu.

Shutting Down a Group from the Command Line

The MessageQ shutdown procedure executable image is `dmqshutdown`, and is run interactively by entering the following command:

```
dmqshutdown -b bus_id -g group_id [-f]
```

Where:

`-b` *bus_id*	Numeric bus ID; maximum of four digits
`-g` *group_id*	Numeric group ID; maximum of five digits
`-f`	Fast shutdown. If you select this option, MessageQ terminates all MessageQ processes in the group immediately. When the group is stopped, messages can no longer be delivered to the queues and all pending messages in the queues are lost.

The group shutdown procedure, with the `-f` option not selected, stops all queues immediately without draining the messages in the queues. Further, it stops all links immediately and provides an orderly shutdown of all MessageQ processes.

Shutting Down a Group From the Monitor Utility

Use the following procedure to shutdown a group from the Monitor Utility:

1. Select the **Manage** pull-down menu.
2. Select **Group**.
3. Select **Stop Fast** or **Stop**.

   If you select **Stop**, MessageQ stops all queues in the group, allowing them to be emptied. Applications can continue to read messages from the queues until all are empty. However, applications cannot place messages in the queues. After all the queues are empty, the MessageQ processes in the group will terminate.

   If you select **Stop Fast**, MessageQ terminates all MessageQ processes in the group immediately. When the group is stopped, messages can no longer be delivered to the queues and all pending messages in the queues are lost.
4. MessageQ displays a warning:

   ```
   Do you really want to stop group n?
   ```
5. Click **OK**.

Shutting Down a Group From the Group Control Process System Menu

Use the following procedure to stop a message queuing group from the Group Control Process system menu:

1. Select the GCP icon for the running group on the desktop to display the System menu.
2. Select **Close** from the System menu to remove the GCP icon.

   Only icons for the GCP and CLS processes are displayed on the status bar. To check for any orphaned group processes, use the Windows NT Task Manager utility.

Note: All MessageQ processes can be closed by selecting Close from their respective system menus. However, this should only be done if the GCP shutdown fails to terminate a process normally.

Starting a Cross-Group Connection

You can start a cross-group connection, also called a link, using the MessageQ Monitor utility.

The following procedure describes how to start a Cross-Group connection:

1. Select the **Manage** pull-down menu.
2. Select **Link**.
3. Select **Start**.
4. Select a link number from the list.
5. Click **OK**.

Stopping a Cross-Group Connection

You can stop a cross-group connection using the MessageQ Monitor utility.

The following procedure describes how to stop a cross-group connection:.

1. Select the **Manage** pull-down menu.
2. Select **Link**.
3. Select **Stop Fast**.
4. Select a link number from the list.
5. Click **OK**.

Starting a Queue

You can restart a queue that has been previously stopped using the MessageQ Monitor utility.

The following procedures describe how to start a queue:

1. Select the **Manage** pull-down menu.
2. Select **Queue**.
3. Select **Start**.
4. Select a queue number from the list.
5. Click **OK**.

Stopping a Queue

You can stop a queue using the MessageQ Monitor utility.

The following procedure describes how to stop a queue:

1. Select the **Manage** pull-down menu.
2. Select **Queue**.
3. Select **Stop Fast** or **Stop**

   If you select **Stop**, MessageQ stops all queues in the group, allowing them to be emptied. Applications can continue to read messages from the queues until all are empty. However, applications cannot place messages in the queues. After all the queues are empty, the MessageQ processes in the group will terminate.

   If you select **Stop Fast**, MessageQ terminates all MessageQ processes in the group immediately. When the group is stopped, messages can no longer be delivered to the queues and all pending messages in the queues are lost.
4. Select a queue number from the list.

5. Click **OK**.

Running the Client Library Server

The Client Library Server (CLS) runs as a Windows NT application program that uses MessageQ for Windows NT. It is important to define the environment variables described in Connecting to the MessageQ Environment earlier in this chapter, because they identify the MessageQ Server group where the Windows clients connect. Additionally, the message queues used by Windows clients must be defined in the MessageQ for Windows NT group configuration.

The CLS creates a thread for each client connection from a MessageQ Windows Client. The CLS window displays a count of the number of clients currently connected, and for each client displays the server thread ID, client host name, client task ID, and queue number attached.

For convenience and ease of application management, CLS servers are normally started when a MessageQ group starts, and are stopped when the MessageQ group is shut down. Once the MessageQ group is started ou can start a CLS server from the Monitor utility.

Starting CLS

Use the following procedure to start the CLS:

1. Select the **Manage** pull-down menu.
2. Select **CLS**.
3. Select **Start**.
4. Enter the endpoint number.

Stopping the CLS

Use the following procedure to stop the CLS:

1. Select the **Manage** pull-down menu.
2. Select **CLS**.
3. Select **Stop Fast**.
4. Enter the endpoint number.

Restricting Remote Access to CLS

The CLS security file is a text file containing a table of client entries. Each client entry contains a list of endpoints and queues which the client may use. CLS uses the security file to restrict access by remote clients to those endpoints and queues. MessageQ groups can have their own separate security files, or can share one file jointly.

You can create a security file in one of two ways:

- Edit the template file that is distributed in the \TEMPLATES directory of the media kit.
- On Windows systems, you can use the Security Utility to edit the security file.

For LIBRARY client servers on NT systems, a template security file is available at the following location:

```
Program Files\BEA Systems\MessageQ\TEMPLATES\dmqclsec.txt
```

Begin by making a copy of the template security file. Then, edit the copy to remove the sample entries and add entries for the client systems in your environment. Copy the file to its target location and then associate the security file with the message queuing group.

When a CLS is started, it loads the security file specified in the %CLS section of the group initialization file. If no security file is specified, CLS will not restrict access by remote clients. Each CLS can have a separate security file, or a security file can be shared by multiple CLS processes.

For information about the CLS Security Utility, refer to the *MessageQ Client for Windows User's Guide*.

Managing Message Recovery Services

MessageQ Message Recovery Services (MRS) provide a mechanism for guaranteed message delivery by storing messages on disk and automatically attempting redelivery until the message is received by the target system.

If you enabled MRS as part of your group configuration, you can replay the contents of postconfirmation or dead letter journals using the Journal Replay utility. You may also print the contents of postconfirmation, dead letter, destination queue, or store and forward journals using the Journal Dump utility.

All journal file names are 16 characters long. Each journal file name contains a 12-digit filename, a dot, and a 3-character extension that identifies the journal type. Note that journal file names are unique within a group, but not between groups. Therefore, message queuing environments running more than one message queuing bus must ensure that journal files are not accidentally shared by groups running on different message queuing buses.

Valid journal filenames are of the format:

Format	Journal Type
ggggqqqqssss.DQF	Destination Queue File
ggggrrrrssss.SAF	Store and Forward
*gggg*sssssssss.PCJ	Postconfirmation Journal
*gggg*sssssssss.DLJ	Dead Letter Journal

Where:

g	Four-digit local group number in hexadecimal
q	Four-digit queue number in hexadecimal

r	Four-digit remote group number in hexadecimal
s	File sequence number in hexadecimal within a set of files

Replaying Journal Messages

The Journal Replay utility lets you resend the contents of postconfirmation or dead letter journals. The Journal Replay utility can use live recovery journals; however, if the journal file is currently in use, you must make a copy of the file before you can replay it. The Journal Replay utility can resend the entire journal file.

Messages sent from the postconfirmation or dead letter journal will be sent using the same source address, target address, delivery mode, and user notification. If the original message requested notification, the notification message is directed to the journal replay process instead of the original sender program.

To invoke the Journal Replay utility, enter the following command syntax. Switches enclosed in parentheses indicate optional command qualifiers.

```
dmqjplay -b bus -g group -m mode -t journal_type -j journal_path [-l log_path]
```

Table 3-1 describes the command options for `dmqjplay`.

Table 3-1 Journal Replay Utility Command Options

Switch	Argument	Description
`-b`	*bus*	Numeric Bus ID; maximum of 4 digits
`-g`	*group*	Numeric Group ID; maximum of 5 digits
`-m`	*mode*	A constant identifying the selected processing mode, as follows: r=replay — Retransmit the contents of the journal file and delete each message when it reaches the delivery interest point. t=transfer — Retransmit the contents of the journal file and delete each message when it reaches the delivery interest point. d=delete — Delete the contents of the selected journal.

Table 3-1 Journal Replay Utility Command Options

Switch	Argument	Description
-t	*journal_type*	A constant that designates the file to be replayed, as follows: d — Dead letter p — Postconfirmation
-j	*journal_path*	The pathname of the dead letter journal or postconfirmation journal that is to be replayed
-l	*log_path*	Optional log file. You must specify a pathname and file specification.

Printing Journal Files

The Journal Dump utility lets you produce a formatted report of the contents of a dead letter journal, postconfirmation journal, destination queue file, or a store and forward file. The Journal Dump utility can be used to print a live journal file if it is not in use. If the journal file is in use, you must halt the message recovery system for the group, and then use the Journal Dump utility to print the contents of the journal file.

To run the Journal Dump utility, enter the following command syntax (switches enclosed in parentheses indicate optional command qualifiers):

```
dmqjdump -b bus -g group -q queue -t journal_type -h header_type -m message_format
-j journal_path [-d] [-l log_path] [-o output_file] [-n number]
```

Table 3-2 describes the command options for `dmqjdump`.

Table 3-2 Journal Dump Utility Command Options

Switch	Argument	Description
-b	*bus*	Numeric Bus ID; maximum of 4 digits
-g	*group*	Numeric Group ID; maximum of 5 digits
-q	*queue*	Queue number

Table 3-2 Journal Dump Utility Command Options

Switch	Argument	Description
-t	*journal_type*	A constant that designates the type of journal to be replayed, as follows: dlj — dlj – Dead letter journal dqf — dqf – Destination queue file pcj — pcj – Postconfirmation journal saf — saf – Store and forward file
-h	*header_type*	A constant that designates the header type, as follows: summary — Displays the source target, type and class of each message that is dumped. detail — Displays internal header fields of each message that is dumped.
-m	*message_format*	Specifies a valid message format, as follows: hex — Displays output in hexadecimal bytes with ASCII translation. script — Displays output in MessageQ script format.
-j	*journal_path*	The pathname of the dead letter journal, postconfirmation journal, destination queue, or store and forward file that is to be dumped.
-d		Deletes messages from the journal as they are dumped. **Note:** This option should be used with caution.
-l	*log_path*	Optional log file. You must specify a pathname and file specification.
-o	*output_file*	A pathname and file specification to receive the dump output
-n	*number*	The number of messages in the file you want to print, starting with the first message

Changing Group Characteristics at Runtime

You can use the Loader utility to dynamically reload the group initialization file without having to shutdown and restart MessageQ. The Loader utility lets you enter modified settings or parameter values into a running group initialization file.

To run the Loader utility, enter the following command format:

```
dmqloader -b n -g n -f init_file_path [-l logfile]
```

Where:

-b	Numeric Bus ID; maximum of 4 digits
-g	Numeric Group ID; maximum of 5 digits
-f	The pathname and file specification of the group initialization file
-l	Optional log file. You must specify a pathname and file specification

Table 3-3 describes the parameters in the group initialization file that can be modified at runtime.

Table 3-3 Modifiable Parameters in the Group Initialization File

Section in Initialization File	Parameter	Runtime Restriction?
%PROFILE	ACCEPT_KILL_COMMAND	NO
	ATTACH_TMO	NO
	DEFAULT_NAMESPACE_PATH	NO
%CLS	MAX_CLIENTS	YES. CLS must be stopped. This parameter applies to OpenVMS systems only.
	SECURITY_FILE	YES. CLS must be stopped.

Table 3-3 Modifiable Parameters in the Group Initialization File

Section in Initialization File	Parameter	Runtime Restriction?
%XGROUP	RECONNECT	YES. The link must be disabled.
	RECONN_TIMER	YES. The link must be disabled.
	WINDOW_DELAY	YES. The link must be disabled.
	WINDOW_SIZE	NO
%QCT	BYTE_QUOTA	NO
	MESSAGE_QUOTA	NO
	MESSAGE_QUOTA_ENABLE	NO
	BYTE_QUOTA_ENABLED	NO
	TYPE	YES. The queue must be empty and have no processes attached. When changing a primary queue to a secondary queue, the primary queue cannot have any secondary queues defined.
	OWNER	YES. The queue must be empty and have no processes attached. To set this parameter to a value other than zero, the queue must be defined as a secondary queue, and the owning queue must be defined and be a primary queue.
	MRS_CONFIRM_STYLE	NO
	PERM_ACTIVE	YES. The queue must be empty and have no processes attached.
	SECURITY_ENABLED	NO

Enabling or Disabling Quotas

The MessageQ Monitor utility lets you enable or disable quotas on permanent queues that you previously defined in the %QCT section of the group initialization file. You can enforce quotas on the maximum number of collected messages and bytes that can reside in a given queue.

If you are not concerned whether your application exceeds quotas, you should disable quotas to increase performance and reduce screen clutter.

Use the following procedure to enable or disable quotas:

1. Select the **Manage** pull-down menu.
2. Select **Quotas**.
3. Select **Enable** or **Disable**.

Setting Quotas for Dynamic Queues

Queue numbers with a number greater than Temporary Queue (FIRST_TEMP_QUEUE) are dynamically assigned. The BYTE_QUOTA and MSG_QUOTA values for dynamically assigned queues are set from the TEMPLATE entry in the %QCT section. Do not change any other value in the TEMPLATE queue entry.

Enabling Queue Quotas for Selective Queues

Previous versions of MessageQ required you to enable or disable queue quotas for an entire message queuing group. However, you can now enable queue quotas on a per queue basis on UNIX systems. You can enable or disable message and byte quotas for a selected queue at startup by setting the Quota Enable field for each entry in the %QCT section of the group initialization file. Table 2-9 contains the values that you can specify for the QUOTA_ENABLED attribute.

You can use the Monitor utility to enable or disable message and byte quotas for a selected queue at runtime. (See the Enabling or Disabling Quotas topic for more information.)

Defining Timeout Intervals for Link Drivers

The MessageQ link drivers provide parameters that let you define timeout intervals for testing the state of a cross-group connection, or aborting a connection with a cross-group link. These parameters are implemented as the following environment variables:

DMQLD_PING_INTERVAL	Specifies the amount of time a Link Sender process should wait before issuing another ping message to test the state of a connection. The ping interval is the number of seconds between tests. The default value is 30 seconds.
DMQLD_PING_TIMEOUT	Specifies the amount of time a Link Sender should wait for a response to a ping message before aborting. The default value is 60 seconds.

When tracing is enabled, the Link Receiver logs successful ping responses to the group log file.

If you must use a value other than the default, set both environment variables before starting your group.

4 Monitoring MessageQ

The Monitor utility is a MessageQ application that allows you to monitor and control the performance of your MessageQ for Windows NT message queuing system.

Using the Monitor utility requires that you be connected to a bus and group. If you set the DMQ_BUS_ID and DMQ_GROUP_ID environment variables prior to invoking the Monitor utility, you will connect to that bus and a group when you invoke the Monitor utility. If you do not set the environment variables before invoking the Monitor utility, you must connect to the bus and group from the command line or from the Monitor utility.

Using the Monitor utility, you can:

- Open a Monitor view and attach it to a MessageQ group.
- Enable, disable, and reset statistics collection.
- Enable and disable quotas.
- View cumulative counts of the messages sent and received for all links and queues of a selected group.
- Shut down groups.

Invoking the Monitor Utility

To start the Monitor utility:

1. Open the MessageQ Version 5.0 group in the Program Manager.
2. Double click the Monitor icon.

Enabling Statistics

To monitor your application's messaging capabilities, you need to collect statistics on the queue and link activities of the groups. Because statistic collection is not automatically started when you open the Monitor utility, you must enable statistics gathering using the Enable Statistics option in the Manage menu. To disable statistics gathering, click again on the Enable Statistics option in the Manage menu to remove the check mark. Statistics stop being collected when you exit the Monitor utility.

At any time while statistics are enabled, they can be reset for a selected group's links and queues using the Statistics and Reset items under the Manage menu. Link and queue statistics can be reset either individually or as a whole. Reset zeroes out the statistics collection counters. Using Reset allows you to view your application's messaging traffic over short-term periods, which can be useful for detecting messaging bottlenecks.

Enabling Quotas

The Monitor utility allows you to enable or disable quotas on permanent queues that you have defined in the %QCT section of the group initialization file. You can enforce quotas on the maximum number of uncollected messages and bytes that can reside in a given queue. For more information on setting these quotas, see the Defining Queues in the Queue Configuration Section topic in Chapter 2.

To enable quota enforcement, use the Quotas and Enable items from the Manage menu. You can disable quota enforcement by using the Disable item. If you are not concerned whether your application exceeds quotas, you should disable quotas to increase performance and reduce screen clutter.

Viewing Link and Queue Activity

To view link and queue statistics of MessageQ message queuing groups, use the Link Information and Queue Information items of the View menu to display the following dialogs:

- Link Information
  - Traffic Counts
  - Traffic Rates
  - Traffic Detail
  - Detail
- Queue Information
  - Queue Traffic Counts
  - Queue Traffic Rates
  - Queue Traffic Detail
  - Detail

Viewing Link Traffic Counts

Use the Link Information and Traffic Counts menu items to view the connection state and the cumulative messages and bytes for links between groups. Understand that the group for the node from which you are using the Monitor utility will not appear in this dialog; only the groups to which that group is currently connected or could be connected to will appear.

Table 4-1 describes the fields of the Link Traffic Counts dialog.

Table 4-1 Link Traffic Counts Dialog

Statistic	Definition
Group Number	Number of the group

Table 4-1 Link Traffic Counts Dialog

Statistic	Definition
State Receiver	State of the inbound connection; either connected or unconnected
State Sender	State of the outbound connection; either connected or unconnected
Messages Sent	Cumulative messages sent from the last time statistics were reset
Messages Received	Cumulative messages received from the last time statistics were reset
Bytes Sent	Cumulative bytes sent from the last time statistics were reset
Bytes Received	Cumulative bytes received from the last time statistics were reset
Disconnected In	Cumulative number of successful connections since the last disconnect
Disconnected Out	Cumulative number of failed connection attempts since the last disconnect

Use the Disconnected In/Out statistic to see if any link between groups is constantly going up and down (cycling). A cycling link could indicate a network timeout problem.

This dialog can be useful for diagnosing bad message puts. For example, if you sent a cross-group message and it does not show up, yet the Viewing Link Traffic Counts dialog shows that all the links are up, your application may be in error.

Viewing Link Traffic Rates

Use the Link Information and Traffic Rates menu items to view the connection state and the average byte and message counts per second for the links of all groups. Average bytes and message counts per second is referred to as “throughput rates”. Table 4-2 describes the fields of the Link Traffic Rates dialog.

Table 4-2 Link Traffic Rates Dialog

Statistic	Description
Group Number	Number of the group
State Receiver	State of the inbound connection; either connected or unconnected

Table 4-2 Link Traffic Rates Dialog

Statistic	Description
State Sender	State of the outbound connection; either connected or unconnected
Messages Sent per Second	Average number of messages sent per second since the last time statistics were reset
Messages Received per Second	Average number of messages received per second since the last time statistics were reset
Bytes Sent per Second	Average number of bytes sent per second since the last time statistics were reset
Bytes Received per Second	Average number of bytes received per second since the last time statistics were reset

The Viewing Link Traffic Rates dialog is useful for checking the put/get rate of messages:

- If the throughput is lower than expected, the network may be busier than usual (hence slower) due to other traffic.
- If the throughput is lower than expected and you suspect that enqueuing is occurring at a normal rate, but dequeuing is not happening quickly enough, the remote group's system could be congested.

Viewing Link Traffic Detail

Use the Link Information and Traffic Detail menu items to view the connection state and the average byte and message counts of a single link.

Table 4-3 describes the fields of the Link Traffic Detail dialog.

Table 4-3 Link Traffic Detail Dialog

Statistic	Description
Group Name	Name of the group
Group Number	Number of the group

Table 4-3 Link Traffic Detail Dialog

Statistic	Description
State Receiver	State of the inbound connection; either connected or unconnected
State Sender	State of the outbound connection; either connected or unconnected
Messages Sent per Second	Average number of messages sent per second since the last time statistics were reset
Messages Received per Second	Average number of messages received per second since the last time statistics were reset
Bytes Sent per Second	Average number of bytes sent per second since the last time statistics were reset
Bytes Received per Second	Average number of bytes received per second since the last time statistics were reset

Use the Link Information and Traffic Detail dialog if you suspect that a traffic problem is occurring between two particular groups. If you want to narrow down the problem further and suspect that the problem is due to one queue only, use the Queue Traffic Detail dialog.

Viewing Link Detail

Use the Link Information and Detail menu items to view the link attributes of a selected group.

Table 4-4 describes the fields of the Link Detail dialog.

Table 4-4 Link Detail Dialog

Statistic	Description
Group Name	Name of the group
Group Number	Number of the group
Time Connected	Time the current connection to a remote group was made
Time Disconnected	Time the last connection to a remote group was severed

Table 4-4 Link Detail Dialog

Statistic	Description
Time Statistics Reset	Time statistics were last reset
PID	Process ID for the sender and receiver
State	State of the sender and receiver; either connected or unconnected
Window Timer	Number of seconds the sender must wait before using a new window. This value is set by the Window Delay parameter in the %XGROUP section of the group initialization file (see the Enabling Network Connections in the Cross-Group Section topic in Chapter 2).
Window Size	Number of messages the link driver can send before it has to ask permission to send more. Window size is initially set to 0 and changes dynamically, based on the message throughput needs and the minimum and maximum values of the Window Size parameter in the %XGROUP section of the group initialization file. For example, if a sender sends 10 messages per second, but the receiver can only dequeue 1 message per second, a congestion situation is created. The window size changes dynamically in a congestion situation in order to reach an equilibrium between the sender and receiver.
Window Max	Number of messages that can be sent before adjusting for congestion. This value is set by the Window Size parameter in the %XGROUP section of the group initialization file (see the Enabling Network Connections in the Cross-Group Section topic in Chapter 2).

Viewing Queue Traffic Counts

Use the Queue Information and Traffic Counts menu items to view the cumulative counts of the messages sent and received for all queues of the group. Note that statistics must be turned on for this screen to be useful.

Table 4-5 describes the fields of the Queue Traffic Counts dialog.

Table 4-5 Queue Traffic Counts Dialog

Statistic	Description
Queue Number	Number of the queue
Messages Sent	Cumulative number of messages sent since statistics were last reset
Messages Received	Cumulative number of messages received since statistics were last reset
Messages Current	Current number of messages waiting to be read. If this value continuously increments, it could be an indication that the receiving process cannot keep up with traffic.
Bytes Sent	Cumulative size, in bytes, of messages sent since statistics were last reset
Bytes Received	Cumulative size, in bytes, of messages received since statistics were last reset
Bytes Current	Size, in bytes, of the current messages waiting to be read
Queue Attributes	The type of queue, permanent or temporary. Any queue defined in the Configure Queues dialog is by default a primary queue. Queue 100, a reserved queue, is labeled Reserved

Viewing Queue Traffic Rates

Use the Queue Information and Traffic Rates menu items to view the traffic rates of all queues of a particular group. Traffic rates are defined as the throughput of messages, either sent or received, per second. The Queue Traffic Rates dialog allows you to verify that suspected problems with message throughput exist.

Note that you must reset statistics each time you monitor traffic rates. Failing to do so will result in an inaccurate throughput assessment.

Table 4-6 describes the fields of the Queue Traffic Rates dialog.

Table 4-6 Queue Traffic Rates Dialog

Statistic	Description
Queue Number	Number of the queue
Messages Sent per Second	Average number of messages sent per second since statistics were last reset
Messages Received per Second	Average number of messages received per second since statistics were last reset
Bytes Sent per Second	Average number of bytes sent per second since statistics were last reset
Bytes Received per Second	Average number of bytes received per second since statistics were last reset
Peak Messages	Highest number of messages in the queue at any given time since statistics were last reset.
Peak Bytes	Highest number of bytes in the queue at any given time since statistics were last reset.
Queue Attributes	The type of queue, permanent or temporary. Any queue defined in the Configure Queue dialog is by default a primary queue. Queue 100, a reserved queue, is labeled Reserved.

If you expect to see activity in the messages sent and received per second fields, but see 0.00 instead, try resetting the statistics. If an attached queue is inactive for a significant amount of time, the messages sent/received per second values approach zero because the value is derived as an average since statistics were last reset. Unless statistics are reset, these fields will not give an accurate assessment of recent queue activity.

Viewing Queue Traffic Detail

Use the Queue Information and Traffic Detail menu items to view the message and byte traffic rates of a single queue. Table 4-7 describes the fields of the Queue Traffic Detail dialog.

Table 4-7 Queue Traffic Detail Dialog

Statistic	Description
Queue Number	Number of the queue
Queue Attributes	The type of queue, permanent or temporary. Any queue defined in the Configure Queue dialog is by default a primary queue. Queue 100, a reserved queue, is labeled Reserved.
Messages Sent per Second	Average number of messages sent per second since statistics were last reset
Messages Received per Second	Average number of messages received per second since statistics were last reset
Bytes Sent per Second	Average number of bytes sent per second since statistics were last reset
Bytes Received per Second	Average number of bytes received per second since statistics were last reset
Peak Messages	Highest number of messages in the queue at any given time since statistics were last reset
Peak Bytes	Highest number of bytes in the queue at any given time since statistics were last reset

Viewing Queue Detail

Use the Queue Information and Detail screen to view the attributes of a selected queue. If quotas have been enabled, they can be viewed from this dialog. Queue Information and Detail shows the last time a queue was attached and detached, which can be useful for determining whether or not your application is truly interacting with a queue.

Table 4-8 describes the fields of the Queue Detail dialog.

Table 4-8 Queue Detail Dialog

Statistic	Description
Queue Number	Number of the queue
Queue Name	Name of the queue
Primary Queue	If the queue is secondary, the queue number of the controlling queue
Queue Attributes	The type of queue, permanent or temporary. Any queue defined in the Configure Queue dialog is by default a primary queue. Queue 100, a reserved queue, is labeled Reserved.
Owner PID	Process ID of the attached queue
Total Messages	Cumulative messages sent and received from the time statistics were last reset
Total Bytes	Cumulative bytes sent and received from the time statistics were last reset
Message Quotas	The maximum limit of uncollected messages that can reside in this queue, and the current number. This value is set by the Messages Quota parameter in the %QCT section of the group initialization file.
Byte Quotas	The maximum limit of uncollected bytes that can reside in this queue, and the current number. This value is set by the Bytes Quota parameter of the %QCT section of the group initialization file.

5 Troubleshooting

This chapter provides information to help you diagnose problems with your MessageQ application.

Error Logging

MessageQ has an error logging mechanism to display and capture informational, warning, and error messages that can occur during processing. The messages display a description of the condition to help developers gather more information about failure conditions within a message queuing group.

All logging information goes to the log file specified when the system is started using the `-l` option. If not specified, the logging information goes to the standard output on UNIX systems and to `dmqlog`<*busid*>_<*groupid*>`.log` in the directory in which the system is started on Windows NT. On Windows NT, when the system is started using the System Service or `dmqmonw`, the `-l` option defaults to `DMQLOG`<*busid*>_<*groupid*>`.LOG` in the `winnt/system32` directory (this default cannot be changed).

When using the MessageQ Client for Windows, all information logged goes to `dmqerror.log` in the current directory.

Tracing

MessageQ provides an execution tracing program for diagnostic purposes. Tracing produces a time-stamped output file showing the sequence of MessageQ function calls and return status codes. If the DMQ_TRACE_PREFIX environment variable is set, tracing information goes to $DMQ_TRACE_PREFIX.pid. If it is not set and the DMQ_TRACE_FILE environment variable is set, then tracing information will go to $DMQ_TRACE_PREFIX. Otherwise, tracing information will go to the standard output (this is not desirable on Windows NT). Each message will contain a time stamp if the DMQ_TRACE_TIMESTAMPS environment variable is set.

Client message tracing (sent and received messages) that is configured using dmqclconf on UNIX systems or dmqconf on Windows NT systems is written to the file dmqtrace.log in the current working directory.

Because tracing generates a high volume of output, you should enable tracing only for diagnostic purposes in the event of a problem. The following levels of tracing are provided:

Level	Description	Logfile
API	Traces API calls and provides information on the messages sent or received	♦ $DMQ_TRACE_PREFIX.pid if DMQ_TRACE_PREFIX is set ♦ $DMQ_TRACE_PREFIX if DMQ_TRACE_FILE is set
Internal	Traces execution of the lower software layers	♦ $DMQ_TRACE_PREFIX.pid if DMQ_TRACE_PREFIX is set ♦ $DMQ_TRACE_PREFIX if DMQ_TRACE_FILE is set
CLS	Traces DMQCL.DLL execution events	DMQTRACE.LOG

Enabling Tracing

To enable tracing at MessageQ startup, set the environment variable (prior to invoking the MessageQ Group Control Process), as follows:

```
SET PAMS_TRACE=value
SET DMQIPI_TRACE=value
SET DMQCL_TRACE=value
```

where *value* is an arbitrary value (such as 1).

Disabling Tracing

To disable tracing, set the environment variables to a null value as follows:

```
SET PAMS_TRACE=
SET DMQIPI_TRACE=
SET DMQCL_TRACE=
```

You can check your environment variables at any time by entering SET at the command line.

Index

A

Application Programming Interface (API)
    tracing 5-2

B

Backing up the system disk 1-3

C

Changing group characteristics at runtime 3-18
Client Library Server (CLS)
    restricting remote access 3-13
    starting 3-12
    stopping 3-13
Configuring message recovery 2-15
Configuring MessageQ 2-1
    overview 2-1
Configuring the network 1-15
Connecting to the MessageQ Environment 3-2
    from the Monitor Utility 3-3
Connecting to the MessageQ environment 3-2
    from the Monitor Utility 3-3
Creating a custom group configuration 2-8

D

Defining
    name-to-queue translations 2-36
Defining timeout intervals for link drivers 3-21
Directory
    adding to your path 1-16
Documentation 1-13
Drive
    adding to your path 1-16

E

Enabling cross-group connections 2-21
Enabling queue quotas for selective queues 3-20
Enabling quotas 3-19, 4-2
Enabling statistics 4-2

I

Installation prerequisites 1-3
Installation requirements 1-3
Installing MessageQ Server for Windows NT 1-5
Installing the MessageQ software 1-3
    choosing installation options 1-4
    on systems already running MessageQ 1-4

M

Managing MessageQ 3-1
Message queues 2-3
Message Queuing Bus 2-2

Message queuing groups 2-3
MessageQ
    backing up the system disk 1-3
    configuring 2-1
    configuring message recovery 2-15
    configuring the Client Library Server 2-19
    distribution media 1-1
    managing the system 3-1
    Preparing to install the software 1-1
    setting bus and group IDs 3-2
    using the Monitor utility 3-1
Monitoring MessageQ 4-1
Monitoring the MessageQ system 3-1

N

Naming 2-5

P

Postinstallation Tasks 1-14
Preparing to install the software 1-1
Printing journal files 3-16

R

Recovering from installation errors 1-13
Replaying journal messages 3-15
Restricting remote access to CLS 3-13
Running the CLS 3-12

S

Setting bus and group IDs 3-2
Setting group characteristics 2-12
Setting quotas for dynamic queues 3-20
Shutting down a group 3-8
Software distribution media 1-1, 1-2
Starting a cross-group connection 3-10
Starting a group 3-5
    from the command line 3-5
    from the Monitor utility 3-6
    from the startup program group 3-7
Starting a link 3-10
Starting a message queuing group 3-5
Starting a queue 3-11
Starting the MessageQ Service for Windows NT 1-15
Stopping a cross-group connection 3-10
Stopping a link 3-10
Stopping a queue 3-11
support
    technical xiii

T

Tracing 5-2
    disabling 5-3
    enabling 5-2

U

Using the online user documentation 1-13
Utilities
    test 2-42

V

Viewing link and queue activity 4-3
Viewing link detail 4-6
Viewing link traffic counts 4-3
Viewing link traffic detail 4-5
Viewing link traffic rates 4-4
Viewing queue detail 4-10
Viewing queue traffic counts 4-7
Viewing queue traffic detail 4-10
Viewing queue traffic rates 4-8