BEA MessageQ

Installation and Configuration Guide for UNIX

BEA MessageQ for Unix Version 5.0
Document Edition 3.0
October 1998

BEA MessageQ Installation and Configuration Guide for UNIX

Document Edition	Part Number	Date	Software Version
3.0	850-001005-003	October 1998	BEA MessageQ, Version 5.0

Contents

Preface

1. Installing MessageQ

2. Configuring UNIX System Resources for MessageQ

3. Configuring MessageQ

4. Managing MessageQ

5. Monitoring MessageQ

6. Troubleshooting

A. Uninstalling MessageQ V3.x and V4.x

Preface

Purpose of This Document

This document provides information on installing and configuring BEA MessageQ on UNIX 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 UNIX is organized as follows:

- Chapter 1, “Installing MessageQ,” describes preinstallation, installation, and postinstallation tasks.
- Chapter 2, “Configuring UNIX System Resources for MessageQ,” describes how to configure UNIX system resources, such as processes, shared memory, semaphores, and messages, to accomodate MessageQ software.

- Chapter 3, “Configuring MessageQ,” provides instructions on configuring a message queuing group, defining parameters in a group initialization file, and testing the group configuration.
- Chapter 4, “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, changing group characteristics at runtime, and enabling and disabling quotas.
- Chapter 5, “Monitoring MessageQ,” describes how to enable, disable, and reset statistics, and how to use the Monitor utility to view link and queue activity.
- Chapter 6, “Troubleshooting,” describes how to use the logging and tracing capabilities to diagnose problems with your MessageQ application.
- Appendix A, “Uninstalling MessageQ V3.x and V4.x,” lists the files installed on your system during the MessageQ for UNIX installation.

How to Use This Document

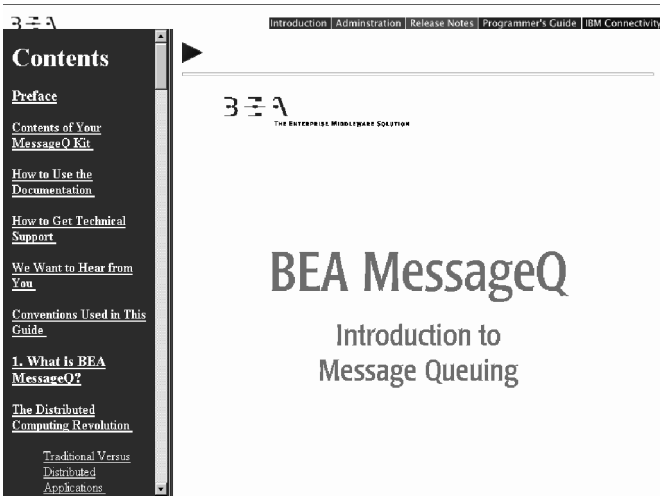
This document is designed primarily as an online, hypertext document. If you are reading this as a paper publication, note that to get full use from this document you should access it as an online document via the BEA MessageQ Online Documentation CD. The following sections explain how to view this document 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 later, or Microsoft Internet Explorer version 3.0 or later are recommended.

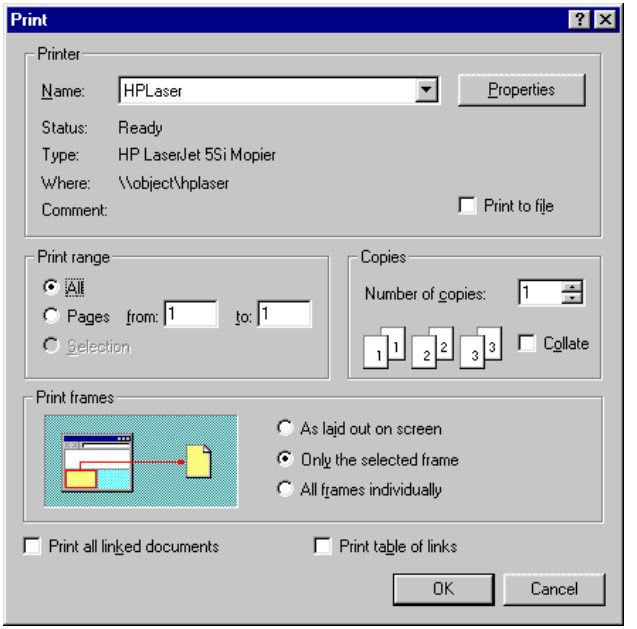
Figure 1 Online Document Displayed in a Netscape Web Browser



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.

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*: `int32` **`pams_exit`**`( )`
`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*: `dmqshutdown -b` *`integer`* `-g` *`integer`* `[-f]`
`|`	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 section lists 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 Installation and Configuration for Windows NT

BEA MessageQ System Messages

BEA MessageQ Client for UNIX User's Guide

BEA MessageQ Client for Windows User's Guide

BEA MessageQ FML Programmer's Guide

BEA MessageQ Reference Manual

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.

Related BEA Documentation

BEA MessageQ can be installed with an existing BEA TUXEDO V6.4 or BEA M3 V2.1 installation. This allows transparent forwarding of messages from a MessageQ application to a TUXEDO queue or service. You will need to refer to the applicable BEA TUXEDO documentation set for information about TUXEDO concepts, functions, and system messages.

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 BEA MessageQ, or if you have problems installing and running BEA MessageQ, 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 license information
- 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 UNIX systems:

- Preparing to Install the MessageQ Software
- Installing the MessageQ Software on UNIX Systems
- Performing Postinstallation Tasks

Preparing to Install the MessageQ Software

Before you install MessageQ, there are several installation tasks you should perform to ensure that the installation runs smoothly and the software is installed correctly. They are:

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 installation subsets you want to install.

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-ROM readers, labeled MessageQ for UNIX, Version 5.0.

Hardcopy and Online User Documentation

Your kit contains the following hardcopy documentation:

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

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

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

MessageQ provides this online documentation in HTML format which you can read using any World Wide Web browser. For example, Digital UNIX includes the Netscape Navigator Web browser. For more information on how to access the online documentation, refer to the section on Using the MessageQ Online Documentation.

Complying with the Installation Requirements

You may need to be able to log in as root (superuser) on the system where you are installing MessageQ if your system requires you to mount the cdrom drive as root. Otherwise, unless you need to uninstall a version of MessageQ prior to V5.0, or you want your new MessageQ installation to be owned by root, there is no need to be logged in as root to install MessageQ.

Any installation of BEA MessageQ with a version less than 5.0 must be uninstalled prior to using BEA MessageQ 5.0 or later. Prior to V5.0, only one version of BEA MessageQ was allowed to be installed on any single machine at a time, and the installation procedure symbolically linked files into the `/usr/bin`, `/usr/lib`, `/usr/man`, etc. directories. Starting with V5.0, installation files are no longer placed in any subdirectory of `/usr`, and multiple versions may be installed one a single machine as long as they are all V5.0 or later. If you do not uninstall MessageQ installations with versions less than 5.0 before you attempt to use V5.0 or later installations of MessageQ, you may experience serious operational problems. See Appendix A, "Uninstalling MessageQ V3.x and V4.x" for instructions on uninstalling older versions of MessageQ.

Your system must also meet the hardware, software, and disk space requirements listed in this manual.

Installing the MessageQ Software on UNIX Systems

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

- Choosing Which Installation Subsets to Install
- Installing the MessageQ Server for UNIX
- Installing the MessageQ Client for UNIX
- 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. Appendix A lists the files and directories created during the installation.

Note: MessageQ provides a conversion utility that lets you convert your an existing MessageQ for UNIX, Version 3.2B group initialization file to run under MessageQ for UNIX, Version 5.0. For a description of the conversion utility, see the Converting Your Version 3.2B Group Initialization File to Version 5.0 topic.

Group initialization files from BEA MessageQ V4.0 and V4.0A are compatible with BEA MessageQ V5.0 and require no conversion.

Choosing Installation Options

The MessageQ installation dialogues display a list of options that let you choose installation options. If you choose a package or option and then decide you don't want to install it, you can exit the installation and start over. The installation options include:

- installing the MessageQ product or the MessageQ online documentation
- installing with or without BEA TUXEDO or BEA M3
- installing the client and server, or just the client

Table 1-1 describes the installation packages.

Table 1-1 MessageQ Packages

Installation Subset	Description
MessageQ	MessageQ software for UNIX platforms.
MessageQ Online Documentation	MessageQ Online Documentation in HTML format.

Table 1-2 describes the installation options for connecttivity with BEA TUXEDO..

Table 1-2 MessageQ and TUXEDO Installation Options

Installation Option	Description
BEA TUXEDO 6.4	Install MessageQ with an existing BEA TUXEDO 6.4 installation.
BEA M3 2.1	Install MessageQ with an existing BEA M3 2.1 installation
None	Install MessageQ as a standalone product, with no BEA TUXEDO or BEA M3 connectivity.

Table 1-3 describes the installation options.

Table 1-3 MessageQ Installation Options

Installation Subset	Description
Both client and server	Installed on licensed systems used for developing or running MessageQ applications. This option includes the Client Library Server, include files, and examples. The programming examples illustrate how applications can use interprocess message queuing to exchange information. For more information about the programming examples, refer to the *Programmer's Guide*.
Client Library	Provides remote client applications access to message queuing using MessageQ for UNIX. This option also installs the MessageQ include files. During the MessageQ installation procedure, you can choose to install the MessageQ Client for UNIX by selecting the Client Library installation option. The MessageQ Client for UNIX allows applications running in a UNIX environment to send and receive messages to target applications in a networked environment using the MessageQ Client Library Server (CLS) software that is running on a MessageQ server. The MessageQ Client for UNIX provides applications with full support of MessageQ features without requiring the system resources needed by a MessageQ UNIX message server that supports full message routing. User applications designed as clients or servers can be deployed on systems running MessageQ Client software. The *MessageQ Client for UNIX User's Guide* contains more information about the MessageQ Client for UNIX.

Installing the MessageQ Server for UNIX

This topic describes the steps for installing MessageQ development software on most UNIX systems. You can stop the installation procedure at any time by using the `-q` option or the terminal interrupt key sequence for your UNIX system (see your system user documentation set for a description of the terminal interrupt key). If you stop the installation, files installed up to that point are not deleted automatically; you must delete these files manually.

The steps for installing MessageQ development software are as follows:

1. Mount the CD-ROM media.

   Place the CD-ROM media in the CD-ROM tray and close the door. If your system automatically mounts your CD-ROM media when the door is closed, you may proceed to the next step.

   Depending on your system, you may have to manually mount the CD-ROM media. You may have to do this logged in as root (superuser). If you must log in as root to mount the CD-ROM media, do so now.

   If you do not have a standard procedure or tool for mounting CD-ROM media, use the following table of platform specific syntax information::

AIX	`mount -v cdrfs -r` *`device directory`*
Digital UNIX	`mount -t cdfs -r` *`device directory`*
HP-UX	`mount -F cdfs -r` *`device directory`*
NCR MP-RAS	`mount -F cdfs -r` *`device directory`*
SCO Unixware	`mount -r` *`device directory`*
SCO OpenServer	`mount -r` *`device directory`*
Sequent DYNIX	`mount -r` *`device directory`*
Solaris	`mount -t hsfs -r` *`device directory`*

Note: When installing the MessageQ Client for UNIX on Solaris systems, ensure that the lsocket and lnsl libraries are present on the system. These libraries are required when compiling.

2. Run the installation script.

   Note that it is not necessary to be logged in as root (superuser) to install MessageQ as long as you have permission to write to the directory where you direct the installation script to install MessageQ.

   Before you run the installation script, move to the directory where the CD-ROM media is mounted. Assuming you mounted the CD-ROM media on directory `/cdrom`, you would issue the following command:

   ```
   %cd /cdrom
   ```

   (If your CD-ROM media was automatically mounted, or you used a different tool or process than outlined in the previous step, you may have to move to a subdirectory of the mount point in order to successfully install MessageQ.)

   After you have moved to the appropriate directory, determine the case of the installation script name. This depends on what options or tools you used to mount your CD-ROM media. You are looking for a file named install.sh or INSTALL.SH. Issue the following command to determine whether the script name is lower case or upper case:

   ```
   %ls
   ```

   If the script name is in lower case letters, issue the following command:

   ```
   % sh ./install.sh
   ```

   otherwise, issue the command like this:

   ```
   % sh ./INSTALL.SH
   ```

   A list of supported platforms (operating system and machine or processor type) is displayed.

   ```
   01) aix41/rs6000     02) aix42/rs6000     03) aix43/rs6000
   04) dux/alpha        05) dynix/i386       06) hpux10/hppa
   07) hpux11/hppa      08) mpras/x86        09) sco/x86
   10) sco_uw/x86       11) sol251/sparc     12) sol26/sparc
   ```

   The following table provides detailed platform information :

`aix41/rs6000`	IBM AIX 4.1.4, RS/6000 SP2
`aix42/rs6000`	IBM AIX 4.2.1, RS/6000 SP2
`aix43/rs6000`	IBM AIX 4.3, RS/6000 SP2

dux/alpha	Digital UNIX 4.0, Alpha
dynix/i386	Sequent DYNIX ptx 4.4.1, Intel 386
hpux10/hppa	HP HP-UX 10.2, HPPA
hpux11/hppa	HP HP-UX 11, HPPA
mpras/x86	NCR MP-RAS 3.01 or 3.02, Intel x86
sco/x86	SCO OpenServer 5, Intel x86
sco_uw/x86	SCO UnixWare 2.1, Intel x86
sol251/sparc	Sun Solaris 2.5.1, SPARC
sol26/sparc	Sun Solaris 2.6, SPARC

3. Enter the number next to the selected installation platform:

```
Install which platform's files? [01-   12, q to quit, l for list]: 4

** You have chosen to install from dux/alpha **
```

4. Confirm your choice of platform:

```
BEA MessageQ 5.0

This directory contains the BEA MessageQ Core System for
Digital Unix 4.0 on DEC Alpha.

Is this correct? [y,n,q]: y
```

The following MessageQ packages are listed.

```
1   msgq          BEA MessageQ
2   msgqdoc       BEA MessageQ Online Documentation
```

5. Select the packages you wish to install:

```
Select the package(s) you wish to install (or 'all' to install
all packages) (default: all) [?,??,q]: all
```

Copyright and trademark information is displayed:

```
BEA MessageQ
(alpha) Release 5.0
Copyright (c) 1998 BEA Systems, Inc.
Portions * Copyright 1986-1998 RSA Data Security, Inc.
All Rights Reserved.
Distributed under license by BEA Systems, Inc.
MessageQ is a registered trademark of BEA Systems, Inc.
TUXEDO is a registered trademark.
```

6. Select one of the following connectivity options in respect to other BEA Systems products:

```
The following connectivity options are available:

  1     tux64           Install On Top Of BEA TUXEDO v6.4
  2     m3_21           Install On Top Of BEA M3 v2.1
  3     none            Install Without BEA TUXEDO

Select an option (default: none) [?,??,q]: 2
```

Note: The BEA M3 V2.1 installation option is only available on platforms where BEA M3 V2.1 is supported.

7. Select one of the following installation options with respect to client and server environments:

```
The following installation options are available:

  1     both            Server and client
  2     client          Client only

Select an option (default: both) [?,??,q]: both
```

8. Specify the directory where MessageQ files are to be installed:

```
Directory where BEA M3 files are installed [?,q]: /opt/messageq
Creating /opt/messageq
```

The system determines if sufficient space is available in the installation directory, unloads the files for the selected installation options, sets file permissions, and processes the license file:

```
Determining if sufficient space is available ...
11794 blocks are required
6186634 blocks are available to /opt/messageq

Using /opt/messageq as the installation base directory
.
.
.
Changing file permissions...
.. finished

Processing default license file...
.. finishel
```

Caution: You should carefully read the section on licensing before you proceed so that you do not disable other BEA products.

```
Please don't forget to manually copy the contents of the license
file from the enclosed floppy to '/opt/messageq/udataobj/lic.txt'.
Refer to the Installation Guide for details on how to do this.

Installation of BEA MessageQ was successful
```

At this point, the online documentation is installed.

```
BEA MessageQ Online Documentation
(alpha) Release 5.0
Copyright (c) 1998 BEA Systems, Inc.
Portions * Copyright 1986-1998 RSA Data Security, Inc.
All Rights Reserved.
Distributed under license by BEA Systems, Inc.
MessageQ is a registered trademark of BEA Systems, Inc.
TUXEDO is a registered trademark.

Base directory for documentation installation (default:
/opt/messageq) [?,q]:

Determining if sufficient space is available ...
27810 blocks are required
6186632 blocks are available to /opt/messageq

Using /opt/messageq for the documentation base directory

Installing documentation...
```

```
Changing file permissions...
... finished

Installation of BEA MessageQ Online Documentation was successful

Please don't forget to fill out and send in your registration card
```

Before you unmount your CD-ROM media, don't forget to move out of the /cdrom directory or you will get a message that the /cdrom device is busy:

```
% cd /
% umount /cdrom
```

After the software is installed, you may want to perform the postinstallation tasks described in the Performing Postinstallation Tasks topic.

Licensing BEA MessageQ

There are two cases for license installation. You must follow the instructions below for your particular installation or you may disable other licensed BEA Systems products

Licensing BEA MessageQ Installed Over BEA TUXEDO or BEA M3

If you have installed BEA MessageQ on top of either a BEA TUXEDO or BEA M3 installation, you must append your BEA MessageQ license file (`lic.txt`) to the license file (also `lic.txt`) found in the `udataobj` directory of the installed products. A recommended procedure for doing this follows:

1. Change directory to the BEA TUXEDO or BEA M3 license directory. For example:

```
% cd <directory>/udataobj
```

2. Rename the existing BEA TUXEDO or BEA M3 license file:

```
% mv lic.txt lic.tuxedo
```

3. Move your MessageQ license file from the floppy media and put it into the `udataobj` directory. The MessageQ license file is also named `lic.txt`.

4. Rename the BEA MessageQ license and append the BEA MessageQ license file to the BEA TUXEDO or BEA M3 license file, creating a new `lic.txt` file:

```
% mv lic.txt lic.messageq
% cat lic.tuxedo lic.messageq > lic.txt
% chmod 444 lic.txt
```

Licensing BEA MessageQ as a Standalone Product

If you have not installed BEA MessageQ on top of either BEA TUXEDO or BEA M3, move your MessageQ license file (`lic.txt`)from the floppy media and put it into the `udataobj` directory of the BEA MessageQ installed product. After you have put the MessageQ `lic.txt` file into the `udataobj` directory, continue as follows:

```
% cd <directory>/udataobj
% chmod 444 lic.txt
```

Recovering from Installation Errors

Errors can occur during the installation if the following conditions exist:

- The operating system version is not supported by MessageQ software.
- Sufficient disk space is not available.
- You do not have write privileges to the directory into which you attempted to install MessageQ.

For descriptions of error messages generated by these conditions, see the system management user documentation for the UNIX system on which you are running. If an error occurs while installing MessageQ and you believe the error is caused by a problem with MessageQ, call BEA Technical Support at the number provided in the Preface.

Performing Postinstallation Tasks

After completing the installation procedure, you may want to perform several recommended postinstallation tasks. MessageQ does not require you to perform the postinstallation tasks in a particular order.

The postinstallation tasks are as follows:

- Using the MessageQ Online Documentation — Optional
- Convert group configuration files to Version 5.0 — Required

  Note: MessageQ provides a conversion utility that lets you convert your an existing MessageQ for UNIX, Version 3.2B group initialization file to run under MessageQ for UNIX, Version 5.0. For a description of the conversion utility, see the Converting Your Version 3.2B Group Initialization File to Version 5.0 topic.

  Group initialization files from BEA MessageQ V4.0 and V4.0A are compatible with BEA MessageQ V5.0 and require no conversion.

- Configure the network — Optional

Using the MessageQ Online Documentation

MessageQ includes online documentation in HTML format that can be read using a World Wide Web browser. If your UNIX system does not currently have a Web browser installed, you can obtain a copy of a Web browser free of charge or for a nominal fee from the Internet.

In addition, most PC environments now include a Web browser. Therefore, if the UNIX system that you are using does not include a Web browser for you to view the MessageQ documentation, you can also can view the documentation using a PC that is networked to a UNIX system running MessageQ.

To read the online user documentation, follow these steps:

1. Install the optional subset, if not previously installed. The subset is labeled:
   `msgqdoc  BEA MessageQ Online Documentation`

2. Invoke your Web browser.
3. Use the "Open File" option of your Web browser and open the following file:

   `/install_dir/doc/bmq/v5_0/index.htm`

4. Begin navigating the online documentation using the hyperlinks. Note that each book or category of book in the online documentation set is listed and accessible from the `index.htm` page. The first page in each book is the table of contents, shown in the lefthand frame, and the title page, in the righthand frame. 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. You can also click on the BEA Systems logo to jump to the bookshelf 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.

Converting Your Version 3.2B Group Initialization File to Version 5.0

MessageQ for UNIX provides a conversion utility that lets you quickly convert your existing MessageQ Version 3.2B group initialization file so that it is compatible with the Version 5.0 software. (MessageQ Version 4.0 and 4.0A group initialization files do not need to be converted to work with Version 5.0.) The conversion utility checks to make sure that the current sections and parameters in the group initialization file are appropriate for use with MessageQ for UNIX, Version 5.0.

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 change and use the new Version 5.0 features, refer to Chapter 3, which describes all the sections and parameters in the group initialization file.

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

1. Change directory as follows:

```
# cd /install_dir/bin/
```

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

```
# dmqconvert -f inputfile > outputfile
```

Where:

`-f`	The pathname and file specification of the group initialization file you want to convert.
`>`	Uses the UNIX redirect command to create the converted Version 5.0 initialization file. **Note:** Do not specify the input file name for your output file. If the output file name is the same as the input file name, your input file will be overwritten.

Listing 1-1 and Listing 1-2 provide examples of the %QCT section of a group initialization file prior to and after conversion by the `dmqconvert` utility.

Listing 1-1 QCT Section Prior to Conversion

```
%QCT

TEMPLATE 0      64000   100     P       0       N       L
QUEUE1  1       64000   100     P       0       N       L
QUEUE2  2       64000   100     P       0       Y       G
QUEUE3  3       64000   100     P       0       Y       L
QUEUE4  4       64000   100     M       0       Y       L
QUEUE5  5       64000   100     S       3       N       L
QUEUE6  6       64000   100     S       3       Y       L
QUEUE7  7       64000   100     S       0       N       L


QUEUE10 10      64000   100     P       0       N       L
QUEUE11 11      64000   100     S      10       N       L
QUEUE12 12      64000   100     S      10       N       L
```

```
! queues with differing quotas. quotas can be turned on
! using the monitor programs (dmqmonc and dmqmonm).

QUEUE13 13       8000    25     P       0       N       L
QUEUE14 14      16000    50     P       0       N       L
QUEUE15 15      32000   100     P       0       N       L


! queues that require explicit confirmation and allow out of
! order confirmation.

QUEUE16 16      64000   100     PE      0       N       L
QUEUE17 17      64000   100     PEO     0       N       L


! the example_q_n queues are used by the example programs
! provided in the MessageQ Example Programs subset

example_q_1 193 64000   100     P       0       N       L
example_q_2 194 64000   100     P       0       N       L


! The DEAD_LETTER_QUEUE is the target for  UMA modes of
! xx_DLQ. A user process may attach to this queue to collect
! undeliverable messages for application-specific processing

DEAD_LETTER_QUEUE 96 100000 500 P       0       N       L

%EOS
```

Listing 1-2 QCT Section After Conversion

```
%QCT

TEMPLATE    0   64000   100   None  .  P    0  .  N    L    N
QUEUE1      1   64000   100   None  .  P    0  .  N    L    N
QUEUE2      2   64000   100   None  .  P    0  .  Y    G    N
QUEUE3      3   64000   100   None  .  P    0  .  Y    L    N
QUEUE4      4   64000   100   None  .  M    0  .  Y    L    N
QUEUE5      5   64000   100   None  .  S    3  .  N    L    N
```

```
QUEUE6      6   64000   100   None  .  S    3  .  Y    L    N
QUEUE7      7   64000   100   None  .  S    0  .  N    L    N


QUEUE10     10  64000   100   None  .  P    0  .  N    L    N
QUEUE11     11  64000   100   None  .  S    10 .  N    L    N
QUEUE12     12  64000   100   None  .  S    10 .  N    L    N


! queues with differing quotas.   quotas can be turned on
! using the monitor programs (dmqmonc and dmqmonm).

QUEUE13     13  8000    25    None  .  S    10 .  N    L    N
QUEUE14     14  16000   50    None  .  P    0  .  N    L    N
QUEUE15     15  32000   100   None  .  P    0  .  N    L    N


! queues that require explicit confirmation and allow out
! of order confirmation.

QUEUE16     16  64000   100   None  .  P    0   EI   N    L     N
QUEUE17     17  64000   100   None  .  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     64000   100     None  .   P    0  .  N    L   N
example_q_2     194     64000   100     None  .   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
! undeliverable messages for application-specific  proces-
! sing

DEAD_LETTER_QUEUE    96   100000   500   None  .  P   0  .  Y    L    N

%EOS
```

Configuring the Network

If you plan to use MessageQ for UNIX to communicate with MessageQ on other systems, you must have configured TCP/IP networking and started the network service.

The remote systems with which you intend to communicate must have their names and addresses entered in an Internet host database:

- If a Network Information Service (NIS) is used, the systems must be defined in the NIS host database.
- If a Domain Name Server (DNS) is used, the systems must be known to the DNS.
- otherwise, the systems must be defined in `/etc/hosts`

For information on configuring and starting the network service, refer to the system user documentation for your particular platform.

2 Configuring UNIX System Resources for MessageQ

This chapter contains the following topics that describe how to configure UNIX system resources to accommodate MessageQ software:

- Overview of Configuring UNIX Resources for MessageQ
- Calculating Process Resources
- Calculating Shared Memory Segment Resources
- Location of Shared Segments
- Calculating Semaphore Resources
- Calculating Message Resources
- Rebuilding the Kernel and Rebooting the System
- Verifying System Resources

Overview of Configuring UNIX Resources for MessageQ

MessageQ for UNIX software installs on all supported UNIX systems regardless of kernel configuration parameters. However, MessageQ applications using multiple buses, queues, and groups will probably require system resources beyond those currently allocated for your system.

After installing MessageQ for UNIX systems, you must calculate the kernel resources your MessageQ implementation requires and change those resource parameters. Your system must have sufficient resources before you configure MessageQ message queuing buses and groups. To configure your UNIX system to support MessageQ software, you must:

1. Calculate system resources and edit the kernel's configuration to change those resources.
2. Rebuild the kernel if necessary and then reboot the system.

   Note: Whenever you change any system parameters in the kernel configuration file, you may need to modify the kernel and reboot your system to ensure those modifications take effect. For information on modifying and rebooting the kernel configuration file, see the Modifying the Kernel and Rebooting the System topic.

3. Verify system resources.

To properly configure your system, you need good working knowledge of the UNIX system on which the MessageQ software will be running. You also need access to your system's user documentation because you will reference it frequently. Table 2-1 lists the UNIX system resources affected by the MessageQ configuration.

Table 2-1 MessageQ for UNIX Systems Resources

System Resources	Associated Parameters
Processes	MAXUPRC
Shared memory segments	SHMMNI, SHMSEG, SHMMAX

Table 2-1 MessageQ for UNIX Systems Resources

System Resources	Associated Parameters
Semaphores	SEMMNI, SEMMNS, SEMMNU, SEMMSL, SEMOPM, SEMUME
Messages	MSGMAX, MSGMNB, MSGMNI, MSGSEG, MSGSSZ, MSGTQL

To accommodate your proposed MessageQ configuration, you might have to edit the kernel configuration and change the current system parameters. This is usually done with a system administration tool or by editing a configuration file. To change system parameters on UNIX platforms, see the user documentation for your particular system.

Calculating Process Resources

MessageQ for UNIX systems is implemented as a set of discrete system processes that communicate by way of shared memory, semaphores, and interprocess communication message queues. You must also ensure that your system has sufficient process slots for MessageQ to run properly. Calculate the process slots required by MessageQ for UNIX systems as follows:

Number of Process Slots	For Each...
3	Group
1	Group with Message Recovery Services (MRS) enabled
2	Cross-group connection
1	CLS process
1	Group with Selective Broadcast Services (SBS) enabled
1	Group using global naming

For example, you would calculate the process resources for a simple configuration with one bus, one group (X) with a cross-group connection, SBS and MRS enabled, one link to a group (Z) on another system, and a local naming agent as follows:

```
3 process slots per group                       = 3

1 process slot per group with MRS enabled       = 1

1 process slot per group with SBS enabled       = 1

1 process slot per group with naming enabled    = 1

2 process slots per cross-group connection      = 2
    (Group X has 1 link to Z)

Total DMQ_process_slots                         = 8
```

A more complex configuration could consist of one bus and three groups labeled A, B, and C. All three groups have cross-group messaging enabled, with each group connected to the other two. Group A has MRS enabled. The number of processes required would be calculated as follows:

```
3 process slots per group                       = 9

1 process slots per group with MRS enabled      = 1

2 process slots per cross-group connection      = 12

    (Group A has 2 links to B and C = 4)

    (Group B has 2 links to C and A = 4)

    (Group C has 2 links to A and B = 4)

Total DMQ_process_slots                         = 22
```

To determine the system configuration parameter MAXUPRC (the maximum number of process slots needed), use the following equation:

```
new_maxuprc = DMQ_process_slots + margin
```

Adding a margin to the required MessageQ process slots allows you to make small adjustments to your MessageQ configuration without having to reconfigure the kernel immediately.

Note: Do not change the configuration parameter MAXUPRC if `new_maxuprc` is less than the current value of MAXUPRC.

Calculating Shared Memory Segment Resources

MessageQ uses shared memory segments to store the information needed to manage the messaging environment. You need to calculate the number, size, and location of the shared segments for your MessageQ for UNIX systems configuration.

Number of Shared Segments

User programs and MessageQ utilities must be able to map the MessageQ shared memory segments, as well as any other shared memory segments specific to a user application. For each group in your MessageQ configuration, you should have 6 shared memory segments. Each application will attach to a specified number of shared memory segments.

You will also need to specify the maximum number of shared memory identifiers available to support your entire system configuration. Shared memory identifiers are set using the system configuration parameter SHMMNI. To determine the new value of SHMMNI, calculate the total number of shared segments for all buses and groups, include a margin for expansion, and add this total to the current value of SHMMNI. The formula is as follows:

```
new_shmmni = current_shmmni + margin + (6 *  total_groups)]
```

Adding a margin to the system configuration parameter SHMMNI allows you to make small adjustments to your MessageQ configuration without having to reconfigure the kernel immediately. Setting the margin arbitrarily high might cause resource problems; setting the margin too low might prevent MessageQ from working correctly.

Size of Shared Segments

The maximum permitted shared memory segment size must be large enough for the largest MessageQ shared memory segment, which (in most cases) is the queue information segment. Calculate the size (in bytes) of the queue information segment as follows:

```
queue_info_segment = 404 *  (3999 + 2)
                   [which equals 1616404]
```

You need to provide for shared segments up to 1,616,404 bytes. Using the allocation size for your system and the queue information segment, you can set the configuration parameter SHMMAX by using the following formula:

```
new_shmmax = (1616404 / alloc_size) + margin
```

Round up the result to the nearest whole number before adding the margin. For example, divide 1,616,404 (`queue_info_segment`) by 4096 (`alloc_size`), which comes out to 394.62988. Round up the result to 395 and add a margin of 30 to get a value of 425 for the configuration parameter SHMMAX. Adding the margin to SHMMAX allows you to make small adjustments to your MessageQ configuration without having to reconfigure the kernel immediately.

Note: Do not change the configuration parameter SHMMAX if `new_shmmax` is less than the current value of SHMMAX.

Location of Shared Segments

MessageQ maps shared memory by using the system default mapping. This means the system selects the address at which the shared memory will be mapped. Programs that dynamically expand their private data space can encounter problems if the default mapping location is not set sufficiently high.

Use the system configuration parameter SHMBRK to set the distance between private and shared data space. For most systems, setting the parameter to 4 megabytes is adequate. Note that on systems where SHMBRK is expressed in pages, divide the number of bytes by the page size to get the parameter value.

For example, the MRS journal process requires a minimum of 4 megabytes of heap space between the application code and the shared memory segments. If your application uses recoverable messaging, you may have to raise the SHMBRK parameter accordingly.

Calculating Semaphore Resources

MessageQ uses semaphores to serialize access to shared data structures. You must have sufficient available semaphore resources for MessageQ to operate properly. The three resources you need to calculate are:

- SEMMNI—The number of semaphore identifiers required by MessageQ software on any given system. Calculate SEMMNI as follows:

  ```
  SEMMNI = (number of groups * 4)
  ```

- SEMMNS—The number of semaphores required by MessageQ software on any given system. Calculate SEMMNS as follows:

  ```
  SEMMNS = SEMMNI * 3
  ```

- SEMMNU—The number of undo structures required by MessageQ software on any given system. Use the number of MessageQ process slots you calculated in Calculating Process Resources to calculate SEMMNU, as follows:

  ```
  SEMMNU = DMQ_process_slots + number_of_application_processes
  ```

For example, setting the following values are adequate for most applications:

Parameter	Recommended Value	Description
SEMMNI	50	Number of semaphore identifiers in the kernel. On most systems, the default is 50.
SEMMNS	150	Number of semaphores in the system. On most systems, the default is 60.
SEMMNU	30	Number of undo structures in the system. On most systems, the default is 30.

The SEMMNU parameter can limit the number of MessageQ processes, including all MessageQ system and application processes, that can run on a given system. Increasing the value of SEMMNU will allow you to attach more processes. The `dmqivpipc` does not check or verify the SEMMNU parameter.

Note: If you have other applications that use semaphores, you must account for them when calculating your semaphore resources. For example, if you have determined MessageQ software needs 30 semaphore identifiers and you use another product that requires 20, you should set SEMMNI to at least 50. Otherwise, you will not be able to use both products simultaneously.

The following parameters should not be set lower than the listed minimum values:

Parameter	Minimum Value	Description
SEMOPM	3	Maximum number of semaphore operations executed per semop system call.
SEMMSL	3	Maximum number of semaphores per semaphore ID.
SEMUME	5	Maximum number of undo entries per undo structure.

Note: Do not change the current values of these parameters if they are higher than the minimum values listed.

Calculating Message Resources

MessageQ uses message queues to communicate with the queuing engine. Calculate the number of message queues needed as follows:

Number of Message Queues	For Each...
2	Group
1	MessageQ server process (such as GCP, LD, JRN, SBS, user applications)
1	User application process

The following kernel resource values are adequate for most applications.

Parameter	Recommended Value	Description
MSGMAP	100	Size of control map that manages message segments
MSGMAX	32767	Maximum size of a message, in bytes
MSGMNB	65535	Maximum length of a message queue, in bytes
MSGMNI	50	Maximum number of message queues system-wide
MSGSSZ	8	Size of a message segment, in bytes
MSGTQL	40	Number of message headers in the system
MSGSEG	16384	Number of message segments in the system (128 kilobytes total)

Note that some parameters are not defined on some systems. For systems that have these parameters, use the following equation to correctly set the parameter values:

```
MSGMNB < = MSGSEG *  MSGSSZ < = 131072
```

Modifying the Kernel and Rebooting the System

If you modified the kernel configuration file and changed system parameters to support running MessageQ software, you may need to rebuild the kernel and reboot your system to ensure those changes take effect. To reconfigure the kernel, refer to the user documentation for your particular system.

After rebuilding the system kernel, replace the old system kernel with the new version. Be sure to save a backup copy of the old system kernel. Then reboot your system.

Verifying System Resources

After you change your system resources, you should test your proposed configuration against the new settings before configuring message queuing buses and groups.

To verify that you have set the system resources to support your proposed MessageQ configuration correctly, run the Installation Verification Process Interprocess Communications (`dmqivpipc`) program. The `dmqivpipc` program calculates the system resources required to support the proposed number of buses and local groups running on the local system, and compares the calculations against your current settings. The `dmqivpipc` program warns you if the current settings are low or too low, and provides recommended settings that ensure your MessageQ software will have sufficient resources to run properly.

Before running the `dmqivpipc` program, change directory to the bin subdirectory of your MessageQ installation directory, for example:

```
# cd /install_dir/bin/
```

To run the `dmqivpivc` program, enter the following command:

```
# dmqivpipc -v -b num_buses -g num_groups -p num_processes
```

Where:

`-v`	Lists all parameters.
`-b`	Specifies the number of buses in the proposed configuration. The default is 1.
`-g`	Specifies the number of local groups in the proposed configuration. The default is 1.
`-p`	Specifies the number of application processes in the proposed configuration. The default is 1.

The following example shows the system resource analysis for a proposed configuration of 3 buses and 7 groups.

```
# dmqivpipc -v -b 3 -g 7
```

```
Message Queue Parameters
      MSGMNI = 50
      MSGMAX = 8192
    * MSGMNB = 8192
          parameter is low
          preferred value is 32767
    * MSGSSZ * MSGSEG  = 8192
          parameter is low
          preferred value is 65535
      MSGTQL = 40

Semaphore Parameters
    * SEMMNI = 10
          parameter is too low
          minimum value is 24
          preferred value is 31
    * SEMMNS = 60
          parameter is too low
          minimum value is 72
          preferred value is 93
      SEMMSL = 25
      SEMOPM = 10
      SEMUME = 10

Shared Memory Parameters
      SHMMNI = 100
      SHMSEG = 6
      SHMMAX = 131072
```

Messages provided by the `dmqivpipc` program are interpreted as follows:

Message	Description
Parameter is low	The current setting supports the proposed number of buses and groups; however, it might not support MRS and cross-group messaging
Parameter is too low	The current setting will not support the proposed number of buses and groups
Minimum value is ...	The value suggested will support the proposed number of buses and groups, but will not support MRS or cross-group messaging.

Message	Description
Preferred value is ...	The value suggested will support the proposed number of buses and groups, as well as MRS and cross-group messaging.

3 Configuring MessageQ

This chapter contains the following topics that describe how to configure MessageQ message queuing buses and groups:

- Overview of MessageQ Configuration
- Configuring a Message Queuing Group
- Defining Parameters in the Group Initialization File
- Using the Test Utility to Test Your Group Configuration

If you have not already adjusted your system resources to support MessageQ software, use the information in Chapter 2 to configure your system resources, then rebuild and reboot the kernel.

Overview of MessageQ Configuration

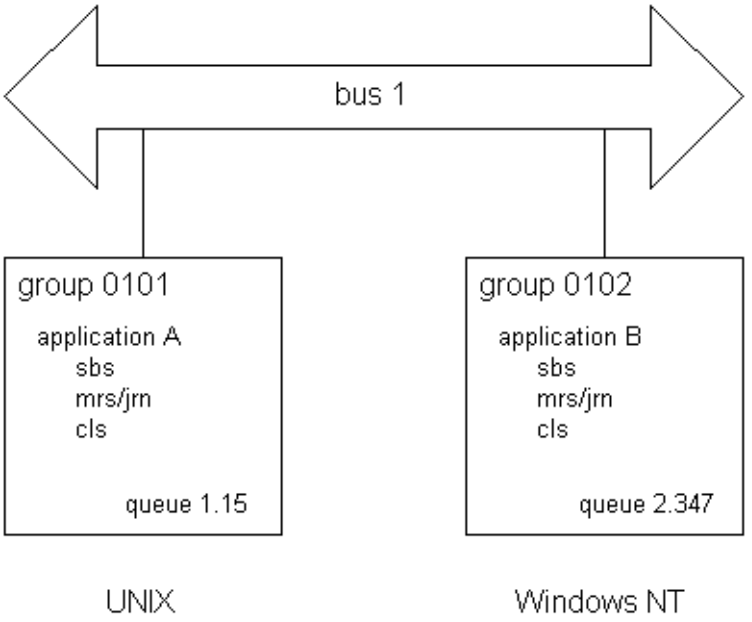
Before applications can exchange information, you must configure the MessageQ environment by defining the default values and parameter settings for the following components:

- Message queuing bus
- Message queuing groups
- Message queues
- Naming
- Cross-group connections

- Server Processes

Figure 3-1 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 utilizes the services of several MessageQ Servers such as the recoverable messaging (MRS) and journalling (JRN) server, message broadcasting (SBS), and Client Library support (CLS).

Figure 3-1 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.

Note: Aside from assigning a bus ID when you configure each message queuing group, there are no configuration steps you need to perform to configure the message queuing bus. MessageQ for UNIX automatically creates the bus when you enter the `dmqstartup` command to start a message queuing group (see the Starting a Message Queuing Group topic in Chapter 4 for more information).

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. The MessageQ startup procedure (`dmqstartup`) automatically creates the resources required for a group using the values and settings in the group initialization file. Message queuing groups are started by the startup procedure.

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 queue address. The group ID and a unique queue number form the queue address for each message queuing group.

MessageQ offers two types of queues: temporary and permanent:

- Temporary queues are created by MessageQ at runtime when they are requested using the `pams_attach_q` function. Applications use temporary queues when the need for the queue is short-lived.
- Permanent queues must be defined in the group initialization file. Permanent queues can become active when the group starts or when an application attaches. Applications use permanent queues when there is an ongoing need for the queue to service the application and when applications need to refer to the queue by name or number.

After you have selected the type of queue to use, you must set the following attributes of the queue:

- Primary or secondary
- Single reader or multireader
- Active on attach or permanently active

Each process that attaches to the MessageQ message queuing bus must have a primary queue assigned to it. This queue functions as the "main mailbox" for receiving messages from other processes using MessageQ. In addition, MessageQ applications can use secondary queues as a means of exchanging information among application components without interrupting the flow of messages taking place in the primary queue. In this way, secondary queues are used by application processes as alternate "mailboxes" for selected application messages.

Applications can be designed to read messages from one or more queues. Queues defined to be read by a single program are called single reader queues. When a process attaches to a single reader queue, it owns the queue and is the only process that can read from the queue. Queues that are designed to be read by multiple applications are called multireader queues (MRQs). MRQs are used to store messages that can be read

by many simultaneous readers, creating a central “mailbox” for several applications or application components to receive messages. Only permanent queues can be defined as MRQs. In addition, MRQs must have the attribute permanently active.

When defining permanent queues in the group initialization file, you have the choice of determining whether the queue becomes active when a process attaches to the queue or if the queue is active when the group starts up regardless of whether any process is attached. If the queue is configured as permanently active, processes can insert messages into the queue even if it is not currently attached by a client process.

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 have been configured and the group is started, the sender program must know the queue address to direct messages containing temperature readings. The receiver program 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.

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 address translation is provided in the Global Name Table (%GNT) section (see the Defining Name to Queue Translations in the

Group Name Section topic for more information) of the MessageQ configuration file for the group. Dynamic definition means that an application has created a name-to-queue address 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 global naming capability is called the Naming Agent. The Naming Agent stores the global name-to-queue address translations that were created using a static or dynamic approach. In addition, the Naming Agent provides the run-time lookup capability for global name-to-queue address translations requested using the `pams_locate_q` function.

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 the Configuring Global Naming topic 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 UNIX 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.

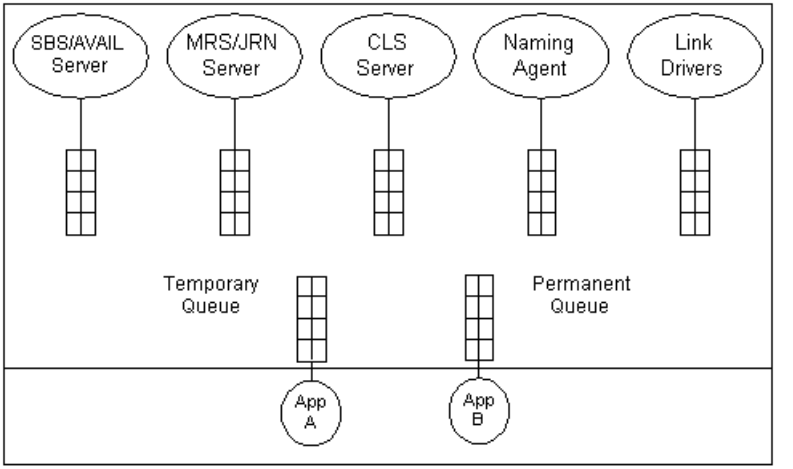
See the Enabling Network Connections in the Cross-Group Section topic and the Defining the Message Routing Database in the Route Section topic for instructions on how to configure direct and indirect cross-group connections.

You can also specify timeout intervals for testing cross-group connections. See the Defining Timeout Intervals for Link Drivers topic in Chapter 4 for more information.

Server Processes

MessageQ for UNIX 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 3-2 shows the server processes and other components of a MessageQ message queuing group.

Figure 3-2 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 3-1 describes the server processes that are available to MessageQ for UNIX.

Table 3-1 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.

Table 3-1 MessageQ Server Processes

Server Process	Description
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.
Naming Agent	Maintains the namespace for name-to-queue address translations and performs the runtime queue lookup when an application refers to a queue by name
Client Library Server (CLS)	Provides full message queuing support for MessageQ Clients.

For a complete description of MessageQ concepts and components, refer to the *Introduction to Message Queuing* documentation.

Configuring a Message Queuing Group

When configuring a message queuing group, you have two options:

- Let the `dmqsetup` utility configure the group for you by creating a standard group.

  For new MessageQ users or users who are configuring a new installation, the `dmqsetup` utility will help you quickly create a minimum message queuing group configuration to get MessageQ software up and running on your system in the shortest amount of time. For more information about the `dmqsetup` utility, see the Creating a Standard Group Configuration topic.

- Configure the group yourself.

  For users who want to create a custom configuration and understand the individual components of the group initialization file, see the Creating a Custom Group Configuration topic.

Creating a Standard Group Configuration

The dmqsetup utility lets you configure a standard group configuration and begin running MessageQ software on your system in the shortest time possible. The dmqsetup utility creates a standard, minimum configuration that uses default values and recommended parameter settings. This utility is ideal for users who have never performed a MessageQ group configuration, or for users who are considerably expanding an existing group configuration. The dmqsetup utility is installed with the product and can be found at the following location: */install_dir*/util.

The dmqsetup utility is a write-only utility. That is, after you exit from the utility and write the group initialization file, you cannot later use dmqsetup to make changes to that file. However, you can make changes to the group configuration file using a text editor.

The main menu of the dmqsetup utility contains the following options:

```
1.   Profile
2.   Links
3.   Routes
4.   Queues
5.   Names
6.   Create standard group initialization
7.   Save
8.   Help
9.   Exit
```

To create a standard group configuration, follow these steps:

1. Invoke the dmqsetup utility:

```
# dmqsetup
MessageQ for UNIX: Setup
1.   Profile
2.   Links
3.   Routes
4.   Queues
5.   Names
6.   Create standard group initialization
7.   Save
8.   Help
9.   Exit
```

2. Create a standard group initialization file by entering option 6.

```
*  Enter selection (1-9) [6]
```

3. Enter the name of the group, the group ID, and the node name.

```
* Enter group name [GROUP1] ?
* Enter group id (1-32000) [1] ?
* Enter the node name [chopin] ?
Standard group initialization has been created.
```

4. The standard group configuration has been created.
   Save by selecting 7.

```
MessageQ for UNIX: Setup
1.   Profile
2.   Links
3.   Routes
4.   Queues
5.   Names
6.   Create standard group initialization
7.   Save
8.   Help
9.   Exit
*  Enter selection (1-9) [6] ?  7
```

5. Enter the name you want to give the group initialization file you just created, and confirm that you want to save the file.

```
*  Enter group initialization file name [mygroup.init] ?
*  Save group initialization in file mygroup.init [Y] ?
Group initialization saved in file mygroup.init.
```

6. After you finish creating standard groups, exit from the `dmqsetup` utility.

```
MessageQ for UNIX: Setup
1.   Profile
2.   Links
3.   Routes
4.   Queues
5.   Names
6.   Create standard group initialization
7.   Save
8.   Help
```

```
9.   Exit
*  Enter selection (1-9) [7] ?  9
Bye.
```

A standard group has the following attributes:

- One template queue (Queue0) used as a template when temporary queues are created.
- 15 permanent queues.
- Five example queues used by the example programs provided in the MessageQ Example Programs installation subset.
- A DEAD_LETTER_QUEUE (Queue 96).
- A TCP/IP cross-group connection definition for the group. A standard group uses cross-group communications to exchange messages directly with other groups on nodes that share a network link. Note, however, that incoming TCP/IP connections will not be validated. See the XGROUP_VERIFY parameter in Table 3-3 for more information.

A standard group will not have Message Recovery Services (MRS) enabled, nor will it exchange messages with groups on nodes that do not share a network link. However, Selective Broadcast Services (SBS) will be enabled.

Creating a Custom Group Configuration

To enable you to configure groups to meet your specific messaging needs, MessageQ for UNIX systems provides a group initialization file template for each supported UNIX system. The system-specific templates can be found at the following location:

/install_dir/templates/group.init

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

Table 3-2 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.

Table 3-2 Sections of the Group Initialization File

Section Header	Name	Description
%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 Enabling Network Connections in the Cross-Group Section topic for more information.
%CLS	Client Library Server	Enables the support of multiple MessageQ Client connections to a single UNIX process using TCP/IP transports. See the Configuring 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.
%SBS	Selective Broadcast Services (SBS)	On OpenVMS systems, this section of the initialization file controls the operation of the local SBS server. On UNIX and Windows NT systems, 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.

Table 3-2 Sections of the Group Initialization File

Section Header	Name	Description
%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

You define parameters in the group initialization file by providing numeric values within a specified range or enabling settings to be YES or NO. The group initialization file is a typical ASCII text file that can be created and modified using any text editor. To create a custom group initialization file, make a copy of the template file and edit the copy. Do not edit the actual template file. You may want to add comments on your specific configuration. Use the exclamation, pound sign, semicolon, or asterisk characters (!, #, ;, or *) to as comment characters.

MessageQ also lets you modify a subset of parameters in the group initialization file at runtime. You can then use the Loader utility to dynamically reload the group initialization file without having to stop and restart MessageQ. See the Changing Group Characteristics at Runtime topic in Chapter 4 for instructions on how to run the Loader utility.

Several of the modifiable parameters can only be changed if certain conditions exist. These conditions are noted in the parameter description. 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 identifying the group initialization file version. (Note that this is the initialization file version, and is not equivalent to the product version number.) MessageQ for UNIX systems accepts group initialization files with version numbers of 4.0 only. MessageQ uses the version number to be able to correctly parse configuration information for the message queuing group at startup.

Listing 3-1 shows a sample %VERSION section.

Listing 3-1 Sample %VERSION Section

```
%VERSION 4.0
```

Setting Group Characteristics in the Profile Section

The %PROFILE section is used to configure 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.
- JRN Server—Enables journaling services for postconfirmation and dead letter journals.
- SBS Server—Enables message broadcasting and AVAIL/UNAVAIL notification.

Listing 3-2 shows a sample %PROFILE section:

Listing 3-2 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 3-3 describes the parameters of the %PROFILE section.

Table 3-3 %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 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).
FIRST_TEMP_QUEUE	101- 3999	200	Selects the starting size of the 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.

Table 3-3 %Profile Parameters

Parameter	Range	Default	Description
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_MESSAGE_SIZE	8192-4,194,304	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 tenths of seconds. **Note:** This parameter can be modified at runtime.
ENABLE_MRS	YES NO	NO	Enables Message Recovery Services.
ENABLE_JRN	YES NO	YES	Enables PCJ journaling when MRS is enabled. The default journaling action is not to write messages to the PCJ.
ENABLE_SBS	YES NO	YES	Enables Selective Broadcast Services and AVAIL/UNAVAIL services. The %SBS section in the group initialization file is not used by MessageQ for UNIX

Table 3-3 %Profile Parameters

Parameter	Range	Default	Description
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 UNIX systems, message recovery is provided by the `dmqjourn` journal process.

The server process that handles recoverable messaging is the MRS Server. In MessageQ, the MessageQ startup procedure starts up the MRS Server only if the ENABLE_MRS parameter is set to YES in the Profile section; then `dmqjourn` is started. When choosing a common directory where the journal files will be written, you should consider other files used by your programs (such as databases and logs) and the amount of available disk space. For a complete description of MRS, refer to the *Programmer's Guide*.

MessageQ for UNIX 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 resent after communication with the target group is restored.

Journal Type	Description
Destination Queue File (DQF)	Used to store recoverable messages targeted at the local group on the local system. Messages are automatically resent 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 not to 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 resent under user or application control.

Listing 3-3 shows a sample %MRS section and Table 3-4 describes the parameters of the %MRS section.

Listing 3-3 Sample %MRS Section

```
%MRS

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

Table 3-4 %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 a 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 3-4 %MRS Parameters

Parameter	Range	Default	Description
CACHE_PERCENTAGE	1 – 100	50	Specifies the receive quota to determine the number of messages that MRS maintains queued in the message pool to a given target queue. Multiply the CACHE_PERCENTAGE value by the MSG_QUOTA value (see the %QCT section) to determine the maximum number of recoverable messages that can reside in the memory based queue at one time. We recommend that you always set this parameter to less than 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.
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.

Table 3-4 %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 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 UNIX User's Guide.*

Listing 3-4 shows a sample %CLS section.

Listing 3-4 Sample %CLS Section

```
%CLS

!

!                              Max       Security
!   Endpoint     Transport     Clients   File

    12345        TCPIP         25        /my/dmqclsec.txt
     64354         TCPIP           50
```

```
%EOS
```

Table 3-5 describes the parameters of the %CLS section:

Table 3-5 %CLS Parameters

Parameter	Range	Default	Description
ENDPOINT	TCP/IP: 1024-65535	—	Identifies either the TCP/IP port number that the server uses to accept incoming connections. Using endpoint numbers under 5000 is recommended. 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.
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, MessageQ provides a default file name. For more information about the CLS security file, see the Restricting Remote Access to CLS topic in Chapter 4. **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 3-5 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 3-6 describes the parameters of the %XGROUP section.

Table 3-6 %XGROUP Parameters

Parameter	Range	Default	Description
GROUP NAME	--	--	Name by which the remote MessageQ group is known to the local group.
GROUP NUMBER	—	—	MessageQ Group Number of the remote group.
HOST	—	—	Name by which the remote host is known to the local host.
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 3-6 %XGROUP Parameters

Parameter	Range	Default	Description
THRESHOLD	1,048,576	1,048,576	This value is ignored and 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. **Note:** This parameter can be modified at runtime, but the link must be disabled.
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: 1024-3276	--	Protocol-stack-dependent address (port number) of the remote link listener process. Using endpoint numbers under 5000 is recommended. 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 up to 15 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 3-6 shows a sample %ROUTE section.

Listing 3-6 Sample %ROUTE Section

```
%ROUTE

!Target    Routethrough
!Group     Group

  7        1
  10       2

%EOS
```

Table 3-7 describes the parameters of the %ROUTE section.

Table 3-7 %ROUTE Parameters

Parameter	Range	Default	Description
TARGET_GROUP	--	--	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 3-7, the %XGROUP and %ROUTE sections are correctly defined. Group1 and Group2 are not defined as target groups in the %ROUTE section, and they are correctly defined as route-through groups.

Listing 3-7 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 3-8, 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 3-8 Incorrectly Defined %XGROUP and %ROUTE Sections

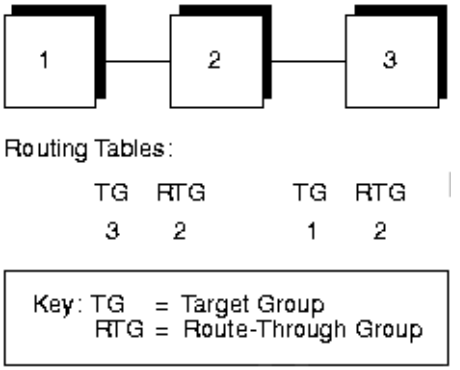
```
%XGROUP

!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, MessageQ automatically loads 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 3-3 provides an example of how route discovery allows a group to acquire routing information about other groups.

Figure 3-3 Route Discovery



In the example, Group 2 is connected to Groups 1 and 3, and Group 1's %ROUTE section was loaded at startup. When Group 1 sends a message to Group 3, MessageQ loads the return address of the message into Group 3's %ROUTE section. 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. When a message is sent between groups, 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.

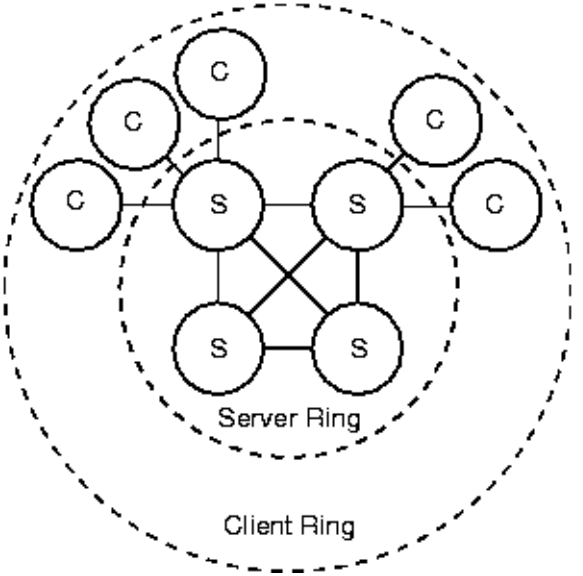
Full message routing rules followed by the MessageQ server are as follows:

- 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 3-4. Each server can route messages to other groups without requiring clients to be directly connected to the target group of the message.

Figure 3-4 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 UNIX systems, queues numbered 90 to 100 are reserved exclusively for MessageQ.

Listing 3-9 shows a sample %QCT section.

Listing 3-9 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 3-8 describes the parameters of the %QCT section.

Table 3-8 %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 3-8 %QCT Parameters

Parameter	Range	Default	Description
QUEUE NUMBER	--	--	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 template queue 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 template queue is specified, the default value of 128 is used. **Note:** This parameter can be modified at runtime.
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 field is not used. It exists only for compatibility with the MessageQ for OpenVMS software.

Table 3-8 %QCT Parameters

Parameter	Range	Default	Description
QUEUE TYPE	P S M	P	Designates the queue type as follows: P: Primary queue S: 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.
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 EI: 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.

Table 3-8 %QCT Parameters

Parameter	Range	Default	Description
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 `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.
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. **Note:** This parameter can be modified at runtime. See the Setting Security for Queues topic in Chapter 4 for more information.
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 of 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 3-10).

Table 3-8 %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 3-10).

The Flush Timer and Flush Depth parameters must be manually added to the file as the last two columns in the QCT group. Listing 3-10 shows the addition of these parameters:

Listing 3-10 Sample %GNT 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 3-11 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 3-11 Sample %GNT Section

```
%GNT

!
!Name           Group.Queue     Scope


INVENTORY_IN    9.10            G

ACCOUNTS_PAY    0.0             G

%EOS
```

Table 3-9 describes the parameters in the %GNT section.

Table 3-9 %GNT Parameters

Parameter	Range	Default	Description
NAME	--	--	Equivalence name. Queue reference used by applications to refer to the queue by name. Names can be up to 255 characters. **Note:** MessageQ for UNIX systems is case-sensitive, while MessageQ for OpenVMS is not. Use both upper and lower case for developers porting applications from OpenVMS systems.
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 connection to the group in which the Naming Agent runs.

Listing 3-12 shows a sample %NAM section.

Listing 3-12 Sample %NAM Section

```
%NAM

NA_GROUP    10

NA_GROUP    28

%EOS


%END
```

Valid ranges for the NA_GROUP parameter are 0 to 32,000. If the parameter is set to 0, a naming agent is started in the local group.

Configure the Namespace

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

To use global naming, you must create a namespace on the nodes on which the Naming Agents will run. 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 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 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.

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 create and maintain 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 syntax shows how to set the default namespace to be created and maintained in the 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.

Listing 3-13 shows static and dynamic global name definitions in the GNT section of the initialization file.

Listing 3-13 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 "`dev`" and the DEFAULT_NAMESPACE_PATH is set to "`/inventory`", the Naming Agent would search for the name "`widgets`" in:

```
/dev/inventory/widgets
```

partially qualified name — The application 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 "dev" and the DEFAULT_NAMESPACE_PATH were set to "`/inventory`", the Naming Agent would search for the name "`test/widgets`" in:

```
/dev/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 "`dev`" and the DEFAULT_NAMESPACE_PATH is set to "`/inventory`", the Naming Agent would search for the name "`/production/test/widgets`" in:

```
dev/production/test/widgets
```

The use of unqualified, partially qualified, and fully qualified names gives application developers significant flexibility in using global name references. provides several more examples of how global names are resolved. In Listing 3-14, the DMQNS_DEVICE environment variable is set to "dev" and the DMQNS_DEFAULT_PATH is set to "/dmq/dmqns".

Listing 3-14 Sample Global Names and Their Resolution

```
Name Used in API  DEFAULT_NAMESPACE_PATH  Name Searched


toto                bus1          /dev/dmq/dmqns/bus1/toto
mypath/toto         bus1          /dev/dmq/dmqns/bus1/mypath/toto
/anotherpath/toto   bus1          /dev/anotherpath/toto
toto                /bus1         /dev/bus1/toto
/mypath/toto        /bus          /dev/bus1/mypath/toto
/anotherpath/toto   /bus          /dev/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 application 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. The Test utility can be invoked by way of the Motif user interface or a character-cell user interface on UNIX systems.

To invoke the Test utility using the Motif user interface, set the environment variables for the bus and group ID and enter the following command:

```
dmqtestm
```

To invoke the Test utility using the character-cell interface, set the environment variables for the bus and group ID and enter the following command:

```
dmqtestc
```

Listing 3-15 shows the character-cell Test Utility main menu options.

Listing 3-15 Test Utility Main Menu

```
Main Menu

1 Parameters
2 Actions
3 Exit

Enter Menu Selection >> 1
```

Refer to Table 3-10 for the Test Utility Parameters and Actions menu Options.

Table 3-10 Test Utility Parameters and Actions Menu Options

Parameters Menu Option	Actions Menu Options
Attach Parameters	Attach Queue
Bind Parameters	Bind Queue
Detach Parameters	Detach Queue
Locate Parameters	Locate Queue

Table 3-10 Test Utility Parameters and Actions Menu Options

Parameters Menu Option	Actions Menu Options
Put Parameters	Put Message
Get Parameters	Get Message
Set Timer Parameters	Set Timer
Cancel Timer Parameters	Cancel Timer
View Current Parameters	View Current Parameters
Subscribe MOT Parameters	Subscribe MOT
Previous Menu	Previous Menu

The examples in the following section show how to use the Test Utility to perform various tasks such as:

- Setting the Attach parameters to specify a temporary primary queue
- Setting the Put parameters for the message
- Attaching to queues
- Putting the message to queue
- Detaching from the temporary queue
- Exiting from the Test Utility

Specifying a Temporary Queue

Listing 3-16 shows how to specify a temporary queue.

Listing 3-16 Specify a Temporary Queue

```
Main Menu
```

```
1 Parameters
2 Actions
3 Exit

Enter Menu Selection >> 1

Parameters Menu

 1 Attach Parameters
 2 Bind Parameters
 3 Detach Parameters
 4 Locate Parameters
 5 Put Parameters
 6 Get Parameters
 7 Set Timer Parameters
 8 Cancel Timer Parameters
 9 View Current Parameters
10 Subscribe MOT Parameters
11 Previous Menu

Enter Menu Selection >> 1

SELECT ATTACH TYPE
       1) Attach Primary
       2) Attach Secondary
Select attach type [1] ?

SELECT ATTACH_MODE
       1) Attach by name
       2) Attach by number
       3) Attach temporary
Select attach mode [3] ?
```

Setting the PAMS_PUT_MESSAGE Parameters

Listing 3-17 shows how to set the `pams_put_message` parameters.

Listing 3-17 Set the Put Parameters

```
Parameters Menu
```

```
 1 Attach Parameters
 2 Bind Parameters
 3 Detach Parameters
 4 Locate Parameters
 5 Put Parameters
 6 Get Parameters
 7 Set Timer Parameters
 8 Cancel Timer Parameters
 9 View Current Parameters
10 Subscribe MOT Parameters
11 Previous Menu

Enter Menu Selection >> 5

SELECT PRIORITY
       1) Standard Priority
       2) High Priority
Select priority [1] ?

SELECT DELIVERY MODE
       1) PDEL_MODE_AK_xxx
       2) PDEL_MODE_NN_xxx
       3) PDEL_MODE_WF_xxx
Select deliver mode [2] ? 3

SELECT DELIVERY MODE
       1) PDEL_MODE_xx_ACK
       2) PDEL_MODE_xx_CONF
       3) PDEL_MODE_xx_DEQ
       4) PDEL_MODE_xx_DQF
       5) PDEL_MODE_xx_MEM
       6) PDEL_MODE_xx_SAF
Select delivery mode [5] ? 5

SELECT UMA
       1) PDEL_UMA_DISC
       2) PDEL_UMA_RTS
       3) PDEL_UMA_SAF
       4) PDEL_UMA_DLQ
       5) PDEL_UMA_DLJ

Select UMA [1] ? 1
```

```
Enter target group [9] ? 9
Enter target queue [205] ?
Enter response queue [0] ?
Enter timeout in seconds [30] ?
Enter message class [1] ? 12
Enter message type [-100] ? 34
Enter message text ? This is a test message.
```

Attaching to a Queue

Listing 3-18 shows how to attach to a queue.

Listing 3-18 Attach to Queue 206 in Group 9

```
Main Menu

1 Parameters
2 Actions
3 Exit
Enter Menu Selection >> 2

Actions Menu

 1 Attach Queue
 2 Bind Queue
 3 Detach Queue
 4 Locate Queue
 5 Put Message
 6 Get Message
 7 Set Timer
 8 Cancel Timer
 9 View Current Parameters
10 Subscribe MOT
11 Previous Menu

Enter Menu Selection >> 1
attached to queue 9.206
```

If your MessageQ message queing group is properly configured, the Test utility returns a success message indicating that the attach operation was successful. However, if your MessageQ message queing group is not properly configured, an error message is displayed indicating the source of the problem as shown in Listing 3-19.

Listing 3-19 Common Errors

```
Actions Menu

 1 Attach Queue
 2 Bind Queue
 3 Detach Queue
 4 Locate Queue
 5 Put Message
 6 Get Message
 7 Set Timer
 8 Cancel Timer
 9 View Current Parameters
10 Subscribe MOT
11 Previous Menu

Enter Menu Selection >> 1
PAMS_RESRCFAIL, Failed to allocate resources.
```

Refer to the Troubleshooting chapter for additional information.

Sending a Message

Listing 3-20 shows how to send a message.

Listing 3-20 Put the Message to Queue 205 in Group 9

```
Actions Menu

 1 Attach Queue
 2 Bind Queue
 3 Detach Queue
```

```
 4 Locate Queue
 5 Put Message
 6 Get Message
 7 Set Timer
 8 Cancel Timer
 9 View Current Parameters
10 Subscribe MOT
11 Previous Menu

Enter Menu Selection >> 5
put message to queue 9.205
```

Detaching from a Temporary Queue

Listing 3-21 shows how to detach from the temporary queue.

Listing 3-21 Detach from Temporary Queue 205

```
Main Menu

1 Parameters
2 Actions
3 Exit

Enter Menu Selection >> 2

Actions Menu

 1 Attach Queue
 2 Bind Queue
 3 Detach Queue
 4 Locate Queue
 5 Put Message
 6 Get Message
 7 Set Timer
 8 Cancel Timer
 9 View Current Parameters
10 Subscribe MOT
11 Previous Menu
```

```
Enter Menu Selection >> 3
detached from queue 9.205
```

Setting the PAMS_GET_MSG Parameters

Listing 3-22 shows how to set the `pams_get_msg` parameters.

Listing 3-22 Set the PAMS_GET_MSG Parameters

```
Main Menu

1 Parameters
2 Actions
3 Exit

Enter Menu Selection >> 1

Parameters Menu

 1 Attach Parameters
 2 Bind Parameters
 3 Detach Parameters
 4 Locate Parameters
 5 Put Parameters
 6 Get Parameters
 7 Set Timer Parameters
 8 Cancel Timer Parameters
 9 View Current Parameters
10 Subscribe MOT Parameters
11 Previous Menu

Enter Menu Selection >>

SELECT INPUT QUEUE
       1) FIFO : default queue
       2) FIFO : alternate queue ?
Select input queue [1] ?


SELECT PRIORITY
```

```
       1) Standard Priority
       2) High Priority
Select priority [1] ?

SELECT WAIT OPTION
       1) PAMS_GET_MSG
       2) PAMS_GET_MSGW
Select wait option [1] ?
```

Receiving a Message

Listing 3-23 shows how to receive a message.

Listing 3-23 Receiving a Message

```
Main Menu

1 Parameters
2 Actions
3 Exit

Enter Menu Selection >> 2

Actions Menu

 1 Attach Queue
 2 Bind Queue
 3 Detach Queue
 4 Locate Queue
 5 Put Message
 6 Get Message
 7 Set Timer
 8 Cancel Timer
 9 View Current Parameters
10 Subscribe MOT
11 Previous Menu

Enter Menu Selection >> 6
received message from 9.205, class:12, type:34, : This is a test
message from dmq
```

Setting Detach Parameters

Listing 3-24 shows how to set the detach parameters.

Listing 3-24 Setting Detach Parameters

```
Main Menu

1 Parameters
2 Actions
3 Exit

Enter Menu Selection >> 1

Parameters Menu

 1 Attach Parameters
 2 Bind Parameters
 3 Detach Parameters
 4 Locate Parameters
 5 Put Parameters
 6 Get Parameters
 7 Set Timer Parameters
 8 Cancel Timer Parameters
 9 View Current Parameters
10 Subscribe MOT Parameters
11 Previous Menu

Enter Menu Selection >> 3

Detach all [0] ?

Flush on detach [1] ?

Enter queue number [205] ?
```

Exiting from the Test Utility

Listing 3-25 shows how to exit from the Test Utility.

Listing 3-25 Exit from the Test Utility

```
Main Menu

1 Parameters
2 Actions
3 Exit

Enter Menu Selection >> 3
```

4 Managing MessageQ

This chapter contains topics that describe how to manage MessageQ software on your system from the command line or from the MessageQ Monitor utility.

The topics include:

- 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
- Starting a Queue
- Stopping a Queue
- Starting and Stopping the CLS
- Managing Message Recovery Services (MRS)
- Changing group characteristics at runtime
- Enabling Quotas
- Setting Security for Queues
- Defining Timeout Intervals for Link Drivers

Using the Monitor Utility

The Monitor utility is a MessageQ application that allows you to monitor and control the performance of your MessageQ for UNIX 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

Note: For information about other MessageQ Monitor utility functions such as enabling, disabling, and resetting statistics collection, or viewing cumulative counts of the messages sent and received for all links and queues of a selected group, see Chapter 5, “Monitoring MessageQ”.

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 using the Monitor utility. The Monitor utility provides a character-cell and Motif user interface.

Invoking the Character-Cell Interface

If your monitor does not have graphics capabilities, you can invoke the character-cell user interface. To invoke the character-cell user interface without attaching to a bus or group, enter the following command:

```
# dmqmonc
```

To invoke the character-cell user interface and specify the bus and group, use the following command format:

```
# dmqmonc -b bus_id -g group_id
```

Where:

-b *bus_id*	bus number
-g *group_id*	group number

The bus and group you specified appear at the bottom of the main window. If you did not specify a bus and group, the bus and group listed will be specified by the environment variables DMQ_BUS_ID and DMQ_GROUP_ID.

To monitor or control your MessageQ groups or buses, enter the letter or number associated with the activity you want to perform. In most cases, a secondary screen appears allowing you to monitor or control the performance of your messaging system. To return to the main menu from any secondary screen, press Return.

Invoking the Motif User Interface

To use the Motif user interface, you must have a graphics monitor or workstation, and you must set the UIDPATH environment variable. Enter the following command to set this variable:

```
# setenv UIDPATH /usr/lib/X11/uid/%U:/install_dir/bin/%U
```

You must also set the NLSPATH environment variable, which allows MessageQ to properly translate error codes into text strings. To set the NLSPATH variable, enter the following command:

```
% setenv NLSPATH
/usr/lib/nls/msg/%N:/install_dir/lib/nls/msg/%N
```

To invoke the Motif user interface, enter the following command:

```
# dmqmonm
```

Use the **File** menu item to subsequently attach to a bus and group.

You can also invoke the Monitor utility and specify the bus and group on the command line, as shown in the following format:

```
# dmqmonm -b bus_id -g group_id
```

Where:

`-b` *`bus_id`*	bus number
`-g` *`group_id`*	group number

You can run the Motif user interface from a character-cell monitor by setting the display. For example, if you are running the Monitor utility on node *abc*, you can display the Motif user interface on node *xyz* by using the following commands:

csh Syntax	**sh Syntax**
`# setenv DISPLAY xyz:0.0`	`# DISPLAY=xyz0.0` `# export DISPLAY`
`# dmqmonm -b` *`bus_id`* `-g` *`group_id`*	`dmqmonm -b` *`bus_id`* `-g` *`group_id`*

Connecting to the MessageQ Environment

Before running a program that uses MessageQ, you must set up your environment to properly identify several paths, some of them platform specific.

You must define the environment variable BEADIR as the full path of the MessageQ installation directory. `$BEADIR/lib` must be included in the exported environment variable LIBPATH on AIX, SHLIB_PATH on HPUX, and LD_LIBRARY_PATH on all other Unix platforms.

You must also identify the message queuing bus and the message queuing group with which the program will be associated. A MessageQ program associates itself with a specified bus ID and group ID at runtime by obtaining the values of the DMQ_BUS_ID and DMQ_GROUP_ID environment variables.

To set environment variables that designate a bus and group ID using `csh` syntax, enter the following commands:

```
# setenv DMQ_BUS_ID bus_id
# setenv DMQ_GROUP_ID group_id
```

To set environment variables that designate a bus and group ID using sh syntax, enter the following commands:

```
# DMQ_BUS_ID = bus_id
# export DMQ_BUS_ID

# DMQ_GROUP_ID = group_id
# export DMQ_GROUP_ID
```

You must set these environment variables each time you install a new version of MessageQ for UNIX software.

Note that all MessageQ application programs, except those running on clients, should run under the same user identifier.

Connecting to a Group From the Monitor Utility

You can connect to a group from the MessageQ Monitor utility. The following procedures describe how to connect to a group using the character cell or Motif interface.

Connecting to a Group From the Character-Cell Interface

Use the following procedure to connect to a group:

1. Select **Connect to Group** from the **Main** menu.
2. Enter the Group number you want to connect to.

Connecting to a Group From the Motif Interface

Use the following procedure to connect to a group:

1. Select the **File** pull-down menu.
2. Select **Connect**.
3. Enter the Bus ID.
4. Enter the Group ID.
5. Select **OK**.

Starting a Message Queuing Group

To start MessageQ for UNIX systems, 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 uses the bus ID and the group ID to locate configuration information for a given application. See the topic in the previous section for more information.

You must set the NLSPATH environment variable, which allows MessageQ to properly translate error codes into text strings. MessageQ error or informational messages are displayed on your monitor screen. To set the NLSPATH variable, enter the following command:

```
% setenv NLSPATH
/usr/lib/nls/msg/%N:/install_dir/lib/nls/msg/%N
```

Starting a Group From the Monitor Utility

You can start a group from the MessageQ Monitor utility. The following procedures describe how to start a group using the character cell and Motif interfaces.

Starting a Group From the Character-Cell Interface

Use the following procedure to start a group:

1. Select **Start Group** from the **Main** menu.
2. Enter the bus number.
3. Enter the group number.
4. Enter the group name.
5. Enter the full path to the Group Initialization File.

   For example: `/user3/users/jones/bigfile.init`
6. Enter the full path for the log file name (optional).

   For example: `/user3/users/jones/group10.log`

   Note: If you do not use a log file, the output displays to the screen in an "unformatted" manner, making it difficult to read.

   MessageQ displays the message: `Group# (#) is started`.

Starting a Group From the Motif Interface

Use the following procedure to start a group:

Select the **Manage** pull-down menu.

Select **Start**.

1. Enter the bus number.
2. Enter the group number.
3. Enter the group name.
4. Enter the full path to the Group Initialization File.

For example: `/user3/users/jones/bigfile.init`

5. Enter the full path for the log file name (optional).

   For example: `/user3/users/jones/group10.log`

   Note: If you do not use a log file, the output displays to the screen in an "unformatted" manner, making it difficult to read.

6. Select **OK**.

Shutting Down a Running Group

MessageQ offers a shutdown procedure to shut down a running group immediately. 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 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 Using the Monitor Utility

You can shutdown a group from the MessageQ Monitor utility. The following procedures describe how to shutdown a group using the character cell or Motif interface.

Shutting Down a Group From the Character-Cell Interface

Use the following procedure to shutdown a group:

1. Select **Shutdown group** from the **Main** menu.

   MessageQ displays the question: `Stop gracefully, y or n?`

2. Enter **y** or **n**.

   If you enter **y**, 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 enter **n**, 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. You can use this option to stop a group immediately even after selecting that the group stops gracefully.

3. MessageQ displays the question:

   `Do you really want to shutdown group #?`

4. Enter **y** or **n**.

Shutting Down a Group From the Motif Interface

Use the following procedure to shutdown a group:

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

If you select **Soft Shutdown**, 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 **Fast Shutdown**, 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:

   ```
   Are you sure you really want to shutdown group #?
   ```

5. Select **OK**.

Starting a Cross-Group Connection

You can start a cross-group connection, also called a Link, using the MessageQ Monitor utility. The following procedures describe how to start a Cross-Group connection using the character cell or Motif interface.

Starting a Cross-Group Connection From the Character-Cell Interface

Use the following procedure to start a cross-group connection:

1. Select **Start Xgroup link** from the **Main** menu.

   MessageQ prompts you to enter the number of the group to which you want to form a cross-group connection.

2. Enter the cross-group number.

   A successful return code is indicated with the message:

   ```
   Link to Group # is started.
   ```

Starting a Cross-Group Connection From the Motif Interface

Use the following procedure 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. Select **OK**.

Stopping a Cross-Group Connection

You can stop a cross-group connection using the MessageQ Monitor utility. The following procedures describe how to stop a cross-group connection using the character cell or Motif interface.

Stopping a Cross Group Connection From the Character-Cell Interface

Use the following procedure to stop a cross-group connection:

1. Select **Stop Xgroup link** from the **Main** menu.

   MessageQ prompts you to enter the number of the group for which you want the cross-group connection stopped.

2. Enter the cross-group number of the link you want to stop.

   MessageQ immediately disables the link, which causes all cross-group communication with the selected group to cease. A successful return code is indicated with the message:

```
Link to Group # is stopped.
```

Note: To restart the link, you must issue the start command from the node that stopped the link. The link cannot be restarted from the remote node with which it was communicating.

Stopping a Cross Group Connection From the Motif Interface

Use the following procedure to stop a cross-group connection:

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

Starting a Queue

You can start a queue using the MessageQ Monitor utility. The following procedures describe how to start a queue using the character-cell or Motif interface.

Starting a Queue From the Character-Cell Interface

Use the following procedure to start a queue:

1. Select **Group Information** from the **Main** menu.
2. Select **Queue Start** from the **Group Information** menu.
3. Enter the queue number you want started.

Starting a Queue From the Motif Interface

Use the following procedure 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. Select **OK**.

Stopping a Queue

You can stop a queue using the MessageQ Monitor utility. The following procedures describe how to stop a queue using the character-cell or Motif interface.

Stopping a Queue From the Character-Cell Interface

Use the following procedure to stop a queue:

1. Select **Group Information** from the **Main** menu.
2. Select **Queue Stop** from the **Group Information** menu.
3. Enter the queue number you want stopped.

   MessageQ displays the question:

   ```
   Stop gracefully, y or n?
   ```

4. Enter **y** or **n**.

   If you enter **y**, MessageQ stops the specified queue in an orderly manner. When stopped gracefully, a queue will no longer accept new messages; however, it

does allow messages to be read until the queue is empty. When the queue is empty, the queue will be placed in a stopped state.

If you enter **n**, MessageQ stops the specified queue immediately. A stopped queue will not accept any new messages and will not allow messages to be read from the queue. You can use this option to stop a queue immediately even after selecting that the queue stops gracefully.

Stopping a Queue From the Motif Interface

Use the following procedure to stop a queue:

1. Select the **Manage** pull-down menu.
2. Select **Queue**.
3. Select **Fast Shutdown** or **Soft Shutdown**.
4. Select a queue number from the list.
5. Select **OK**.

Starting and Stopping the CLS

You can start and stop the CLS from the MessageQ Monitor utility. The following procedures describe how to start and stop the CLS using the character-cell or Motif interface

Starting the CLS From the Character-Cell Interface

Use the following procedure to start the CLS:

1. Select **Group Information** from the **Main** menu.
2. Select **CLS Start** from the **Group Information** menu.

3. Enter the endpoint number.
4. Enter **1** for TCP/IP.

Starting CLS From the Motif Interface

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.
5. Enter **1** for TCP/IP.

Stopping CLS From the Character-Cell Interface

Use the following procedure to stop the CLS:

1. Select **Group Information** from the **Main** menu.
2. Select **CLS Stop** from the **Group Information** menu.
3. Enter the endpoint number.
4. Enter **1** for TCP/IP.

Stopping CLS From the Motif Interface

Use the following procedure to stop the CLS:

1. Select the **Manage** pull-down menu.
2. Select **CLS**.
3. Select **Fast Shutdown**.

4. Enter the endpoint number.
5. Enter **1** for TCP/IP.

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.

The following is a sample security file template:

```
! %MODIFICATION HISTORY
!
!
! %ENDMODIFICATION HISTORY
!
!
!    Client Entry List
!
!
!   <Node>    : <Endpoint> : <Queue access>
!
sec_mgr       :            : ANY,CLS_SECURITY_FILE
client_01     : 5000       : ANY
client_02     : 5000       : 3,4
client_03     : 6010,6020  : QUEUE1
client_04     :            : ANY
client_05     :            : QUEUE1,2
client_06     :            : 2,3,4,5
```

The node, endpoint, and queue access field are defined as follows:

- Node -- The node name is usually the TCP/IP hostname for the client. The node name may also be an IP Address (for example, 1.23.45.156) or an alias name as defined in the TCP/IP hosts table on the CLS server system.

- Endpoint -- The endpoint access is defined by a list of 0 or more endpoints that this client can use. The endpoint list may contain up to 10 different endpoints. An empty endpoint list defaults to allowing access to any endpoint with no restrictions.
- Queue access -- The queue access identifies a list queues that this client can use. The queue list may contain up to 10 different queues. An empty queue list defaults to allowing access to any queue with no restrictions. Queues can be identified by queue names or queue numbers. Clients can also be restricted to allow access only to temporary queues by entering Temporary in this field.

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 UNIX systems, a template security file is available at the following location:

`/install_dir/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. Refer to Table 3-5 for a description of how to associate a security file with a 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 bus, but not between buses. 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

Caution: Do not use the Journal Replay utility on live recovery journals. Always make a copy of a live journal and run the utility against the copy. Using the Journal Replay utility on a live journal can cause irrecoverable file corruption.

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 4-1 describes the command options for `dmqjplay`.

Table 4-1 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 leave the contents intact (default). 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.
`-t`	*journal_type*	A constant that designates the file to be replayed, as follows: `d` Dead letter `p` Postconfirmation

Table 4-1 Journal Replay Utility Command Options

Switch	Argument	Description
`-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 -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 4-2 describes the command options for `dmqjdump`.

Table 4-2 Journal Dump Utility Command Options

Switch	Argument	Description
`-g`	*group*	Numeric Group ID; maximum of 5 digits
`-q`	*queue*	Queue number. When dumping a DQF file, specify the target queue number. When dumping an SAF file, specify the target group number. The `-q` is not used when dumping a PCJ or DLJ file.

Table 4-2 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` – Dead letter journal `dqf` – Destination queue file `pcj` – Postconfirmation journal `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 4-3 describes the parameters in the group initialization file that can be modified at runtime.

Table 4-3 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 4-3 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 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. For more information on setting these quotas in the %QCT section, see Defining Queues in the Queue Configuration Section in Chapter 3.

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

Enabling or Disabling Quotas From the Character-Cell Interface

Enable or disable quotas using the following procedure:

1. Select **Enable Quotas** or **Disable Quotas** from the **Main** menu.

   MessageQ displays the message: `Quotas enabled or Quotas disabled.`

Enabling or Disabling Quotas From the Motif Interface

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 Security for Queues

MessageQ lets you protect a queue from unauthorized reading by implementing an Access Control List (ACL) queue security. When set, the queue security prevents a process from attaching to a queue if the process does not have privilege to open a protected file on disk. MessageQ implements this security feature by creating a directory tree structure in the /var/tmp/dmq directory. The pathname to the security file for a selected queue is a combination of the bus, group, and queue number as follows:

```
/var/tmp/dmq/b_xxxx/g_yyyyy/acl/q_zzzz
```

Where:

xxxx	The number of the bus in decimal with leading zeros.
yyyyy	The number of the group in decimal with leading zeros.
zzzz	The number of queues in decimal with leading zeros.

For example, the security file for queue 50 in group 2 of bus 777 is:

```
/var/tmp/dmq/b_0777/g_00002/acl/q_0050
```

To configure a queue to use queue security, place the letter “y” in the last column of a queue entry in the %QCT section of the group initialization file, as follows:

```
QUEUE1  1  64000  100   None  .  P   0  .  N    Y
```

(See the Defining Queues in the Queue Configuration Section topic in Chapter 3 for more information on how to configure a queue.)

If the file system is shared for the ACL files for two different releases of MessageQ, there must be two versions of each security file.

If you specify queue security and the security file does not exist, you will not be able to attach the queue and you will receive the error status PAMS__NOACL. If the security queue file exists, you must have read/write access to the security file to attach the queue. If you cannot open the file for read/write access, the attach operation will fail and return the error status PAMS__NOPRIV.

The commands used to set file protections can vary depending on the file system. For most UNIX systems, the commands `chmod`, `chown`, and `chgrp` are used to set file protection. For more information about setting file protection, refer to the user documentation for the file system you are using.

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. 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 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 an idle 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.

5 Monitoring MessageQ

This chapter describes how to monitor the following types of MessageQ system activity on your system:

- Enable, disable, and reset statistics collection.
- View cumulative counts of the messages sent and received for all links and queues of a selected group.

Note: For additional information about using the MessageQ Monitor utility and some of its other managing functions, see Chapter 4, "Managing MessageQ".

Enabling, Disabling, and Resetting Statistics

To monitor your application's messaging capabilities, you need to collect statistics on the queue and link activity of the groups. Because statistics collection is not automatically started when you invoke the Monitor utility, you must enable statistics-gathering from the utility.

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

Enabling Statistics From the Character-Cell Interface

To enable statistics:

1. Select **Enable Statistics** from the **Main** menu.

   The utility displays the message: `Statistics enabled`

Disabling Statistics From the Character-Cell Interface

To disable statistics:

1. Select **Disable Statistics** from the **Main** menu.

   The utility displays the message: `Statistics disabled`

Resetting Statistics From the Character-Cell Interface

To reset statistics:

1. Select one of the **Reset** parameters (**All**, **Queue**, **Link**, **Single Queue**, or **Single Link**) from the **Main** menu.

   The utility displays the message: `Statistics reset`

Enabling, Disabling, or Resetting Statistics From the Motif Interface

To enable, disable, or reset statistics:

1. Select the **Manage** pull-down menu.
2. Select **Statistics**.
3. Select **Enable** or **Disable**, or **Reset**.

Using the Monitor to View Link and Queue Activity

You can use the Monitor utility to view link and queue statistics of MessageQ message queuing groups.

Please note that these statistics must be enabled before they can be viewed using the Monitor utility (see the Enabling, Disabling, and Resetting Statistics topic for information about enabling statistics).

Viewing Link and Queue Activity From the Character-Cell Interface

Use the following procedure to view all linking and queuing activity from the character-cell interface:

1. Select **Group Information** from the **Main** menu.
2. Select **Link** or **Queue** from the **Group Information** menu.
3. Select **Traffic Counts**, **Traffic Rates**, **Traffic Detail**, or **Link Detail** to view the activity.

Viewing Link and Queue Activity From the Motif Interface

Use the following procedure to view all linking and queuing activity from the Motif interface:

1. Select the **View** pull-down menu.
2. Select **Link Information** or **Queue Information**.

3. Select **Traffic Counts**, **Traffic Rates**, **Traffic Detail**, or **Detail** to view the activity.

Link and Queue Activity

The descriptions and tables in this section explain the link and queue activity that can be viewed using the MessageQ Monitor utility.

You can view statistics with the following dialogs:

- Traffic Counts
- Traffic Rates
- Traffic Detail
- Detail

Viewing Link Traffic Counts

Use the Monitor utility 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 will appear.

Table 5-1 describes the fields of the Link Traffic Counts dialog.

Table 5-1 Link Traffic Counts Dialog

Statistic	Definition
Group Number	Number of the group
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 form the last time statistics were reset

Table 5-1 Link Traffic Counts Dialog

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

This dialog can also be useful for diagnosing message flow between cross-group connections. 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 are referred to as **throughput rates**. Table 5-2 describes the fields of the Link Traffic Rates dialog.

Table 5-2 Link Traffic Rates Dialog

Statistic	Definition
Group Number	Number of the group
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

Table 5-2 Link Traffic Rates Dialog

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 is 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. See the Controlling Message Flow topic in the Programmer's Guide for more information.

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 5-3 describes the fields of the Link Traffic Detail.

Table 5-3 Link Traffic Detail Dialog

Statistic	Definition
Group Name	Name of the group
Group Number	Number of the group
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

Table 5-3 Link Traffic Detail Dialog

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 queues. If you want to narrow down the problem further and suspect that the problem is due to one queue only, use the Viewing 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 5-4 describes the fields of the Link Detail dialog.

Table 5-4 Link Detail Dialog

Statistic	Definition
Group Name	Name of the group
Group Number	Number of the group
Time Created	First time a link was made to a remote 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
Time Statistics Reset	Time statistics were last reset
PID	Process ID for the sender and receiver programs
State	State of the sender and receiver programs; either connected or unconnected

Table 5-4 Link Detail Dialog

Window Timer	Number of seconds the sender program must wait before using a new window. This value is set by the DELAY parameter in the %XGROUP section of the group initialization file (see the Enabling Network Connections in the Cross-Group Section topic in Chapter 3 for more information).
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 parameter in the %XGROUP section of the group initialization file. For example, if a sender program sends 10 messages per second, but the receiver program 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 programs.
Number of Messages	Highest number of messages in the queue at any given time
Number of Bytes	Highest number of bytes in the queue at any given time
Number of Connections	Cumulative number of successful connections made by the group since it started up
Number of Connection Attempts	Number of connection attempts that failed since the last successful connection. Whenever a connection attempt is successful, this number is reset to 0.

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 dialog to be useful. Table 5-5 describes the fields of the Queue Traffic Counts dialog.

Table 5-5 Queue Traffic Counts Dialog

Statistic	Definition
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

Table 5-5 Queue Traffic Counts Dialog

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	Size, in bytes, of the current messages waiting to be read
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 %QCT section is by default a primary queue. Queues numbered 90 to 100 and 150 to 199 are reserved exclusively for MessageQ.

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 5-6 describes the fields of the Queue Traffic Rates dialog.

Table 5-6 Queue Traffic Rates Dialog

Statistic	Definition
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

Table 5-6 Queue Traffic Rates Dialog

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 %QCT section is by default a primary queue. Queues numbered 90 to 100 and 150 to 199 are reserved exclusively for MessageQ.

Note: 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 5-7 describes the fields of the Queue Traffic Detail dialog.

Table 5-7 Queue Traffic Detail Dialog

Statistic	Definition
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

Table 5-7 Queue Traffic Detail Dialog

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 %QCT section is by default a primary queue. Queues numbered 90 to 100 and 150 to 199 are reserved exclusively for MessageQ.

Viewing Queue Detail

Use the Queue Information and Detail dialog to view the attributes of a selected queue. If quotas are 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 5-8 describes the fields of the Queue Detail dialog.

Table 5-8 Queue Detail Dialog

Statistic	Definition
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 %QCT section is by default a primary queue. Queues numbered 90 to 100 and 150 to 199 are reserved exclusively for MessageQ.
Owner PID	Process ID of the attached queue
Time Attached	Last time the queue was attached to the bus
Time Last Detached	Last time the queue was detached from the bus
Time Statistics Reset	Last time statistics for this queue were reset
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

Table 5-8 Queue Detail Dialog

Message Quotas	The maximum limit of uncollected messages that can reside in this queue, and the current number. This value is set by the MSG_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 BYTE_QUOTA parameter of the %QCT section in the group initialization file.

6 Troubleshooting

This chapter contains the following topics that describe how to troubleshoot MessageQ software on your system:

- Error Logging
- Solving Startup Problems
- Solving Queue Attachment Problems
- Logging and Tracing CLS Events

Error Logging and Recovery

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.

The MessageQ error log file is created when the group is started using the dmqstartup procedure. An error log is created for each message queuing group. If you start a group without creating a log file and later determine that you want to use a log file, you must start the message queue group and restart it using the appropriate command line switch to create the error log.

The following example shows the kind of information contained in the error log included the following information for each log event:

- The name of the process that logged the error, its process identification number (PID) and the date and time on which the event was logged

- An abbreviation of the MessageQ processing level at which the error occurred
- A description of the successful event or error condition

Listing 6-1 Error Logging and Recovery

```
************ dmqgcp (4150) 10-DEC-1996 15:25:23 ************
gcp, group control process for group 19 is running

************ dmqqe (4536) 10-DEC-1996 15:25:24 ************
qe, queuing engine is running

************ dmqloader (3366) 10-DEC-1996 15:25:24 ************
ldr, MessageQ System Loader starting

ldr, Parsing PROFILE Section
ldr, Parsing MRS Section
ldr, Parsing GROUP Section
ldr, Parsing ROUTE Section
ldr, Parsing QCT Section
ldr, Parsing GNT Section
ldr, Bad parameter sent to the GCP at line 318
ldr, Bad parameter sent to the GCP at line 330
ldr, Bad parameter sent to the GCP at line 331
ldr, Parsing CLS Section
ldr, Parsing NAM Section
ldr, Loader exiting normally
************ dmqjourn (2590) 10-DEC-1996 15:25:27 ************
jrn, journal process for group 19 is running

************ dmqld (2579.0) 10-DEC-1996 15:25:27 ************
ld, link listener for group 19 is running

************ dmqld (2591.0) 10-DEC-1996 15:25:27 ************
ld, link sender for group 19 to group 18 is running

************ dmqld (2579.0) 10-DEC-1996 15:25:32 ************
ld, link receiver for group 19 from group 18 is running

************ dmqgcp (4150) 10-DEC-1996 15:26:21 ************
```

```
ipi, dequeue message failed

************ dmqgcp (4150) 10-DEC-1996 15:26:21 ************
gcp, group control process for group 19 has exited

************ dmqqe (4536) 10-DEC-1996 15:26:21 ************
qe, queuing engine has exited
```

Solving Startup Problems

If MessageQ does not start, possible problems and solutions are as follows:

- Your system parameters are not properly configured.

  If you have not already done so, use the Installation Verification Process Interprocess Communications (`dmqivpipc`) program to verify that your system parameters are adequate to run your proposed MessageQ configuration. For more information, see "Verifying System Resources" on page 2-10.

- Your group initialization file is not properly configured.

  Check your existing group initialization file for errors, or use the `dmqsetup` utility described in the Creating a Standard Group Configuration topic in Chapter 3 to create a minimal configuration file that will get MessageQ up and running in a short amount of time.

Solving Queue Attachment Problems

If MessageQ is running but you cannot attach to the queue, check that the DMQ_BUS_ID and DMQ_GROUP_ID environment variables are set. If the environment variables are not set, follow the instructions in the Connecting to the MessageQ Environment topic in Chapter 4.

If the environment variables are properly set and you still cannot attach to the queue, check to see if the group is running by entering the following command:

```
# ps | grep dmq
```

In this example, Bus 3000, Group 6 was running when the command was entered:

```
BUS 3000 Group 6 is running:
ps | grep dmq
29715 ttyp1    S  +  0:00.02 grep dmq
23191 ttyp3    I     0:00.07 dmqld -t listener -b 3000 -g 6 -l /u/jones/b
24306 ttyp3    S     0:03.69 dmqld -t sender -b 3000 -g 6 -r 5 -l /u/jones
25163 ttyp3    S     0:47.59 dmqqe -b 3000 -g 6 -l /u/jones/bus3000-heliu
25182 ttyp3    I     0:00.02 dmqbcp -n wsh -b 3000 -g 1 -f /u/jones/bus30
25614 ttyp3    S     0:00.37 dmqld -t listener -b 3000 -g 6 -l /u/jones/b
25725 ttyp3    S     0:21.04 dmqjourn -b 3000 -g 6 -l /u/jones/bus3000-he
32422 ttyp3    S     0:07.56 dmqgcp -b 3000 -n U3AL -g 6 -f /u/jones/bus3
```

If the group is not running, start it by entering the `dmqstartup` command as described in the Starting a Message Queuing Group topic in Chapter 4.

If the group is running and you still cannot attach, perform the following checks:

- Ensure that the queues you are attempting to attach to are listed in the group initialization file.
- Ensure the case of your queue name is correct.

  Note: Remember that MessageQ for UNIX is case-sensitive, unlike MessageQ for other platforms.

If you have performed all of the checks and still cannot attach to the group, follow these steps:

1. Check the MessageQ log files. You will need privileges to access the files; see your system manager to get access to the files.
2. Enable `PAMS_TRACE` and `DMQIPI_TRACE`, and attempt to attach to the group again.
3. Call the Customer Support Center (see the Customer Support topic in the Preface for more information).

Logging and Tracing CLS Events

CLS event logging is integrated with the MessageQ for UNIX logging facility. CLS server initialization and client connect/disconnect events are logged to the common log file.

The CLS also supports trace output to provide detailed information about the MessageQ API functions executed by the CLS on behalf of client applications. To enable CLS tracing at MessageQ startup, set the CLS environment variable as follows:

csh Syntax	sh Syntax
`#setenv DMQCLS_TRACE 1`	`#DMQCLS_TRACE = 1` `# export DMQCLS_TRACE`

Tracing output is directed to the file `dmqtrace.log` in the current working directory

A Uninstalling MessageQ V3.x and V4.x

Perform the following steps to uninstall MessageQ V3.x or V4.x installations from your system:

1. Determine if version 3.x or 4.x is installed. If you are unsure of whether or not you have a previous installation of BEA MessageQ installed, enter the following command:

```
# ls -l /usr/bin/dmqqe
```

   If this file exists, then an older version of MessageQ is installed and you should proceed with these instructions.

2. Log in as root. If you are not already logged in as root (superuser), do so now. You must be logged in as root in order to deinstall previous versions of MessageQ.

3. Determine what MessageQ version and components are installed by issuing the following command:

```
# setld -i | grep -i MessageQ
```

   The output will look something like this (the output and installed component names differ from machine to machine):

```
# setld -i | grep -i MessageQ
DMACL32B                        MessageQ Client Library
DMACL40A            installed   MessageQ Client Library
DMACLS32B                       MessageQ Client Library Server
DMADEV32B                       MessageQ Development Environment
DMADEV40A           installed   MessageQ Development Environment
DMADOC40A           installed   MessageQ Online Documentation
```

```
DMAEXA32B                       MessageQ Example Programs
DMAEXA40A           installed   MessageQ Example Programs
DMAINC40A           installed   MessageQ Include Files
DMAMAN32B                       MessageQ Manual Reference Pages
DMARLS32B                       MessageQ Release Notes
DMARLS40A           installed   MessageQ Release Notes
DMARTO32B                       MessageQ Run Time Environment
DMARTO40A           installed   MessageQ Run Time Environment
#
```

If the `setld` command can not be found or `setld` did not find any installed components, issue the command from the `/etc/ directory` as follows:

```
#/etc/setld -i | grep -i MessageQ
```

If `/etc/setld` does not exist or `/etc/setld` did not find any installed components, call your BEA Systems support representative for additional help in removing the previous version of MessageQ. Note that if you do not remove versions of MessageQ prior to version V5.0, you may experience serious operational problems with your new installation.

4. Remove the installed components. Issue either `setld -d` (or `/etc/setld -d` if `/etc/setld` was used in Step3) to remove the installed components. Notice in the example above that some components are installed and others are not installed; only remove the installed components. Using the example in Step 3, the command would look like this (split here into two calls to setld):

```
#setld -d DMACL40A DMADEV40A DMADOC40A DMAEXA40A
#setld -d DMAINC40A DMARLS40A DMARTO40A
```

The output from this will look something like this:

```
# setld -d DMACL40A DMADEV40A DMADOC40A DMAEXA40A

Deleting "MessageQ Example Programs" (DMAEXA40A).

Deleting "MessageQ Online Documentation" (DMADOC40A).

Deleting "MessageQ Client Library" (DMACL40A).

Deleting "MessageQ Development Environment" (DMADEV40A).
# setld -d DMAINC40A DMARLS40A DMARTO40A

Deleting "MessageQ Release Notes" (DMARLS40A).

Deleting "MessageQ Run Time Environment" (DMARTO40A).
```

```
Deleting "MessageQ Include Files" (DMAINC40A).
#
```

5. Confirm that the older version has been removed and that `setld` removed everything by re-issuing the `setld -i` (or `/etc/setld -i`) command:

```
# setld -i | grep -i MessageQ
DMACL32B                        MessageQ Client Library
DMACL40A                        MessageQ Client Library
DMACLS32B                       MessageQ Client Library Server
DMADEV32B                       MessageQ Development Environment
DMADEV40A                       MessageQ Development Environment
DMADOC40A                       MessageQ Online Documentation
DMAEXA32B                       MessageQ Example Programs
DMAEXA40A                       MessageQ Example Programs
DMAINC40A                       MessageQ Include Files
DMAMAN32B                       MessageQ Manual Reference Pages
DMARLS32B                       MessageQ Release Notes
DMARLS40A                       MessageQ Release Notes
DMARTO32B                       MessageQ Run Time Environment
DMARTO40A                       MessageQ Run Time Environment
#
```

If nothing is marked as being "installed", you are done with `setld`. To be sure, look for installed files that may not have been removed by issuing the following command:

```
# ls -l /usr/bin/dmq* /usr/lib/dmq*
```

If all files have been removed, no files will be found by the command. If you can not remove all of the MessageQ components, contact you BEA Systems support representative for additional help.