Script (MQSC) Command Reference
First edition (June 2002)

This edition applies to the following WebSphere MQ V5.3 products:
- WebSphere MQ for AIX
- WebSphere MQ for HP-UX
- WebSphere MQ for iSeries
- WebSphere MQ for Linux for Intel
- WebSphere MQ for Linux for zSeries
- WebSphere MQ for Solaris
- WebSphere MQ for Windows
- WebSphere MQ for z/OS

and to all subsequent releases and modifications until otherwise indicated in new editions.

Unless otherwise stated, the text also applies to these products:
- MQSeries for AT&T GIS (NCR) UNIX, V2.2.1
- MQSeries for Compaq NonStop Kernel, V5.1
- MQSeries for Compaq OpenVMS Alpha, V5.1
- MQSeries for Compaq OpenVMS VAX, V2.2.1
- MQSeries for Compaq Tru64 UNIX, V5.1
- MQSeries for OS/2 Warp, V5.1
- MQSeries for SINIX and DC/OSx, V2.2.1
- MQSeries for Sun Solaris, Intel Platform Edition, V5.1

and to all subsequent releases and modifications until otherwise indicated in new editions.

This book was previously entitled MQSeries®MQSC Command reference, SC33-1369-13.

© Copyright International Business Machines Corporation 1993, 2002. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>STOP CMDSERV</td>
<td>330</td>
</tr>
<tr>
<td>Usage notes</td>
<td>330</td>
</tr>
<tr>
<td>STOP LISTENER</td>
<td>331</td>
</tr>
<tr>
<td>Parameter descriptions</td>
<td>331</td>
</tr>
<tr>
<td>STOP QMGR</td>
<td>333</td>
</tr>
<tr>
<td>Parameter descriptions</td>
<td>333</td>
</tr>
<tr>
<td>STOP TRACE</td>
<td>335</td>
</tr>
<tr>
<td>Parameter descriptions</td>
<td>335</td>
</tr>
<tr>
<td>SUSPEND QMGR</td>
<td>338</td>
</tr>
<tr>
<td>Parameter descriptions</td>
<td>338</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix. Notices</td>
<td>341</td>
</tr>
<tr>
<td>Trademarks</td>
<td>343</td>
</tr>
<tr>
<td>Index</td>
<td>345</td>
</tr>
<tr>
<td>Sending your comments to IBM</td>
<td>353</td>
</tr>
</tbody>
</table>
## Tables

1. How to read syntax diagrams ........................................... 7
2. Examples of how the LOCLADDR parameter can be used ............ 104
3. CipherSpecs that can be used with WebSphere MQ SSL support ....... 109
4. CHLDISP and CMDSCOPE for DISPLAY CSTATUS CURRENT ......... 200
5. CHLDISP and CMDSCOPE for DISPLAY CSTATUS SHORT ............ 200
6. CHLDISP and CMDSCOPE for DISPLAY CSTATUS SAVED ............ 200
7. Parameters that can be returned by the DISPLAY QUEUE command 243
8. CHLDISP and CMDSCOPE for PING CHANNEL .......................... 268
9. CHLDISP and CMDSCOPE for RESET CHANNEL ....................... 282
10. CHLDISP and CMDSCOPE for RESOLVE CHANNEL .................... 291
11. CHLDISP and CMDSCOPE for START CHANNEL ....................... 309
12. Destinations allowed for each trace type ............................ 321
13. Constraints allowed for each trace type .............................. 321
14. IFCID descriptions for IFCID trace events and classes ............... 321
15. Resource Manager identifiers that are allowed ....................... 322
16. CHLDISP and CMDSCOPE for STOP CHANNEL .......................... 326
About this book

This book describes the MQSC commands, which system operators and administrators can use to manage queue managers on the following WebSphere MQ platforms:

- Compaq NSK
- Compaq OpenVMS
- OS/2 Warp
- OS/400
- UNIX operating systems
- Windows
- z/OS

The commands are described in alphabetic order in Chapter 2, “The MQSC commands” on page 9. At the start of each command description, the platforms on which you can use the command are shown.

The term “UNIX systems” is used to denote the following UNIX operating systems:

- AIX
- AT&T GIS (NCR) UNIX
- Compaq Tru64 UNIX
- HP-UX
- Linux (for Intel and zSeries)
- SINIX and DC/OSx
- Solaris (SPARC and Intel Platform Editions)

The term “Windows” is used throughout this book to denote the following Windows operating systems, unless stated otherwise:

- Windows NT
- Windows 2000

“z/OS” means any release of z/OS or OS/390 that supports the current version of WebSphere MQ.

Who this book is for

This book is intended for system programmers, system administrators, and system operators.

What you need to know to understand this book

To understand this book, you should be familiar with the system facilities for the platform on which you are using the WebSphere MQ product.

If you are unfamiliar with the concepts of messaging and queuing, you should read An Introduction to Messaging and Queuing.

How to use this book

For platforms other than z/OS, read those sections of the WebSphere MQ Administration Guide, System Management Guide, or System Administration book for your platform that relate to the task you want to perform.
For z/OS, read the sections of the WebSphere MQ for z/OS System Administration Guide, WebSphere MQ for z/OS System Setup Guide, or both that relate to the task you want to perform.

When you have decided which commands you need to use, use this book to learn their syntax.

The syntax of the MQSC commands is represented in syntax diagrams. To learn how to read these diagrams, see “How to read syntax diagrams” on page 7. The parameters for each command are listed in the following order in the syntax diagrams:

• Parameters that are required are listed first, in alphabetic order.
• Parameters that are optional follow, again in alphabetic order.
Summary of changes

This section describes changes in this edition of *WebSphere MQ Script (MQSC) Command Reference*. Changes since the previous edition of the book are marked by vertical lines to the left of the changes.

Changes for this edition (SC33-1369-14)

- WebSphere MQ has been added to the following new platforms:
  - Linux for Intel
  - Linux for zSeries
- Secure Sockets Layer (SSL) support has been added to WebSphere MQ, and the following new commands have been added to support this feature:
  - ALTER AUTHINFO
  - DEFINE AUTHINFO
  - DELETE AUTHINFO
  - DISPLAY AUTHINFO
- WebSphere MQ for z/OS now supports persistent shared queue messages that can survive the failure of a Coupling Facility (CF) list structure. To enable this function, changes have been made to the DISPLAY USAGE command, and the following new commands have been created:
  - ALTER CFSTRUCT
  - BACKUP CFSTRUCT
  - DEFINE CFSTRUCT
  - DELETE CFSTRUCT
  - DISPLAY CFSTRUCT
  - DISPLAY CFSTATUS
  - RECOVER CFSTRUCT
- WebSphere MQ for z/OS now allows you to change the queue manager systems parameters dynamically. The SET LOG and DISPLAY LOG commands have changed, and the following new commands have been added to support this feature:
  - DISPLAY ARCHIVE
  - DISPLAY SYSTEM
  - SET ARCHIVE
  - SET SYSTEM
- A new command, REFRESH QMGR, has been introduced to enable you to perform special operations on queue managers.
- A new queue manager id parameter, QMID, has been added to the RESET CLUSTER command.
- A number of new parameters have been added to the channel commands.
- New parameters, CONFIGEV, and EXPRYINT have been added to the following commands:
  - ALTER QMGR
  - DISPLAY QMGR
Changes

- Multiple channel exits are now supported on z/OS, which involves minor changes to the ALTER CHANNEL and DEFINE CHANNEL commands.
- WebSphere MQ for z/OS now supports message grouping for shared and non-shared queues. This function is enabled by the introduction of a new INDXTYPE of GROUPID for local queues. It uses message priority internally to provide an efficient method of finding candidate groups and checking them for completeness. INDXTYPE(GROUPID) can be used on the following commands:
  - ALTER QLOCAL
  - ALTER QMODEL
  - DEFINE QLOCAL
  - DEFINE QMODEL
- DISPLAY USAGE has a new TYPE keyword, in particular TYPE(DATASET), which provides recovery and backup information about WebSphere MQ logs.
- Queue manager restart time on z/OS is no longer impacted by the need to rebuild persistent message indexes. This means that you can change the INDXTYPE of a queue immediately, even if it contains messages or is open, without waiting for a restart.
- The platforms in the tables at the start of each command have been rearranged into alphabetical order.
Chapter 1. Using MQSC commands

MQSC commands provide a uniform method of issuing human-readable commands on WebSphere MQ platforms. For information about programmable command format (PCF) commands (not available on z/OS), see the WebSphere MQ Programmable Command Formats and Administration Interface manual.

This chapter describes:

- “Rules for using MQSC commands”
- “Rules for naming WebSphere MQ objects” on page 4
- “How to read syntax diagrams” on page 7

The general format of the commands is shown in Chapter 2, “The MQSC commands” on page 9.

Rules for using MQSC commands

You should observe the following rules when using MQSC commands:

- Each command starts with a primary parameter (a verb), and this is followed by a secondary parameter (a noun). This is then followed by the name or generic name of the object (in parentheses) if there is one, which there is on most commands. Following that, parameters can usually occur in any order; if a parameter has a corresponding value, the value must occur directly after the parameter to which it relates.

  Note: On z/OS, the secondary parameter does not have to be second.

- Keywords, parentheses, and values can be separated by any number of blanks and commas. A comma shown in the syntax diagrams can always be replaced by one or more blanks. There must be at least one blank immediately preceding each parameter (after the primary parameter) except on z/OS.

- Any number of blanks can occur at the beginning or end of the command, and between parameters, punctuation, and values. For example, the following command is valid:

  \[ ALTER QLOCAL ('Account' ) TRIGDPTH ( 1) \]

  Blanks within a pair of quotation marks are significant.

- Additional commas can appear anywhere where blanks are allowed and are treated as if they were blanks (unless, of course, they are inside quoted strings).

- Repeated parameters are not allowed. Repeating a parameter with its ‘NO’ version, as in REPLACE NOREPLACE, is also not allowed.

- Strings that contain blanks, lowercase characters or special characters other than:
  - Period (.)
  - Forward slash (/)
  - Underscore (_)
  - Percent sign (%)

  must be enclosed in single quotation marks, unless they are:
  - Issued from the WebSphere MQ for z/OS operations and control panels
  - Generic names ending with an asterisk (on OS/400 these must be enclosed in single quotation marks)
Rules for using commands

- A single asterisk (for example, TRACE(*)) (on OS/400 these must be enclosed in single quotation marks)
- A range specification containing a colon (for example, CLASS(01:03))

If the string itself contains a quotation mark, the quotation mark is represented by two single quotation marks. Lowercase characters not contained within quotation marks are folded to uppercase.

- A string containing no characters (that is, two single quotation marks with no space in between) is interpreted as a quoted blank space, in other words, (' ') is interpreted in the same way as (' ').
- A left parenthesis followed by a right parenthesis, with no significant information in between, for example

  NAME ( )

is not valid except where specifically noted.
- Keywords are not case sensitive – AltER, alter, and ALTER are all acceptable. Names that are not contained within quotation marks are folded to uppercase.
- Synonyms are defined for some parameters. For example, DEF is always a synonym for DEFINE, so DEF QLOCAL is valid. Synonyms are not, however, just minimum strings; DEFI is not a valid synonym for DEFINE.

Note: There is no synonym for the DELETE parameter. This is to avoid accidental deletion of objects when using DEF, the synonym for DEFINE.

Characters with special meanings

The following characters have special meaning when you build MQSC commands:

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blanks are used as separators. Multiple blanks are equivalent to a single blank, except in strings that have quotation marks (' ') round them.</td>
<td></td>
</tr>
<tr>
<td>Commas are used as separators. Multiple commas are equivalent to a single comma, except in strings that have quotation marks (' ') round them.</td>
<td></td>
</tr>
<tr>
<td>A single quotation mark indicates the beginning or end of a string. WebSphere MQ leaves all characters that have quotation marks round them exactly as they are entered. The containing quotation marks are not included when calculating the length of the string.</td>
<td></td>
</tr>
<tr>
<td>Two quotation marks together inside a string are treated by WebSphere MQ as one quotation mark, and the string is not terminated. The double quotation marks are treated as one character when calculating the length of the string.</td>
<td></td>
</tr>
<tr>
<td>An open parenthesis indicates the beginning of a parameter list.</td>
<td></td>
</tr>
<tr>
<td>A close parenthesis indicates the end of a parameter list.</td>
<td></td>
</tr>
<tr>
<td>A colon indicates an inclusive range. For example (1:5) means (1,2,3,4,5). This notation can be used only in TRACE commands.</td>
<td></td>
</tr>
<tr>
<td>An asterisk means “all”. For example, DISPLAY TRACE (<em>) means display all traces, and DISPLAY QUEUE (PAY</em>) means display all queues whose names begin with PAY.</td>
<td></td>
</tr>
</tbody>
</table>

When you need to use any of these special characters in a field (for example as part of a description), you must enclose the whole string in single quotation marks.

Building command scripts

You might want to build the MQSC commands into a script when you use:
Rules for using commands

- The CSQINP1, CSQINP2, and CSQINPX initialization data sets or the CSQUTIL batch utility on z/OS
- The MQSC STRMQM command on OS/400
- The runmqsc command on Compaq OpenVMS, Compaq NSK, Linux, OS/2 Warp, UNIX systems, and Windows

When you do this, follow these rules:
- Each command must start on a new line.
- On each platform, there might be platform-specific rules about the line length and record format. If scripts are to be readily portable to different platforms, the significant length of each line should be restricted to 72 characters.
  - On z/OS, scripts are held in a fixed-format data set, with a record length of 80. Only columns 1 through 72 can contain meaningful information; columns 73 through 80 are ignored.
  - On AIX, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows, each line can be of any length up to the maximum allowed for your platform, as determined by the value of BUFSIZ in stdio.h.
  - On other UNIX systems, and Compaq OpenVMS, each line can be of any length up to and including 80 characters.
  - On Compaq NSK each line can be of any length up to and including 72 characters.
- A line must not end in a keyboard control character (for example, a tab).
- If the last nonblank character on a line is:
  - A minus sign (−), this indicates that the command is to be continued from the start of the next line.
  - A plus sign (+), this indicates that the command is to be continued from the first nonblank character in the next line. If you use + to continue a command remember to leave at least one blank before the next parameter (except on z/OS where this is not necessary).

Either of these can occur within a parameter, data value, or quoted string. For example,

'Fr+
ed'

and

'Fr-
ed'

(where the ‘e’ of the second line of the second example is in the first position of the line) are both equivalent to

'Fred'

MQSC commands that are contained within an Escape PCF (Programmable Command Format) command cannot be continued in this way. The entire command must be contained within a single Escape command. (For information about the PCF commands, see the WebSphere MQ Programmable Command Formats and Administration Interface manual.)

- + and − values used at the ends of lines are discarded when the command is reassembled into a single string.
- On AIX, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows you can use a semicolon character (;) to terminate a command, even if you have entered a +
Rules for using commands

character at the end of the previous line. You can also use the semicolon in the same way on z/OS for commands issued from the CSQUTIL batch utility program.

• A line starting with an asterisk (*) in the first position is ignored. This can be used to insert comments into the file.

A blank line is also ignored.

If a line ends with a continuation character (− or +), the command continues with the next line that is not a comment line or a blank line.

Rules for naming WebSphere MQ objects

WebSphere MQ queue, process, namelist, channel, and storage class objects exist in separate object name spaces, and so objects from each type can all have the same name. However, an object cannot have the same name as any other object in the same name space. (For example, a local queue cannot have the same name as a model queue.) Names in WebSphere MQ are case sensitive; however, you should remember that lowercase characters that are not contained within quotation marks are folded to uppercase.

The character set that can be used for naming all WebSphere MQ objects is as follows:

• Uppercase A–Z
• Lowercase a–z (however, on systems using EBCDIC Katakana you cannot use lowercase characters, and there are also restrictions on the use of lowercase letters for z/OS console support)
• Numerics 0–9
• Period (.)
• Forward slash (/)
• Underscore (_)
• Percent sign (%). The percent sign (%) is a special character to RACF. If you are using RACF as the external security manager for WebSphere MQ for z/OS, you should not use % in object names. If you do, these names are not included in any security checks when RACF generic profiles are used.

Notes:
1. Leading or embedded blanks are not allowed.
2. Avoid using names with leading or trailing underscores, because they cannot be handled by the WebSphere MQ for z/OS operations and control panels.
3. Any name that is less than the full field length can be padded to the right with blanks. All short names that are returned by the queue manager are always padded to the right with blanks.
4. Any structure to the names (for example, the use of the period or underscore) is not significant to the queue manager.
5. When using CL commands or menus on OS/400 systems, lowercase a–z, forward slash (/), and percent (%) are special characters. If you use any of these characters in a name, the name must be enclosed in quotation marks. Lowercase a–z characters are changed to uppercase if the name is not enclosed in quotation marks.

Queue names

Queues can have names up to 48 characters long.
Reserved queue names

Names that start with “SYSTEM.” are reserved for queues defined by the queue manager. You can use the ALTER or DEFINE REPLACE commands to change these queue definitions to suit your installation. The following names are defined for WebSphere MQ:

<table>
<thead>
<tr>
<th>Queue Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM.ADMIN.CHANNEL.EVENT</td>
<td>Queue for channel events</td>
</tr>
<tr>
<td>SYSTEM.ADMIN.COMMAND.QUEUE</td>
<td>Queue to which PCF command messages are sent (not for z/OS)</td>
</tr>
<tr>
<td>SYSTEM.ADMIN.CONFIG.EVENT</td>
<td>Queue for configuration events</td>
</tr>
<tr>
<td>SYSTEM.ADMIN.PERFM.EVENT</td>
<td>Queue for performance events</td>
</tr>
<tr>
<td>SYSTEM.ADMIN.QMGR.EVENT</td>
<td>Queue for queue manager events</td>
</tr>
<tr>
<td>SYSTEM.CHANNEL.COMMAND</td>
<td>Queue used for distributed queuing on z/OS using CICS</td>
</tr>
<tr>
<td>SYSTEM.CHANNEL.INITQ</td>
<td>Queue used for distributed queuing on z/OS without CICS</td>
</tr>
<tr>
<td>SYSTEM.CHANNEL.SEQNO</td>
<td>Queue used for distributed queuing on z/OS using CICS</td>
</tr>
<tr>
<td>SYSTEM.CHANNEL.SYNCQ</td>
<td>Queue used for distributed queuing on z/OS without CICS</td>
</tr>
<tr>
<td>SYSTEM.CICS.INITIATION.QUEUE</td>
<td>Queue used for triggering (not for z/OS)</td>
</tr>
<tr>
<td>SYSTEM.CLUSTER.COMMAND.QUEUE</td>
<td>Queue used to communicate repository changes between queue managers (AIX, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS only)</td>
</tr>
<tr>
<td>SYSTEM.CLUSTER.REPOSITORY.QUEUE</td>
<td>Queue used to hold information about the repository (AIX, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS only)</td>
</tr>
<tr>
<td>SYSTEM.CLUSTER.TRANSMIT.QUEUE</td>
<td>Transmission queue for all destinations managed by cluster support (AIX, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS only)</td>
</tr>
<tr>
<td>SYSTEM.COMMAND.INPUT</td>
<td>Queue to which command messages are sent on z/OS</td>
</tr>
<tr>
<td>SYSTEM.COMMAND.REPLY.MODEL</td>
<td>Model queue definition for command replies (for z/OS)</td>
</tr>
<tr>
<td>SYSTEM.DEAD.LETTER.QUEUE</td>
<td>Dead-letter queue (not for z/OS)</td>
</tr>
<tr>
<td>SYSTEM.DEFAULT.ALIAS.QUEUE</td>
<td>Default alias queue definition</td>
</tr>
<tr>
<td>SYSTEM.DEFAULT.INITIATION.QUEUE</td>
<td>Queue used to trigger a specified process (not for z/OS)</td>
</tr>
<tr>
<td>SYSTEM.DEFAULT.LOCAL.QUEUE</td>
<td>Default local queue definition</td>
</tr>
<tr>
<td>SYSTEM.DEFAULT.MODEL.QUEUE</td>
<td>Default model queue definition</td>
</tr>
<tr>
<td>SYSTEM.DEFAULT.REMOTE.QUEUE</td>
<td>Default remote queue definition</td>
</tr>
<tr>
<td>SYSTEM.MQSC.REPLY.QUEUE</td>
<td>Model queue definition for MQSC command replies (not for z/OS)</td>
</tr>
<tr>
<td>SYSTEM.QSG.CHANNEL.SYNCQ</td>
<td>Shared local queue used for storing messages that contain the synchronization information for shared channels (z/OS only)</td>
</tr>
</tbody>
</table>
**Rules for naming objects**

| SYSTEM.QSG.TRANSMIT.QUEUE | Shared local queue used by the intra-group queuing agent when transmitting messages between queue managers in the same queue-sharing group (z/OS only) |

**Other object names**

Processes, namelists, clusters, and authentication information objects can have names up to 48 characters long. Channels can have names up to 20 characters long. Storage classes can have names up to 8 characters long.

**Reserved object names**

Names that start with “SYSTEM.” are reserved for objects defined by the queue manager. You can use the ALTER or DEFINE REPLACE commands to change these object definitions to suit your installation. The following names are defined for WebSphere MQ:

| SYSTEM.ADMIN.SVRCONN | Server-connection channel used for remote administration of a queue manager by the WebSphere MQ Explorer (remote administration is not available on z/OS) |
| SYSTEM.AUTO.RECEIVER | Default receiver channel for auto definition (AIX, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows only) |
| SYSTEM.AUTO.SVRCONN | Default server-connection channel for auto definition (AIX, HP-UX, Linux, OS/2 Warp, z/OS, OS/400, Solaris, and Windows only) |
| SYSTEM.DEF.CLNTCONN | Default client-connection channel definition |
| SYSTEM.DEF.CLUSRCVR | Default cluster-receiver channel definition (AIX, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS only) |
| SYSTEM.DEF.CLUSSDR | Default cluster-sender channel definition (AIX, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS only) |
| SYSTEM.DEF.RECEIVER | Default receiver channel definition |
| SYSTEM.DEF.REQUESTER | Default requester channel definition |
| SYSTEM.DEF.SENDER | Default sender channel definition |
| SYSTEM.DEF.SERVER | Default server channel definition |
| SYSTEM.DEF.SVRCONN | Default server-connection channel definition |
| SYSTEM.DEFAULT.AUTHINFO.CRLLDAP | Default authentication information definition |
| SYSTEM.DEFAULT.NAMELIST | Default namelist definition (AIX, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS only) |
| SYSTEM.DEFAULT.PROCESS | Default process definition |
| SYSTEMST | Default storage class definition (z/OS only) |
How to read syntax diagrams

This book contains syntax diagrams (sometimes referred to as “railroad” diagrams).

Each syntax diagram begins with a double right arrow and ends with a right and left arrow pair. Lines beginning with a single right arrow are continuation lines. You read a syntax diagram from left to right and from top to bottom, following the direction of the arrows.

Other conventions used in syntax diagrams are:

Table 1. How to read syntax diagrams

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Diagram A]</td>
<td>You must specify values A, B, and C. Required values are shown on the main line of a syntax diagram.</td>
</tr>
<tr>
<td>![Diagram B]</td>
<td>You may specify value A. Optional values are shown below the main line of a syntax diagram.</td>
</tr>
<tr>
<td>![Diagram C]</td>
<td>Values A, B, and C are alternatives, one of which you must specify.</td>
</tr>
<tr>
<td>![Diagram D]</td>
<td>Values A, B, and C are alternatives, one of which you might specify.</td>
</tr>
<tr>
<td>![Diagram E]</td>
<td>You might specify one or more of the values A, B, and C. Any required separator for multiple or repeated values (in this example, the comma (,) is shown on the arrow) is shown on the arrow.</td>
</tr>
<tr>
<td>![Diagram F]</td>
<td>You might specify value A multiple times. The separator in this example is optional.</td>
</tr>
<tr>
<td>![Diagram G]</td>
<td>Values A, B, and C are alternatives, one of which you might specify. If you specify none of the values shown, the default A (the value shown above the main line) is used.</td>
</tr>
</tbody>
</table>
Syntax diagrams

Table 1. How to read syntax diagrams (continued)

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td>The syntax fragment Name is shown separately from the main syntax diagram.</td>
</tr>
</tbody>
</table>
| ![Diagram](image)   | Specifying exactly as shown.
| ![Diagram](image)   | Supply your own text in place of the name variable.                    |
Chapter 2. The MQSC commands

This chapter describes, in alphabetic order, all the MQSC commands that can be issued by operators and administrators.
Use ALTER AUTHINFO to alter an authentication information object. These objects contain the definitions required to perform Certificate Revocation List (CRL) checking using LDAP servers, except on OS/400 where these are defined by the Digital Certificate Manager for each Certification Authority.

**Synonym:** ALT AUTHINFO

**ALTER AUTHINFO**

```plaintext
ALTER AUTHINFO(name) [CONNAME(string)] [DESCR(string)] [LDAPPWD(string)]
```

```plaintext
[LDAPUSER(string)]
```

```plaintext
[CMDSCOPE('') (1)]
[QSGDISP(QMGR) (1)]
```

```plaintext
[CMDSCOPE(qmgr-name) (1)]
[QSGDISP(COPY) (2)]
```

```plaintext
[CMDSCOPE(*) (2)]
[QSGDISP(GROUP)]
[QSGDISP(PRIVATE)]
```

**Notes:**

1. Valid only on z/OS.
2. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.

**Parameter descriptions**

For a description of the parameters see "DEFINE AUTHINFO" on page 62
ALTER CFSTRUCT

Use ALTER CFSTRUCT to alter the CF application structure backup and recovery parameters for any specified application structure.

Notes:
1. This command cannot be run from the CSQINP1 initialization input data set.
2. This command cannot specify the CF administration structure (CSQ_ADMIN).

Synonym: ALT CFSTRUCT

```
ALTER CFSTRUCT

/SM590000/SM590000
ALTER CFSTRUCT(
structure-name
) CFLEVEL(integer) DESCR(string)
/SM590000/SM630000
RECOVER( NO ) YES
```

Keyword and parameter descriptions
For a description of the parameters see “DEFINE CFSTRUCT” on page 67.
Use ALTER CHANNEL to alter the parameters of a channel.

Notes:
1. On z/OS, this command is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, the equivalent function is available using the CKMC transaction. See the WebSphere MQ Intercommunication manual.
2. For cluster-sender channels, you can only alter channels that have been created manually.
3. If you change the XMITQ name or the CONNAME, you must reset the sequence number at both ends of the channel. (See “RESET CHANNEL” on page 281 for information about the SEQNUM parameter.)

Synonym: ALT CHL

There is a separate syntax diagram for each type of channel:
- “Sender channel” on page 13
- “Server channel” on page 15
- “Receiver channel” on page 17
- “Requester channel” on page 19
- “Client-connection channel” on page 22
- “Server-connection channel” on page 24
- “Cluster-sender channel” on page 26
- “Cluster-receiver channel” on page 28
Sender channel

**ALTER CHANNEL**

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR)

BATCHHB(integer)

CMDSCOPE('') (1)

BATCHINT(integer) BATCHSZ(integer)

CMDSCOPE(qmgr-name) (3)

CMDSCOPE(*) (3)

CMDSCOPE(qmgr-name)

CONNAME(string)

CONVERT( NO ) YES

DESCR(string)

DISCINT(integer)

HBINT(integer) KAINT(integer) AUTO

LONGRTY(integer) LONGTMR(integer)

MAXMSGL(integer)

MCANAME(string)

MCATYPE(PROCESS) THREAD

MCAUSER(string)

MODENAME(string)

MSGDATA(string)

MSGEXIT(string)

NPMSPEED(FAST) NORMAL

QSGDISP(QMGR) COPY GROUP PRIVATE

PASSWORD(string)

RCVDATA(string) RCVEXIT(string)

SCYDATA(string)
```
ALTER CHANNEL

SCYEXIT(string)
SENDATA(string)
SENDEXIT(string)
SEQWRAP(integer)
SHORTRTY(integer)
SHORTTMR(integer)
SSLCIPH(string)
SSLPEER(string)
TPNAME(string)
TRPTYPE(DECNET)
LU62
NETBIOS
SPX
TCP
UDP
USERID(string)
XMITQ(string)

Notes:
1 This parameter must follow immediately after the channel name except on z/OS.
2 Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows and z/OS.
3 Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
4 Valid only on z/OS.
5 Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
6 Valid only if TRPTYPE is LU62.
7 You can specify more than one value on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS only.
8 Not valid on z/OS.
9 Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
10 Valid only on Compaq OpenVMS.
11 Valid only on OS/2 Warp and Windows.
12 Valid only on AIX.
Chapter 2. The MQSC commands
Notes:

1. This parameter must follow immediately after the channel name except on z/OS.
2. Valid only on Compaq NSK when TRPTYPE is LU62.
3. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows and z/OS.
4. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
5. Valid only on z/OS.
6. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
7. Valid only if TRPTYPE is LU62.
8. You can specify more than one value on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS only.
9. Not valid on z/OS.
10. Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
11. Valid only on Compaq OpenVMS.
12. Valid only on OS/2 Warp and Windows.
13. Valid only on AIX.
Receiver channel

**ALTER CHANNEL**

```
ALTER CHANNEL(channel-name) CHLTYPE(RCVR)

(1) AUTOSTART(DISABLED)

(2)

CHLTYPE(RCVR)

(3) CMDSCOPE(' ')

CMDSCOPE(qmgr-name)

CMDSCOPE(*)

(4) HBINT(integer)

KAINT(integer)

MAXMSGL(integer)

(5) MCAUSER(string)

MRDATA(string)

MREXIT(string)

(6) MRRTY(integer)

MRTMR(integer)

MSGDATA(string)

(7) MSGEXIT(string)

NPMSPEED(FAST)

NORMAL

PUTAUT(DEF)

CTX

ONLYMCA

ALTTCMA

QSGDISP(QMGR)

QSGDISP(COPY)

QSGDISP(GROUP)

QSGDISP(PRIVATE)

RCVDATA(string)

(8)

```
ALTER CHANNEL

Notes:
1. This parameter must follow immediately after the channel name except on z/OS.
2. Valid only on Compaq NSK when TRPTYPE is LU62.
3. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
4. Valid only on z/OS.
5. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
6. Not valid on z/OS.
7. You can specify more than one value on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS only.
8. Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
9. Valid only on Compaq OpenVMS.
10. Valid only on OS/2 Warp and Windows.
11. Valid only on AIX.
ALTER CHANNEL

Requester channel

ALTER CHANNEL(channel-name) — CHLTYPE(RQSTR)

AUTOSTART(DISABLED — ENABLED)

BATCHSZ(integer)

CMDSCOPE(' ')

CMDSCOPE(qmgr-name)

CMDSCOPE(*)

CONNAME(string)

DESCR(string)

HBINT(integer)

KAINT(AUTO)

LOCLADDR(string)

MAXMSGL(integer)

MCANAME(string)

MCATYPE(PROCESS — THREAD)

MCAUSER(string)

MODENAME(string)

MRDATA(string)

MREXIT(string)

MRRTY(integer)

MRTMR(integer)

MSGDATA(string)

MSGEXIT(string)

NPMSPEED(FAST — NORMAL)

PASSWORD(string)

PUTAUT(DEF — CTX — ONLYMCA — ALTMCA)
**ALTER CHANNEL**

Notes:

1. This parameter must follow immediately after the channel name except on z/OS.
2. Valid only on Compaq NSK when TRPTYPE is LU62.
3. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
4. Valid only on z/OS.
5. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
6. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
7. Valid only if TRPTYPE is LU62.
8. Not valid on z/OS.
You can specify more than one value on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS only.

Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.

Valid only on Compaq OpenVMS.

Valid only on OS/2 Warp and Windows.

Valid only on AIX.
ALTER CHANNEL

Client-connection channel

ALTER CHANNEL

```sql
ALTER CHANNEL(channel-name) CHLTYPE(CLNTCONN) (1)

CMDSCOPE('') (3)

CHLTYPE(CLNTCONN) CMDSCOPE(qmgr-name) (2)

CMDSCOPE(*) (3)

CONNAME(string) DESCR(string) HBINT(integer) (4)

KAINT(integer) LOCLADDR(string) MAXMSGL(integer) (5)

MODENAME(string) PASSWORD(string) QMNAME(string) (5)

QSGDISP(QMGR) (3)

QSGDISP(COPY) QSGDISP(GROUP) QSGDISP(PRIVATE) (2)

RCVDATA(string) (6)

RCVEXIT(string) SCYDATA(string) SCYEXIT(string) (6)

SENDDATA(string) (6)

SENDEXIT(string) SSLCIPH(string) SSLPEER(string) (7)

TPNAME(string) TRPTYPE(DECNET LU62 NETBIOS SPY TCP) (8)

USERID(string) (5)
```
Notes:

1. This parameter must follow immediately after the channel name except on z/OS.
2. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
3. Valid only on z/OS.
4. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
5. Valid only if TRPTYPE is LU62.
6. You can specify more than one value on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS only.
7. Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
8. Valid only on Compaq OpenVMS.
9. Valid only for clients to be run on DOS, OS/2 Warp, and Windows.
Server-connection channel

ALTER CHANNEL

(1) ALTER CHANNEL

ALTER CHANNEL(channel-name) CHLTYPE(SVRCONN)

(2) AUTOSTART

AUTOSTART(DISABLED, ENABLED)

(3) CMDSCOPE

CMDSCOPE(' ')

CMDSCOPE(qmgr-name)

CMDSCOPE(+)

(4) CMDSCOPE

CMDSCOPE(' ')

CMDSCOPE(qmgr-name)

CMDSCOPE(+)

(5) DESCR

DESCR(string)

HBINT

HBINT(integer)

KAINT

KAINT(integer)

(6) MAXMSGL

MAXMSGL(integer)

MCAUSER

MCAUSER(string)

QSGDISP

QSGDISP(QMGR)

QSGDISP(COPY)

QSGDISP(GROUP)

QSGDISP(PRIVATE)

RCVEXIT

RCVEXIT(string)

RCVOATA

RCVOATA(string)

SENDATA

SENDATA(string)

SENDEXIT

SENDEXIT(string)

(7) SSLCAUTH

SSLCAUTH(OPTIONAL, REQUIRED)

(8) TRPTYPE

TRPTYPE(DECNET, LU62, NETBIOS, SPX, TCP)

(9) SSLCIPH

SSLCIPH(string)

SSLPEER

SSLPEER(string)
Notes:

1. This parameter must follow immediately after the channel name except on z/OS.
2. Valid only on Compaq NSK when TRPTYPE is LU62.
3. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
4. Valid only on z/OS.
5. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
6. You can specify more than one value on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS only.
7. Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
8. Valid only on Compaq OpenVMS.
9. Valid only for clients to be run on DOS, OS/2 Warp, and Windows.
ALTER CHANNEL

Cluster-sender channel

ALTER CHANNEL

(1) (2)

ALTER CHANNEL(channel-name)—CHLTYP(CLUSSDR)

BATCHHB(integer)

BATCHINT(integer)  BATCHSZ(integer)  CLUSNL(nlname)

CMDSCOPE(' ')

CONNAME(string)

CONV( NO )  YES

DESCR(string)  DISCINT(integer)  HBINT(integer)

KAINT(integer)

LOCLADDR(string)  LONGRTY(integer)

LONGTMR(integer)  MAXMSGL(integer)

MCANAME(string)

MCATYPE(PROCESS)  THREAD

MCAUSER(string)

MODENAME(string)

MSGDATA(string)  MSGEXIT(string)

NPMSPEED(FAST)  NORMAL

PASSWORD(string)

QSGDISP(QMGR)  COPY  GROUP

RCVDATA(string)  SCYDATA(string)

RCVEXIT(string)
ALTER CHANNEL

- SCYEXIT(string)
- SENDDATA(string)
- SENDEXIT(string)
- SEQWRAP(integer)
- SHORTRY(integer)
- SHORTTMR(integer)
- SSLCIPH(string)
- SSLPEER(string)
- TPNAME(string)
- TRPTYPE(LU62, NETBIOS, SPX, TCP, UDP)
- USERID(string)

Notes:
1. This parameter must follow immediately after the channel name except on z/OS.
2. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
3. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
4. Valid only on z/OS.
5. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
6. Valid only if TRPTYPE is LU62.
7. Not valid on z/OS.
8. Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
9. Valid only on OS/2 Warp and Windows.
10. Valid only on AIX.

Chapter 2. The MQSC commands 27
ALTER CHANNEL

Cluster-receiver channel

ALTER CHANNEL
Notes:
1 This parameter must follow immediately after the channel name except on z/OS.
2 Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
3 Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
4 Valid only on z/OS.
5 Valid only if TRPTYPE is LU62.
6 Not valid on z/OS.
7 Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
8 Valid only on AIX.
9 Valid only on OS/2 Warp and Windows.

Parameter descriptions
For a description of the parameters see “DEFINE CHANNEL” on page 70.
ALTER NAMELIST

Use ALTER NAMELIST to alter a list of names. This is most commonly a list of cluster names or queue names.

Notes:
1. On UNIX systems, the command is valid only on AIX, HP-UX, and Solaris.

Synonym: ALT NL

ALTER NAMELIST

Parameter descriptions

For a description of the parameters see “DEFINE NAMELIST” on page 114
Use ALTER PROCESS to alter the parameters of an existing WebSphere MQ process definition.

**Synonym:** ALT PRO

**ALTER PROCESS**

```
ALTER PROCESS(process-name) [APPLICID(string)] [APPLTYPE(integer)]
```

- **CICS**
- **DEF**
- **DOS**
- **IMS**
- **MVS**
- **NOTESAGENT**
- **NSK**
- **OPENVMS**
- **OS2**
- **OS/400**
- **UNIX**
- **WINDOWS**
- **WINDOWS NT**

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.

**Parameter descriptions**

For a description of the parameters see "DEFINE PROCESS" on page 118.
Use ALTER QMGR to alter the queue manager parameters for the local queue manager.

**Synonym:** ALT QMGR

### ALTER QMGR

#### Qmgr attrs:

- **AUTHOREV**
  - **ENABLED**
  - **DISABLED**

- **CCSID**
  - **integer**

- **CHADEV**
  - **DISABLED**
  - **ENABLED**

- **CHADEXIT**
  - **string**

- **CLWLLEN**
  - **integer**

- **CONFIGEV**
  - **ENABLED**
  - **DISABLED**

- **DEADQ**
  - **string**

- **DEFXMITQ**
  - **string**

- **DESCR**
  - **string**

- **EXPRYINT**
  - **OFF**
  - **integer**

- **IGQ**
  - **DISABLED**
  - **ENABLED**

- **IGQUOT**
  - **DEF**
  - **CTX**
  - **ONLYIGQ**
  - **ALTIQG**

- **IGQUSER**
  - **string**
Notes:
1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.
3. Not valid on z/OS.
4. Valid only on AIX, Compaq NSK, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
5. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
6. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
7. Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
8. Valid only on AIX, HP-UX, Linux, and Solaris.

Parameter descriptions
The parameters you specify override the current values. Attributes that you do not specify are unchanged.

Notes:
1. If you do not specify any parameters, the command completes successfully, but no queue manager options are changed.
2. Changes made using this command persist when the queue manager is stopped and restarted.

FORCE
Specify this to force completion of the command if both of the following are true:
- The DEFXMITQ parameter is specified
- An application has a remote queue open, the resolution for which would be affected by this change
ALTER QMGR

If FORCE is not specified in these circumstances, the command is unsuccessful.

Queue manager parameters

AUTHOREV
Whether authorization (Not Authorized) events are generated:

ENABLED
Authorization events are generated.
This value is not supported on z/OS.

DISABLED
Authorization events are not generated. This is the queue manager’s initial default value.

CCSID(integer)
The coded character set identifier for the queue manager. The CCSID is the identifier used with all character string fields defined by the API. It does not apply to application data carried in the text of messages unless the CCSID in the message descriptor is set to the value MQCCS.Q_MGR when the message is put to a queue.

Specify a value in the range 1 through 65 535. The CCSID must specify a value that is defined for use on your platform, and use a character set that is appropriate to the platform.

If you use this parameter to change the CCSID, applications that are running when the change is applied continue to use the original CCSID. Because of this, you must stop and restart all running applications before you continue. This includes the command server and channel programs. To do this, stop and restart the queue manager after making the change.

This parameter is valid only on AIX, Compaq NSK, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows. See the WebSphere MQ Application Programming Guide for details of the supported CCSIDs for each platform.

CHAD
Whether receiver and server-connection channels can be defined automatically:

DISABLED
Auto-definition is not used. This is the queue manager’s initial default value.

ENABLED
Auto-definition is used.

Cluster-sender channels can always be defined automatically, regardless of the setting of this parameter.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.

CHADEV
Whether channel auto-definition events are generated.

DISABLED
Auto-definition events are not generated. This is the queue manager’s initial default value.
ENABLED
Auto-definition events are generated.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.

CHADEXIT(string)
Auto-definition exit name.

If this name is nonblank, the exit is called when an inbound request for an undefined receiver, server-connection, or cluster-sender channel is received. It is also called when starting a cluster-receiver channel.

The format and maximum length of the name depends on the environment:
- On OS/2 Warp and Windows, it is of the form dllname(functionname) where dllname is specified without the suffix (".DLL"). The maximum length is 128 characters.
- On OS/400, it is of the form:
  
  programname libname

  where programname occupies the first 10 characters and libname the second 10 characters (both blank-padded to the right if necessary). The maximum length of the string is 20 characters.
- On AIX, Compaq OpenVMS, HP-UX, Linux, and Solaris, it is of the form libraryname(functionname). The maximum length is 128 characters.
- On z/OS, it is a load module name, the maximum length is 8 characters.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS. On z/OS, it applies only to cluster-sender and cluster-receiver channels.

CLWLDATA(string)
Cluster workload exit data. The maximum length of the string is 32 characters.

This string is passed to the cluster workload exit when it is called.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

CLWLEXIT(string)
Cluster workload exit name.

If this name is nonblank, the exit is called when a message is put to a cluster queue. The format and maximum length of the name depends on the environment:
- On UNIX systems and Compaq OpenVMS, it is of the form
  
  libraryname(functionname). The maximum length is 128 characters.
- On OS/2 Warp and Windows, it is of the form dllname(functionname), where dllname is specified without the suffix (".DLL"). The maximum length is 128 characters.
- On z/OS, it is a load module name. The maximum length is 8 characters.
- On OS/400, it is of the form:
  
  programname libname
ALTER QMGR

where program name occupies the first 10 characters and libname the second 10 characters (both blank-padded to the right if necessary). The maximum length is 20 characters.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

CLWLLLEN(integer)

The maximum number of bytes of message data that is passed to the cluster workload exit.

Specify a value:
• in the range zero through 100 MB on WebSphere MQ for z/OS systems
• in the range zero through 999 999 999 on other platforms

The initial default value is 100.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows and z/OS.

CMDSCOPE

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

' '

The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

*

The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

CONFIGEV

Whether configuration events are generated:

ENABLED

Configuration events are generated. After setting this value, issue REFRESH QMGR TYPE(CONFIGEV) commands for all objects to bring the queue manager configuration up-to-date.

DISABLED

Configuration events are not generated. This is the queue manager’s initial default value.

This parameter applies only to z/OS.

DEADQ(string)

The local name of a dead-letter queue (or undelivered-message queue) on which messages that cannot be routed to their correct destination are put.

The queue named must be a local queue. See “Rules for naming WebSphere MQ objects” on page 4.
DEFXMITQ(string)
Local name of the default transmission queue on which messages destined for a remote queue manager are put, if there is no other suitable transmission queue defined.

The queue named must be a local transmission queue. See “Rules for naming WebSphere MQ objects” on page 4.

DESCR(string)
Plain-text comment. It provides descriptive information about the queue manager.

It should contain only displayable characters. The maximum length of the string is 64 characters. In a DBCS installation, it can contain DBCS characters (subject to a maximum length of 64 bytes).

Note: If characters are used that are not in the coded character set identifier (CCSID) for this queue manager, they might be translated incorrectly if the information is sent to another queue manager.

EXPRYINT
Specifies how often queues are scanned for expired messages:

OFF Queues are not scanned. No internal expiry processing is performed. This is the default.

integer The approximate interval in seconds at which queues are scanned. The value must be in the range 1 through 999999999. The minimum scan interval used is 5 seconds, even if you specify a lower value.

Note: You must set the same EXPRYINT value for all queue managers within a queue-sharing group that support this attribute.

Changes to EXPRYINT take effect when the current interval expires, unless the new interval is less than the unexpired portion of the current interval, in which case a scan is scheduled immediately and the new interval value takes immediate effect.

This parameter is supported only on z/OS.

IGQ Whether intra-group queuing is used.
This parameter applies only to z/OS.

ENABLED Message transfer between queue managers within a queue-sharing group uses the shared transmission queue (SYSTEM.QSG.TRANSMIT.QUEUE).

DISABLED Message transfer between queue managers within a queue-sharing group uses non-shared transmission queues and channels. This is the same mechanism used for message transfer between queue managers that are not part of a queue-sharing group. This is the default value.

IGQAUT Specifies the type of authority checking and, therefore, the user IDs, to be used by the IGQ agent (IGQA). This establishes the authority to put messages to a destination queue.
This parameter applies only to z/OS.

**DEF** Indicates that the default user ID should be used to establish authority to put messages to a destination queue. This is the default value.

For one user ID check, this involves using the user ID (referred to as QSGSEND) of the queue manager within the queue-sharing group that put the messages to the SYSTEM.QSG.TRANSMIT.QUEUE.

For two user ID checks, this involves using the QSGSEND user ID and the IGQ user ID.

**CTX** Indicates that the user ID from the UserIdentifier field in the message descriptor, of a message on the SYSTEM.QSG.TRANSMIT.QUEUE, should be used to establish authority to put messages to a destination queue.

For one user ID check, this involves using the QSGSEND user ID.

For two user ID checks, this might involve using the QSGSEND user ID, the IGQ user ID and the alternate user id (referred to as ALT) taken from the UserIdentifier field in the message descriptor of a message on the SYSTEM.QSG.TRANSMIT.QUEUE.

**ONLYIGQ** Indicates that only the IGQ user ID should be used to establish authority to put messages to a destination queue.

For all ID checks, this involves using the IGQ user ID.

**ALTIGQ** Indicates that the IGQ user ID and the ALT user ID should be used to establish authority to put messages to a destination queue.

For one user ID check, this uses the IGQ user ID.

For two user ID checks, this uses the IGQ user ID and the ALT user ID.

**IGQUSER** Nominates a user ID (referred to as the IGQ user ID) to be used by the IGQ agent (IGQA) to establish authority to put messages to a destination queue.

This parameter applies only to z/OS. Possible values are:

**Blanks**

This is the default value for the IGQ user ID and indicates that the user ID of the receiving queue manager within the QSG should be used.

**Specific user ID** Indicates that the user ID specified in the receiving queue manager’s IGQUSER parameter should be used.

**Notes:**

1. As the receiving queue manager has authority to all queues it can access, this means that security checking might not be performed for this user ID type.

2. As the default value of blanks has a special meaning, you cannot use IGQUSER to specify a real user ID of blanks.
INHIBTEV
Whether inhibit (Inhibit Get and Inhibit Put) events are generated:

ENABLED
Inhibit events are generated.

DISABLED
Inhibit events are not generated. This is the queue manager’s initial default value.

LOCALEV
Whether local error events are generated:

ENABLED
Local error events are generated.

DISABLED
Local error events are not generated. This is the queue manager’s initial default value.

MAXHANDS(integer)
The maximum number of open handles that any one task can have at the same time.

This is a value in the range zero through 999 999 999.

MAXMSGL(integer)
The maximum length of messages allowed on queues for this queue manager.

This is in the range 32 KB through 100 MB. The default is 4 MB (4 194 403 bytes).

If you reduce the maximum message length for the queue manager, you must also reduce the maximum message length of the SYSTEM.DEFAULT.LOCAL.QUEUE definition, and all other queues connected to the queue manager. This ensures that the queue manager’s limit is not less than that of any of the queues associated with it. If you do not do this, and applications inquire only the value of the queue’s MAXMSGL, they might not work correctly.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.

MAXUMSGS(integer)
The maximum number of uncommitted messages within a syncpoint.

This is a limit on
• the number of messages that can be retrieved, plus
• the number of messages that can be put
within any single syncpoint. It does not apply to messages that are put or retrieved outside syncpoint.

The number includes any trigger messages and report messages generated within the same unit of recovery.

Specify a value in the range 1 through 999 999 999.

PERFMEV
Whether performance-related events are generated:
ALTER QMGR

ENABLED
   Performance-related events are generated.

DISABLED
   Performance-related events are not generated. This is the queue
   manager's initial default value.

Note: On WebSphere MQ for z/OS all the queue managers in a
queue-sharing group should have the same setting.

REMOTE EV
Whether remote error events are generated:

ENABLED
   Remote error events are generated.

DISABLED
   Remote error events are not generated. This is the queue manager's
   initial default value.

REPOS(clustername)
The name of a cluster for which this queue manager provides a repository
manager service. The maximum length is 48 characters conforming to the
rules for naming WebSphere MQ objects.

No more than one of the resultant values of REPOS can be nonblank.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux,
OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

REPOSNL(nlname)
The name of a namelist of clusters for which this queue manager provides
a repository manager service.

No more than one of the resultant values of REPOSNL can be nonblank.

If both REPOS and REPOSNL are blank, or REPOS is blank and the
namelist specified by REPOSNL is empty, this queue manager does not
have a full repository, but might be a client of other repository services that
are defined in the cluster.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux,
OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

SSLCRLNL(nlname)
The name of a namelist of AUTHINFO objects to be used for CRL checking
by the queue manager.

This parameter is supported only on AIX, HP-UX, Linux, OS/400, Solaris,
Windows, and z/OS.

If SSLCRLNL is blank, CRL checking is not invoked.

Changes to SSLCRLNL, or to the names in a previously specified namelist,
or to previously referenced AUTHINFO objects become effective either:
   • On OS/400, Windows and UNIX platforms, when a new channel process
     is started.
   • For channels that run as threads of the channel initiator on OS/400,
     Windows and UNIX platforms, when the channel initiator is restarted.
   • For channels that run as threads of the listener on OS/400, Windows
     and UNIX platforms, when the listener is restarted.
   • On z/OS, when the channel initiator is restarted.
**SSLCRYP**(string)

Sets the name of the parameter string required to configure the cryptographic hardware present on the system.

This parameter is supported on AIX, HP-UX, Linux, and Solaris only.

The string can have one of the following values:

- GSK_ACCELERATOR_RAINBOW_CS_OFF
- GSK_ACCELERATOR_RAINBOW_CS_ON
- GSK_ACCELERATOR_NCIPHER_NF_OFF
- GSK_ACCELERATOR_NCIPHER_NF_ON
- GSK_PKCS11=<the PKCS11 driver path>;<the PKCS11 token label>;<the PKCS11 token password>;

The strings containing “RAINBOW” enable or disable the Rainbow cryptographic hardware. If the Rainbow cryptographic hardware present, it is enabled by default.

The strings containing “NCIPHER” enable or disable the nCipher cryptographic hardware. If the nCipher cryptographic hardware is present, it is enabled by default.

To use cryptographic hardware using the PKCS11 interface, you must specify the string containing ”PKCS11”. The PKCS11 driver path must be an absolute path.

The maximum length of the string is 256 characters. The default value is blank.

If you specify a string that does not begin with one of the cryptographic strings listed above, you get an error. If you specify the GSK_PKCS11 string, the syntax of the other parameters is also checked.

When the SSLCRYP value is changed, the cryptographic hardware parameters specified become the ones used for new SSL connection environments. The new information becomes effective:

- When a new channel process is started.
- For channels that run as threads of the channel initiator, when the channel initiator is restarted.
- For channels that run as threads of the listener, when the listener is restarted.

**SSLKEYR**(string)

The name of the Secure Sockets Layer key repository.

This parameter is supported only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.

The maximum length of the string is 256 characters.

The format of the name depends on the environment:

- On z/OS, it is the name of a key ring.
- On UNIX platforms, it is of the form pathname/keyfile, where keyfile is specified without the suffix (".kdb"), and identifies a GSKit key database file. The default value is /var/mqm/qmgrs/QMGR/ssl/key, where QMGR is replaced by the queue manager name.
On OS/400, it is of the form `pathname/keyfile`, where `keyfile` is specified without the suffix (".kdb"), and identifies a GSKit key database file. The default value is `/QIBM/UserData/ICSS/Cert/Server/Default`.

On Windows, the key database is held in a Microsoft Certificate Store file, which has a filename of the form `xxx.sto`, where `xxx` is your chosen name. The SSLKEYR attribute is the path to this file along with the filename stem, (that is, all characters in the filename up to but not including the .sto file extension). WebSphere MQ automatically appends the .sto suffix.

On OS/400, Windows and UNIX platforms, the syntax of this parameter is validated to ensure that it contains a valid, absolute, directory path.

If SSLKEYR is blank, or is set to a value that does not correspond to a key ring or key database file, channels using SSL fail to start.

Changes to SSLKEYR become effective either:
- On OS/400, Windows and UNIX platforms, when a new channel process is started.
- For channels that run as threads of the channel initiator on OS/400, Windows and UNIX platforms, when the channel initiator is restarted.
- For channels that run as threads of the listener on OS/400, Windows and UNIX platforms, when the listener is restarted.
- On z/OS, when the channel initiator is restarted.

**SSLTASKS** *(integer)*

The number of server subtasks to use for processing SSL calls. To use SSL channels, you must have at least two of these tasks running.

This parameter is valid only on z/OS.

This is in the range zero through 9999.

Changes to this parameter will be effective when the channel initiator is restarted.

**STRSTPEV**

Whether start and stop events are generated:

**ENABLED**

Start and stop events are generated. This is the queue manager’s initial default value.

**DISABLED**

Start and stop events are not generated.

**TRIGINT** *(integer)*

A time interval expressed in milliseconds.

The TRIGINT parameter is relevant only if the trigger type (TRIGTYPE) is set to FIRST (see “DEFINE QLOCAL” on page 127 for details). In this case trigger messages are normally generated only when a suitable message arrives on the queue, and the queue was previously empty. Under certain circumstances, however, an additional trigger message can be generated with FIRST triggering even if the queue was not empty. These additional trigger messages are not generated more often than every TRIGINT milliseconds. See the [WebSphere MQ Application Programming Guide](#) for more information.

Specify a value in the range zero through 999 999 999.
ALTER Queues

This section contains the following commands:

- "ALTER QALIAS"
- "ALTER QLOCAL" on page 45
- "ALTER QMODEL" on page 47
- "ALTER QREMOTE" on page 49

These queues are supported on the following platforms:

<table>
<thead>
<tr>
<th>Compaq NSK</th>
<th>Compaq OpenVMS</th>
<th>OS/400</th>
<th>OS/2 Warp</th>
<th>UNIX systems</th>
<th>Windows</th>
<th>z/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

**ALTER QALIAS**

Use ALTER QALIAS to alter the parameters of an alias queue.

**Synonym:** ALT QA

**ALTER QALIAS**

ALTER QALIAS(q-name)

Common q attrs:

- DEFPRTY(integer)
- DEFPSIST(NO, YES)
- DESCR(string)
- PUT(ENABLED, DISABLED)

Alias q attrs:

- CLUSNL(nlname)
- CLUSTER(clustername)
- DEFBIND(OPEN, NOTFIXED)
- GET(ENABLED, DISABLED)
- SCOPE(QMGR, CELL)
- TARGQ(string)

Notes:

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.
ALTER Queues

3 Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

4 Valid only on Compaq OpenVMS, OS/2 Warp, OS/400, UNIX systems, and Windows.

5 Valid only on Compaq OpenVMS, OS/2 Warp, UNIX systems, and Windows.
ALTER QLOCAL

Use ALTER QLOCAL to alter the parameters of a local queue.

Synonym: ALT QL

ALTER QLOCAL

 ALTER QLOCAL(q-name)

FORCE

CMDSCOPE(' ')

CMDSCOPE(qmgr-name)

CMDSCOPE(*)

QSGDISP(QMGR)

QSGDISP(COPY)

QSGDISP(GROUP)

QSGDISP(PRIVATE)

QSGDISP(SHARED)

common q attrs

local q attrs

Common q attrs:

- DEFPRTY(integer)
- DEFPSIST( NO — YES )
- DESCR(string)
- PUT( ENABLED — DISABLED )

Local q attrs:

- BOQNAME(string)
- BOTHRESH(integer)
- CFSTRUCT(name)
- CLUSNL(iname)

- CLUSTER(clustername)
- DEFBIND( OPEN — NOTFIXED )
- DEFSOPT( EXCL — SHARED )

- DISTL( NO — YES )
- GET( ENABLED — DISABLED )
- HARDENBO — NOHARDENBO

- INDXTYPE( CORRELID — GROUPID — MSGID — MSGTOKEN — NONE )

- MSGDLVSQ( PRIORITY — FIFO )
- PROCESS(string)
- QDEPTHHI(integer)
- QDEPTHLD(integer)

- QDPIHEV( ENABLED — DISABLED )
- QDPLOEV( ENABLED — DISABLED )
- QDPMAXEV( ENABLED — DISABLED )

- QSVCIEV( NONE — HIGH — OK )
- QSVCINT(integer)
- RETINTVL(integer)
- SHARE — NOSHARE
ALTER Queues

```
SCOPE (QMGR)

CELL

STGCLASS (string)

TRIGDATA (string)

TRIGDPTH (integer)

TRIGGER

NOTRIGGER

TRIGMPRI (integer)

TRIGTYPE (FIRST, EVERY, DEPTH, NONE)

USAGE (NORMAL, XMITQ)
```

Notes:

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.
3. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
4. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
5. Valid only on Compaq OpenVMS, OS/2 Warp, OS/400, UNIX systems, and Windows.
6. Valid only on Compaq OpenVMS, OS/2 Warp, UNIX systems, and Windows.
ALTER QMODEL

Use ALTER QMODEL to alter the parameters of a model queue.

Synonym: ALTQM

ALTER QMODEL

**Common q attrs:**

- DEFPRTY(integer)
- DEFPS IST( NO )
- DESCR(string)
- PUT( ENABLED )

**Local q attrs:**

- BOQNAME(string)
- BOTHRESH(integer)
- CTSTRUCT(name)
- INITQ(string)
- MAXDEPTHi(integer)
- MAXMSGL(integer)
- MSGDLVSQ(PRIORITY)
- PROCESS(string)
- QDEPThHi(integer)
- QDEPThLO(integer)
- QOFHIEV(ENABLED)
- QOPLOEV(ENABLED)
- QDPMAXEV(ENABLED)
- QSVCIEV(NONE)
- QSVCINT(integer)
- RETINTVL(integer)
- STGCLASS(string)
- TRIGDATA(string)
- TRIQDPHT(integer)
- TRIGGER
- TRIGDPTH
- TRIGLVSQ(PRIORITY)
ALTER Queues

Model q attr:

Notes:
1 Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2 Valid only on z/OS.
3 Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
ALTER QREMOTE

Use ALTER QREMOTE to alter the parameters of a local definition of a remote queue, a queue manager alias, or a reply-to queue alias.

**Synonym:** ALT QR

**ALTER QREMOTE**

```
ALTER QREMOTE(q-name)
```

```
CMDSCOPE( ' ' )
FORCE
CMDSCOPE(qmgr-name)
CMDSCOPE(*)
```

```
QSGDISP(QMGR)
QSGDISP(COPY)
QSGDISP(GROUP)
QSGDISP(PRIVATE)
```

**Common q attrs:**

- `DEFPRTY(integer)`
- `DEFPSIST(NO)`
- `DESCR(string)`
- `PUT(ENABLED)`

**Remote q attrs:**

- `CLUSNL(nlname)`
- `CLUSTER(clustername)`
- `DEFBIND(OPEN)`
- `DEFBIND(NOTFIXED)`
- `SCOPE(QMGR)`
- `SCOPE(CELL)`
- `XMITQ(string)`
- `RNAME(string)`
- `RQMNAME(string)`

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.
3. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
4. Valid only on Compaq OpenVMS, OS/2 Warp, OS/400, UNIX systems, and Windows.
5. Valid only on Compaq OpenVMS, OS/2 Warp, UNIX systems, and Windows.

**Parameter descriptions**

For a description of the parameters see “DEFINE queues” on page 125.
Use ALTER SECURITY to define system-wide security options.

**Synonym:** ALT SEC

### Parameter descriptions

The parameters you specify override the current parameter values. Attributes that you do not specify are unchanged.

**Note:** If you do not specify any parameters, the command completes successfully, but no security options are changed.

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

' ' The command is executed on the queue manager on which it was entered. This is the default value.

$qmgr-name$

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the
queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

**INTERVAL**(*integer*)

The interval between checks for user IDs and their associated resources to determine whether the TIMEOUT has expired. The value is in minutes, in the range zero through 10080 (one week). If INTERVAL is specified as zero, no user timeouts occur.

**TIMEOUT**(*integer*)

How long security information about an unused user ID and associated resources is retained by WebSphere MQ. The value specifies a number of minutes in the range zero through 10080 (one week). If TIMEOUT is specified as zero, and INTERVAL is nonzero, all such information is discarded by the queue manager every INTERVAL number of minutes.

The length of time that an unused user ID and associated resources is retained by WebSphere MQ depends on the value of INTERVAL. The user ID times out at a time between TIMEOUT and TIMEOUT plus INTERVAL.

When the TIMEOUT and INTERVAL parameters are changed, the previous timer request is canceled and a new timer request is scheduled immediately, using the new TIMEOUT value. When the timer request is actioned, a new value for INTERVAL is set.
Use ALTER STGCLASS to alter the characteristics of a storage class.

**Synonym:** ALT STC

**Parameter descriptions**

For a description of the parameters see "DEFINE STGCLASS" on page 151
Use ALTER TRACE to change the trace events (IFCID) being traced for a particular active trace. ALTER TRACE stops the specified trace, and restarts it with the altered parameters.

**Note:** ALTER TRACE does not affect any RMID(231) settings (although a subsequent DISPLAY TRACE command will show them altered).

**Synonym:** ALT TRACE

**ALTER TRACE**

```
ALTER TRACE [GLOBAL | STAT | ACCTG] [TNO(integer)] [CMDSCOPE(' ')]
```

**Trace attrs:**

```
CLASS(integer) [COMMENT(string)] [IFCID(ifcid)]
```

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group.

**Parameter descriptions**

The trace type you specify determines which IFCIDs are activated. For further descriptions of each trace type, see "START TRACE" on page 319.

Specify one of the following:

- **GLOBAL**  Service data from the entire queue manager (the synonym is G)
- **STAT**    Statistical data (the synonym is S)
- **ACCTG**   Accounting data (the synonym is A)

And:

- **TNO(integer)**  The number of the trace to be altered. This limits the list to a particular trace, identified by its trace number (1 through 32). You can specify only one trace number.
ALTER TRACE

CMDSCOPE

This parameter specifies how the command is executed when the queue
manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first
initialization input data set CSQINP1.

The command is executed on the queue manager on which it was
entered. This is the default value.

qmgr-name

The command is executed on the queue manager you specify,
providing the queue manager is active within the queue-sharing
group.

You can specify a queue manager name, other than the queue
manager on which the command was entered, only if you are
using a queue-sharing group environment and if the command
server is enabled.

Trace parameters

CLASS(integer)

The trace class to be altered. This limits the list to IFCIDs activated for
particular classes. See “START TRACE” on page 319 for a list of allowed
classes. A range of classes can be specified as m:n (for example,
CLASS(01:03)). CLASS(*) activates all default IFCID classes.

COMMENT(string)

A comment that is reproduced in the trace output record (except in the
resident trace tables).

string is any character string. If it includes blanks, commas, or special
characters, it must be enclosed between single quotation marks (’).

IFCID(ifcid)

The events to be traced. This specifies the optional IFCIDs to activate. All
IFCID{s and classes specified are activated for the trace type specified.
Use ARCHIVE LOG as part of your backup procedure. It takes a copy of the current active log (or both logs if you are using dual logging).

In detail, it does the following:
1. Truncates the current active log data sets.
2. Continues logging, switching to the next active log data set.
3. Starts a task to off-load the data sets.
4. Archives previous active log data sets not yet archived.

If the MODE(QUIESCE) parameter is used, the ARCHIVE LOG command quiesces (suspends) all user update activity on the current active log before the off-load process. Once a system-wide point of consistency is reached (that is, when all currently active update users have reached a commit point), the current active log data set is immediately truncated, and the off-load process is initiated. The resulting point of consistency is captured in the current active log before it is off-loaded.

Normally, control returns to the user immediately, and the quiescing is done asynchronously. However, if the WAIT(YES) parameter is used, the quiescing is done synchronously, and control does not return to the user until it has finished.

**Synonym:** ARC LOG

ARCHIVE LOG

**Parameter descriptions**

All the parameters are optional. If none are specified, the current active log data sets are switched and off-loaded immediately.

**CANCEL OFFLOAD**

Cancels any off-loading currently in progress and restarts the off-load process. The process starts with the oldest active log data set and proceeds through all the active data sets that need off-loading.

---

<table>
<thead>
<tr>
<th>Compaq NSK</th>
<th>Compaq OpenVMS</th>
<th>OS/400</th>
<th>OS/2 Warp</th>
<th>UNIX systems</th>
<th>Windows</th>
<th>z/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Notes:
1. Valid only when the queue manager is a member of a queue-sharing group.
ARCHIVE LOG

Use this command only if the off-load task does not appear to be working, or if you want to restart a previous off-load attempt that failed.

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

The command is executed on the queue manager on which it was entered. This is the default value.

**qmgr-name**

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

**MODE(QUIESCE)**

Stops any new update activity on the queue manager, and brings all existing users to a point of consistency after a commit. When this state is reached, or the number of active users is zero, the current active log is archived.

The time that the queue manager waits to reach such a state is limited to the value specified by QUIESCE in the CSQ6ARVP system parameter macro. The value of QUIESCE can be overridden by the TIME parameter of this command. If activity has not quiesced in that time, the command fails; no off-load is done, and logging continues with the current active log data set.

**TIME(nn)**

Overrides the quiesce time period specified by the QUIESCE value of the CSQ6ARVP system parameter macro.

*nn* is the time, in seconds, in the range 001 through 999.

To specify the TIME parameter, you must also specify MODE(QUIESCE).

If you specify the TIME parameter, you must specify an appropriate value for the quiesce period. If you make the period too short or too long, one of the following problems might occur:

- The quiesce might not be complete
- WebSphere MQ lock contention might develop
- A timeout might interrupt the quiesce

**WAIT**

Specifies whether WebSphere MQ is to wait until the quiesce process has finished before returning to the issuer of the ARCHIVE LOG command.

To specify the WAIT parameter, you must also specify MODE(QUIESCE).

**NO**

Specifies that control is returned to the issuer when the quiesce process starts. (The synonym is N.) This makes the quiesce process asynchronous to the issuer; you can issue further MQSC commands when the ARCHIVE LOG command returns control to you. This is the default.
YES Specifies that control is returned to the issuer when the quiesce process finishes. (The synonym is Y.) This makes the quiesce process synchronous to the issuer; further MQSC commands are not processed until the ARCHIVE LOG command finishes.

Usage notes

1. You cannot issue an ARCHIVE LOG command while a previous ARCHIVE LOG command is in progress.
2. You cannot issue an ARCHIVE LOG command when the active log data set is the last available active log data set, because it would use all the available active log data set space, and WebSphere MQ would halt all processing until an off-load had been completed.
3. You can issue an ARCHIVE LOG command without the MODE(QUIESCE) option when a STOP QMGR MODE(QUIESCE) is in progress, but not when a STOP QMGR MODE (FORCE) is in progress.
4. You can issue a DISPLAY THREAD command to discover whether an ARCHIVE LOG command is active. If an ARCHIVE LOG command is active, the DISPLAY command returns message CSQV400I.
5. You can issue an ARCHIVE LOG command even if archiving is not being used (that is, OFFLOAD is set to NO in the CSQ6LOGP system parameter macro), or dynamically using the SET LOG command. In this case, the current active log data sets are truncated and logging continues using the next active log data set, but there is no off-loading to archive data sets.
Use BACKUP CFSTRUCT to initiate a CF application structure backup.

*Synonym:* None

**BACKUP CFSTRUCT**

```
BACKUP CFSTRUCT(structure-name) CMDSCOPE('') EXCLINT(integer)
```

**Notes:**
1. Valid only when the queue manager is a member of a queue-sharing group.
   
   You can use queue-sharing groups only on WebSphere MQ for z/OS.

**Keyword and parameter descriptions**

*structure-name*

The name of the Coupling Facility application structure to be backed up.

The name:
- Cannot have more than 12 characters.
- Must start with an uppercase letter (A through Z).
- Can include only the characters A through Z and 0 through 9.

The name of the queue-sharing group to which the queue manager is connected is prefixed to the name you supply. The name of the queue-sharing group is always four characters, padded with @ symbols if necessary. For example, if you use a queue-sharing group named NY03, and you supply the name PRODUCT7, the resultant Coupling Facility structure name is NY03PRODUCT7.

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

```
''
```

The command is executed on the queue manager on which it was entered. This is the default value.

**qmgr-name**

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and the command server is enabled.
EXCLINT(integer)

Specifies a value in seconds that defines the length of time immediately before the current time where the backup starts. The backup excludes backing-up the last \( n \) seconds activity. For example, if EXCLINT(30) is specified, the backup does not include the last 30 seconds worth of activity for this application-structure.

The value must be in the range 30 through 600. The default value is 30.

**Usage notes**

1. You can concurrently run separate backups for different application structures on different queue managers within the queue-sharing group. You can also concurrently run separate backups for different application structures on the same queue manager.

2. This command fails if the specified CF structure is defined with either a CFLEVEL less than 3, or with RECOVER set to NO.

3. The command fails if a specified application structure is currently in the process of being backed up by another queue manager within the queue-sharing group.

4. The BACKUP CFSTRUCT command cannot be run during queue manager startup, that is, from the initialization input data sets.
Use CLEAR QLOCAL to clear the messages from a local queue.

**Synonym:** CLEAR QL

### CLEAR QLOCAL

**Parameter descriptions**

You must specify which local queue you want to clear.

The command fails if either:

- The queue has uncommitted messages that have been put on the queue under syncpoint
- The queue is currently open by an application (with any open options)

If an application has this queue open, or has a queue open that eventually resolves to this queue, the command fails. The command also fails if this queue is a transmission queue, and any queue that is, or resolves to, a remote queue that references this transmission queue, is open.

**q-name**

The name of the local queue to be cleared. The name must be defined to the local queue manager.

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to SHARED.

- '' The command is executed on the queue manager on which it was entered. This is the default value.

**qmgr-name**

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

Notes:

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.
CLEAR QLOCAL

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

QSGDISP
Specifies whether or not the queue definition is shared. This parameter applies to z/OS only.

PRIVATE
Clear only the private queue named q-name. The queue is private if it was defined using a command that had the parameters QSGDISP(COPY) or QSGDISP(QMGR). This is the default value.

SHARED
Clear only the shared queue named q-name. The queue is shared if it was defined using a command that had the parameters QSGDISP(SHARED).

Usage notes
On Compaq NSK, the command cannot detect when uncommitted messages are being backed out from a queue, so you should verify that the queue files are not open before running the command.
Use DEFINE AUTHINFO to define an authentication information object. These objects contain the definitions required to perform Certificate Revocation List (CRL) checking using LDAP servers, except on OS/400 where these are defined by the Digital Certificate Manager for each Certification Authority.

**Synonym:** DEF AUTHINFO

**Parameter descriptions**

The parameter descriptions also apply to the ALTER AUTHINFO command, with the following exceptions:

- The `AUTHTYPE` parameter applies only to the DEFINE AUTHINFO command.
- The `LIKE` parameter applies only to the DEFINE AUTHINFO command.
- The `REPLACE` and `NOREPLACE` parameter applies only to the DEFINE AUTHINFO command.

- `name` Name of the authentication information object. This is required.
DEFINE AUTHINFO

The name must not be the same as any other authentication information object name currently defined on this queue manager (unless REPLACE or ALTER is specified). See "Rules for naming WebSphere MQ objects" on page 4.

**AUTHTYPE**

The type of authentication information. The value must be CRLLDAP, meaning that Certificate Revocation List checking is done using LDAP servers.

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group. CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP.

''

The command is executed on the queue manager on which it was entered. This is the default value.

```
qmgr-name
```

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name other than the queue manager on which it was entered, only if you are using a shared queue environment and if the command server is enabled.

```
*
```

The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

**CONNAME(string)**

The DNS name or IP address of the host on which the LDAP server is running, with an optional port number. This keyword is required.

The syntax for CONNAME is the same as for channels. For example,

```
conname('hostname(nn)')
```

where `nn` is the port number. If `nn` is not provided, the default port number 389 is used.

The maximum length for the field is 264 characters on OS/400, UNIX systems, and Windows, and 48 characters on z/OS.

**DESCR(string)**

Plain-text comment. It provides descriptive information about the authentication information object when an operator issues the DISPLAY AUTHINFO command (see "DISPLAY AUTHINFO" on page 174).

It should contain only displayable characters. The maximum length is 64 characters. In a DBCS installation, it can contain DBCS characters (subject to a maximum length of 64 bytes).

**Note:** If characters are used that are not in the coded character set identifier (CCSID) for this queue manager, they might be translated incorrectly if the information is sent to another queue manager.
DEFINE AUTHINFO

LDAPPWD(string)

The password associated with the Distinguished Name of the user who is accessing the LDAP server.

Its maximum size is 32 characters. The default value is blank.

LDAPUSER(string)

The Distinguished Name of the user who is accessing the LDAP server.

(See the SSLPEER parameter on page 108 for more information about distinguished names.)

The maximum size for the user name is 1024 characters on OS/400, UNIX systems, and Windows, and 256 characters on z/OS.

On OS/400, UNIX systems, and Windows, the maximum accepted line length is defined to be BUFSIZ, which can be found in stdio.h.

If you use asterisks (*) in the user name they are treated as literal characters, and not as wild cards, because LDAPUSER is a specific name and not a string used for matching.

LIKE(authinfo-name)

The name of an authentication information object, whose parameters will be used to model this definition.

If this field is not filled in, and you do not complete the parameter fields related to the command, the values are taken from the default definition for an object of this type. This is equivalent to specifying LIKE(SYSTEM.DEFAULT.AUTHINFO.CRLLDAP).

A default authentication information object definition is provided, but it can be altered by the installation to the default values required. See “Rules for naming WebSphere MQ objects” on page 4.

On WebSphere MQ for z/OS, the queue manager searches page set zero for an object with the name you specify. The disposition of the LIKE object is not copied to the object you are defining.

QSGDISP

This parameter applies to z/OS only.

Specifies the disposition of the object to which you are applying the command (that is, where it is defined and how it behaves).

<table>
<thead>
<tr>
<th>QSGDISP</th>
<th>ALTER</th>
<th>DEFINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY</td>
<td>The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(COPY). Any object residing in the shared repository, or any object defined using a command that had the parameters QSGDISP(QMGR), is not affected by this command.</td>
<td>The object is defined on the page set of the queue manager that executes the command using the QSGDISP(GROUP) object of the same name as the 'LIKE' object.</td>
</tr>
</tbody>
</table>
QSGDISP ALTER

GROUP

The object definition resides in the shared repository. The object was defined using a command that had the parameters QSGDISP(GROUP). Any object residing on the page set of the queue manager that executes the command (except a local copy of the object) is not affected by this command. If the command is successful, the following command is generated and sent to all active queue managers in the queue-sharing group so that they refresh local copies on page set zero:

```
DEFINE AUTHINFO(name)
REPLACE QSGDISP(COPY)
```

PRIVATE

The object resides on the page set of the queue manager that executes the command, and was defined with QSGDISP(QMGR) or QSGDISP(COPY). Any object residing in the shared repository is unaffected.

QMGR

The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(QMGR). Any object residing in the shared repository, or any local copy of such an object, is not affected by this command. This is the default value.

The object is defined on the page set of the queue manager that executes the command. This is the default value.

REPLACE and NOREPLACE

Whether the existing definition (and on z/OS, with the same disposition) is to be replaced with this one. This is optional. The default is NOREPLACE. Any object with a different disposition is not changed.

REPLACE

The definition should replace any existing definition of the same name. If a definition does not exist, one is created.

NOREPLACE

The definition should not replace any existing definition of the same name.

Usage Notes

1. On OS/400, authentication information objects are only used for channels of type CLNTCONN through use of the AMQCLCHLTAB. Certificates are defined by Digital Certificate Manager for each Certification Authority, and are verified against the LDAP servers.
USE DEFINE BUFFPOOL TO DEFINE A BUFFER POOL THAT IS USED FOR HOLDING MESSAGES IN MAIN STORAGE.

NOTES:
1. DEFINE BUFFPOOL CAN BE ISSUED ONLY FROM THE CSQINP1 INITIALIZATION DATA SET.
2. YOU CAN USE THE DISPLAY USAGE TYPE(PAGESET) COMMAND TO DISPLAY BUFFER POOL INFORMATION, (SEE “DISPLAY USAGE” ON PAGE 261).

SYNONYM: DEF BP

DEFINE BUFFPOOL

Parameter descriptions
If this command is not issued, the default number of buffers is assumed. If more than one DEFINE BUFFPOOL command is issued for the same buffer pool, only the last one is actioned.

(buf-pool-id)
Buffer pool identifier. This is required.

This is an integer in the range zero through 15.

BUFFERS(integer)
The number of 4096-byte buffers to be used in this buffer pool. This is optional. The default number of buffers is 1000, and the minimum is 100. The maximum number of buffers for all the buffer pools is determined by the amount of storage available in the WebSphere MQ address space.

See the WebSphere MQ for z/OS Concepts and Planning Guide for guidance on the number of buffers you can define in each buffer pool.
Use DEFINE CFSTRUCT to define the backup and recovery parameters for a Coupling Facility application structure.

**Synonym:** DEF CFSTRUCT

**DEFINE CFSTRUCT**

```
DEFINE CFSTRUCT(structure-name)
  CFLEVEL(integer)
  DSCR(string)
  LIKE(cfstruct-name)
  RECOVER(YES|NO)
  NOREPLACE
  REPLACE
```

**Keyword and parameter descriptions**

(structure-name)

Name of the Coupling Facility application structure whose backup and recovery parameters you want to define. This is required.

The name:

- Cannot have more than 12 characters.
- Must start with an uppercase letter (A through Z).
- Can include only the characters A through Z and 0 through 9.

The name of the queue-sharing group to which the queue manager is connected is prefixed to the name you supply. The name of the queue-sharing group is always four characters, padded with @ symbols if necessary. For example, if you use a queue-sharing group named NY03 and you supply the name PRODUCT7, the resultant Coupling Facility structure name is NY03PR00UCT7. Note that the administrative structure for the queue-sharing group (in this case NY03CSQ_ADMIN) cannot be used for storing messages.

**CFLEVEL(integer)**

Specifies the functional capability level for this CF application structure. Value can be one of the following:

1. A CF structure that can be "auto-created" by a queue manager at command level 520.
2. A CF structure at command level 520 that can only be created or deleted by a queue manager at command level 530. This is the default CFLEVEL for queue managers at command level 530.
3. A CF structure at command level 530. This CFLEVEL is required if you want to use persistent messages on shared queues (if RECOVER(YES) is set), or for message grouping (when a local queue is defined with INDXTYPE(GROUPID)), or both.
DEFINE CFSTRUCT

You can only increase the value of CFLEVEL to 3 if all the queue managers in the queue-sharing group are at command level 530 - this is to ensure that there are no latent command level 520 connections to queues referencing the CFSTRUCT.

You can only decrease the value of CFLEVEL from 3 if all the queues that reference the CF structure are both empty (have no messages or uncommitted activity) and closed.

DESCR(string)
Plain-text comment that provides descriptive information about the object when an operator issues the DISPLAY CFSTRUCT command.

The string should contain only displayable characters. The maximum length is 64 characters. In a DBCS installation, it can contain DBCS characters (subject to a maximum length of 64 bytes).

Note: If characters are used that are not in the coded character set identifier (CCSID) for this queue manager, they might be translated incorrectly if the information is sent to another queue manager.

LIKE(cfstruct-name)
The name of a CFSTRUCT object, whose attributes are used to model this definition. If this field is not filled in, and you do not complete the attribute fields related to the command, the default values are used.

RECOVER
Specifies whether CF recovery is supported for the application structure. Values are:

NO CF application structure recovery is not supported. (The synonym is N.)

YES CF application structure recovery is supported. (The synonym is Y.)

You can only set RECOVER(YES) if the structure has a CFLEVEL of 3.

You can only change RECOVER(NO) to RECOVER(YES) if all the queue managers in the queue-sharing group are at command level 530; this is to ensure that there are no latent command level 520 connections to queues referencing the CFSTRUCT.

You can only change RECOVER(YES) to RECOVER(NO) if all the queues that reference the CF structure are both empty (have no messages or uncommitted activity) and closed.

REPLACE and NOREPLACE
Defines whether the existing definition is to be replaced with this one. This parameter is optional.

REPLACE
The definition should replace any existing definition of the same name. If a definition does not exist, one is created. If you use the REPLACE option, all queues that use this CF structure must be empty and closed. Any CF structure attributes that are not specified on a DEFINE CFSTRUCT call with REPLACE are reset to their default values.
The definition should not replace any existing definition of the same name. This is the default.

Usage notes

1. This command cannot be run from the CSQINP1 initialization input data set.
2. This command cannot specify the CF administration structure (CSQ_ADMIN).
3. Before any newly defined CF structure can be used by any queues, the structure must be defined in the Coupling Facility Resource Management (CFRM) policy data set.
4. Only CF structures with RECOVER(YES) defined can be backed up and recovered.
Use DEFINE CHANNEL to define a new channel, and set its parameters.

**Notes:**

1. On z/OS, this is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, see the [WebSphere MQ Intercommunication](https://www.ibm.com/support/knowledgecenter/en/SSD7JU_8.6.1/com.ibm.mq.doc/library/chapter24.html) manual.

2. For cluster-sender channels, you can only specify the REPLACE option for channels that have been created manually.

**Synonym:** DEF CHL

There is a separate syntax diagram for each type of channel:

- [“Sender channel” on page 71](#)
- [“Server channel” on page 74](#)
- [“Receiver channel” on page 77](#)
- [“Requester channel” on page 79](#)
- [“Client-connection channel” on page 82](#)
- [“Server-connection channel” on page 84](#)
- [“Cluster-sender channel” on page 86](#)
- [“Cluster-receiver channel” on page 88](#)
Sender channel

**DEFINE CHANNEL**

```
DEFINE CHANNEL(channel-name) CHLTYPE(SDR) CONNAME(string)
```

```
= TRPTYPE (DECNET LU62 NETBIOS SPX TCP UDP)
```

```
= BATCHINT(integer) BATCHSZ(integer) CMDSCOPE(qmgr-name) CMDSCOPE(*)
```

```
= CONVERT(NO) CONVERT(YES)
```

```
= KAINT(AUTO) KAINT(integer) LIKE(channel-name)
```

```
= LONGRTY(integer) LONGTMR(integer)
```

```
= MAXMSGL(integer) MCAUSER(string) MODENAME('string') MSGDATA('string') MSGEXIT('string') NPMSPEED(FAST)
```

```
= DISCINT(integer) HBINT(integer)
```

```
= LOCLADDR('string')
```

```
= CMDSCOPE('*')
```

```
= MCATYPE(PROCESS) MCATYPE(THREAD)
```

```
= MSGDATA(string)
```

```
= MAXMSQL(4 194 304)
```

```
= MCTOSCANAME('')
```

```
= NPMSPEED(NORMAL)
```

Chapter 2. The MQSC commands
Notes:

1. This parameter must follow immediately after the channel name except on z/OS.
2. This is not mandatory on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
3. Valid only on Compaq OpenVMS.
4. Valid only on OS/2 Warp and Windows.
5. Valid only on AIX.
6. This is the default supplied with WebSphere MQ, but your installation might have changed it.
7. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
8. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
9. Valid only on z/OS.
10 Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.

11 Valid only if TRPTYPE is LU62.

12 You can specify more than one value only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, z/OS, Solaris, and Windows.

13 Not valid on z/OS.

14 Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
DEFINE CHANNEL

Server channel

DEFINE CHANNEL

--- DEFINE CHANNEL(channel-name) CHLTYPE(SVR) TRPTYPE(DECNET) 
--- XMITQ(string) AUTOSTART(DISABLED) BATCHSZ(integer) CMDSCOPE('') 
--- CONVERT(NO) DESCR(string) DISCINT(integer) HBINT(integer) 
--- KAINT(AUTO) LOCLADDR('') LONGRTY(integer) 
--- MAXMSGL(integer) MCAUSER('') MODENAME(string) 
--- MSGDATA(string) MSGEXIT(string)
DEFINE CHANNEL

Notes:

1. This parameter must follow immediately after the channel name except on z/OS.
2. This is not mandatory on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
3. Valid only on Compaq OpenVMS.
4. Valid only on OS/2 Warp and Windows.
5. Valid only on AIX.
6. Valid only on Compaq NSK.
7. This is the default supplied with WebSphere MQ, but your installation might have changed it.
8. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
DEFINE CHANNEL

9 Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.

10 Valid only on z/OS.

11 Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.

12 Valid only if TRPTYPE is LU62.

13 You can specify more than one value only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

14 Not valid on z/OS.

15 Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
Receiver channel

DEFINE CHANNEL

```
DEFINE CHANNEL(channel-name) CHLTYPE(RCVR) TRPTYPE(DECNET) AUTOSTART(DISABLED)
BATCHSZ(50) HINT(300) KAINT(AUTO) LIKE(channel-name)
MAXMSGL(integer) MCAUSER(string) MRTMR(integer)
NPMSPEED(Fast) QSGDISP(QMGR)
PUTAUT(DEF) RCVDATA(string) RCVEXIT(string) NOREPLACE
```

Chapter 2. The MQSC commands 77
DEFINE CHANNEL

Notes:

1. This parameter must follow immediately after the channel name except on z/OS.
2. This is not mandatory on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
3. Valid only on Compaq OpenVMS.
4. Valid only on OS/2 Warp or Windows.
5. Valid only on AIX.
6. Valid only on Compaq NSK.
7. This is the default supplied with WebSphere MQ, but your installation might have changed it.
8. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
9. Valid only on z/OS.
10. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
11. Not valid on z/OS.
12. You can specify more than one value only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
13. Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
Requester channel

```
DEFINE CHANNEL (channel-name) CHLTYPE(RQSTR) CONNAME(string)

TRPTYPE (DECNET)
LU62
NETBIOS
SPX
TCP
UDP

CMOSCOPE('')
CMOSCOPE(qmgr-name)
CMOSCOPE(*)

KAINT(AUTO)
KAINT(integer)
LIKE(channel-name)
LOCLADDR('')

MAXMSG(4 194 304)
MCATYPE(PROCESS)
MCATYPE(THREAD)

MCAUSER('')
MODENAME('')
MRDATA('')
MREXIT('')
MRRTY(10)
MRTMR(1000)

MSGDATA('')
MSGEXIT('')

NPMSPEED(FAST)
NPMSPEED(NORMAL)
PASSWORD('')
PUTAUT(DEF)
```

Chapter 2. The MQSC commands 79
Notes:

1. This parameter must follow immediately after the channel name except on z/OS.

2. This is not mandatory on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

3. Valid only on Compaq OpenVMS.

4. Valid only on OS/2 Warp and Windows.

5. Valid only on AIX.

6. Valid only on Compaq NSK.

7. This is the default supplied with WebSphere MQ, but your installation might have changed it.

8. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.

9. Valid only on z/OS.

10. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

11. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
12 Valid only if TRPTYPE is LU62.
13 Not valid on z/OS.
14 You can specify more than one value only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
15 Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
DEFINE CHANNEL

Client-connection channel

DEFINE CHANNEL

- DEFINE CHANNEL(channel-name)—CHLTYPE—(CLNTCONN)—CONNAME(string)
- TRPTYPE
  - DECNET
    - LU62
    - NETBIOS
    - SPX
    - TCP
- CMDSCOPE("")
- CMDSCOPE(qmgr-name)
- CMDSCOPE(*)
- DESCR("")
- DESCR(string)
- HBINT(300)
- HBINT(integer)
- KAINT(AUTO)
- KAINT(integer)
- LOCLADDR("")
- LOCLADDR(string)
- MAXMSG(4 194 304)
- MAXMSG(integer)
- MODENAME("")
- MODENAME(string)
- PASSWORD("")
- PASSWORD(string)
- QMNAME("")
- QMNAME(string)
- QSGDISP(QMGR)
- QSGDISP(COPY)
- QSGDISP(GROUP)
- RCVDATA("")
- RCVDATA(string)
- RCVDATA(string)
- RCVEXIT("")
- RCVEXIT(string)
- NOREPLACE
- NOREPLACE
- REPLACEREECE
- REPLACEREECE
- SCYDATA("")
- SCYDATA(string)
- SCYEXIT("")
- SCYEXIT(string)
- SEDDDATA("")
- SEDDDATA(string)
**DEFINE CHANNEL**

Notes:

1. This parameter must follow immediately after the channel name except on z/OS.

2. This is not mandatory on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

3. Valid only on Compaq OpenVMS.

4. Valid only for clients to be run on DOS, OS/2 Warp, or Windows.

5. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.

6. Valid only on z/OS.

7. This is the default supplied with WebSphere MQ, but your installation might have changed it.

8. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.

9. Valid only if TRPTYPE is LU62.

10. You can specify more than one value only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

11. Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
DEFINE CHANNEL

Server-connection channel

DEFINE CHANNEL(channel-name) (1) CHLTYPE(SVRCONN) (2) TRPTYPE(DECNET) (3)

AUTOSTART(DISABLED) (5) CMDSCOPE(' ') (7) DESCRIPTOR(' ') (8)

HBINT(300) (9) KAIN(AUTO) (7) LIKE(channel-name)

MAXMSGL(4 194 304) (8) MCAUSER(' ') (8) PUTAUTO(DEF) (7)

QSGDISP(QMGR) (7) RCVDATA(' ') (8)

RCVEXIT(' ') (8) NOREPLACE (8) SCYDATA(' ') (8)

SENDDATA(' ') (8) SENDEXIT(' ') (8)
Notes:

1. This parameter must follow immediately after the channel name except on z/OS.
2. This is not mandatory on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
3. Valid only on Compaq OpenVMS.
4. Valid only for clients to be run on DOS, OS/2 Warp, or Windows.
5. Valid only on Compaq NSK.
6. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
7. Valid only on z/OS.
8. This is the default supplied with WebSphere MQ, but your installation might have changed it.
9. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
10. You can specify more than one value only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
11. Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
DEFINE CHANNEL

Notes:

1. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
2. This parameter must follow immediately after the channel name except on z/OS.
3. This is the default supplied with WebSphere MQ, but your installation might have changed it.
4. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
5. Valid only on z/OS.
6. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
7. Valid only if TRPTYPE is LU62.
8. Not valid on z/OS.
9. Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
10. Valid only on OS/2 Warp and Windows.
11. Valid only on AIX.
DEFINE CHANNEL

Cluster-receiver channel

DEFINE CHANNEL(channel-name) CHLTYPE(CLUSRCVR) CONNAME(string)

BATCHHB(integer) BATCHINT(integer) BATCHSZ(integer) CLUSNL(nlname)

CLUSTER(cluster-name) CONV(YES) CMDSCOPE(qmgr-name)

DECR(string) DISCINT(integer) HBINT(integer) KAINT(AUTO)

LIKE(channel-name) LOCLADDR(string) LONGRTY(integer)

LONGTMR(integer) MAXMSGL(integer) MCATYPE(THREAD)

MODENAME(string) MRDATA(string) MREXIT(string)

MRRTY(integer) MRTMR(integer) MSGDATA(string)

MSGEXIT(string) NETPRY(integer) NPMSPEED(FS)

WebSphere MQ Script (MQSC) Command Reference
DEFINE CHANNEL

Notes:

1 Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
2 This parameter must follow immediately after the channel name except on z/OS.
3 This parameter is optional if TRPTYPE is TCP.
4 This is the default supplied with WebSphere MQ, but your installation might have changed it.
5 Valid only when the queue manager is a member of a queue-sharing group.
   You can use queue-sharing groups only on WebSphere MQ for z/OS.
6 Valid only on z/OS.
7 Valid only if TRPTYPE is LU62.
8 Not valid on z/OS.
**DEFINE CHANNEL**

9  Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
10 Valid only on OS/2 Warp and Windows.
11 Valid only on AIX.

**Parameter descriptions**

The parameter descriptions also apply to the ALTER CHANNEL command, with the following exceptions:

- The **LIKE** parameter applies only to the DEFINE CHANNEL command.
- The **REPLACE** and **NOREPLACE** parameter applies only to the DEFINE CHANNEL command.
- The variations in the **CMDSCOPE** and **QSGDISP** parameters between the ALTER CHANNEL and DEFINE CHANNEL commands are described.

Parameters are optional unless the description states that they are required.

**Parameter descriptions**

- **(channel-name)**
  - The name of the new channel definition.
  
  This parameter is required on all types of channel. On CLUSSDR channels it can take a different form to the other channel types. If your convention for naming cluster-sender channels includes the name of the queue manager, you can define a cluster-sender channel using the `+QMNAME+` construction. After connection to the matching cluster-receiver channel, WebSphere MQ substitutes the correct repository queue manager name in place of `+QMNAME+` in the cluster-sender channel definition. This facility applies to AIX, HP-UX, Linux, OS/400, Solaris, and Windows only. For a full explanation of this facility, see [WebSphere MQ Queue Manager Clusters](#).

  The name must not be the same as any existing channel defined on this queue manager (unless REPLACE or ALTER is specified). On z/OS, client-connection channel names can duplicate others.

  The maximum length of the string is 20 characters, and the string must contain only valid characters; see “Rules for naming WebSphere MQ objects” on page 4.

- **AUTOSTART**
  - Specifies whether an LU 6.2 responder process for the channel will be started at queue manager startup.

  **ENABLED**
  - The responder is started.

  **DISABLED**
  - The responder is not started (this is the default).

  This parameter is valid only for channels with a channel type (CHLTYPE) of RCVR, RQSTR, SVR, and SVRCONN. It is supported only on Compaq NSK.

- **BATCHHB**(integer)
  - Specifies whether batch heartbeats are to be used. The value is the length of the heartbeat in milliseconds.

  Batch heartbeats allow a sending channel to verify that the receiving channel is still active just before committing a batch of messages, so that if the receiving channel is not active, the batch can be backed out rather than becoming in-doubt, as would otherwise be the case. By backing out the
DEFINE CHANNEL

- Batch, the messages remain available for processing so they could, for example, be redirected to another channel.

  If the sending channel has had a communication from the receiving channel within the batch heartbeat interval, the receiving channel is assumed to be still active, otherwise a 'heartbeat' is sent to the receiving channel to check.

  The value must be in the range zero through 999 999. A value of zero indicates that batch heartbeating is not used.

  This parameter is valid only for channels with a channel type (CHLTYPE) of SDR, SVR, CLUSSDR and CLUSRCVR.

  **BATCHINT(integer)**
  
  The minimum amount of time, in milliseconds, that a channel keeps a batch open.

  The batch is terminated by whichever of the following occurs first:
  
  - BATCHSZ messages have been sent, or
  - The transmission queue is empty and BATCHINT is exceeded

  The default value is zero, which means that the batch is terminated as soon as the transmission queue becomes empty (or the BATCHSZ limit is reached).

  The value must be in the range zero, through 999 999 999.

  This parameter is valid only for channels with a channel type (CHLTYPE) of SDR, SVR, CLUSSDR, or CLUSRCVR. It is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

  **BATCHSZ(integer)**
  
  The maximum number of messages that can be sent through a channel before taking a checkpoint.

  The maximum batch size actually used is the lowest of the following:
  
  - The BATCHSZ of the sending channel
  - The BATCHSZ of the receiving channel
  - Three less than the maximum number of uncommitted messages allowed at the sending queue manager (or one if this value is zero or less)
  - Three less than the maximum number of uncommitted messages allowed at the receiving queue manager (or one if this value is zero or less)

  The maximum number of uncommitted messages is specified by the MAXUMSGS parameter of the ALTER QMGR command.

  This parameter is valid only for channels with a channel type (CHLTYPE) of SDR, SVR, RCVR, RQSTR, CLUSSDR, or CLUSRCVR.

  The value must be in the range 1 through 9999.

  **CHLTYPE**
  
  Channel type. This is required. It must follow immediately after the (channel-name) parameter on all platforms except z/OS.

  - **SDR** Sender channel
  - **SVR** Server channel
DEFINE CHANNEL

RCVR Receiver channel
RQSTR Requester channel
CLNTCONN Client-connection channel
SVRCONN Server-connection channel
CLUSSDR Cluster-sender channel (valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS)
CLUSRCVR Cluster-receiver channel (valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS)

Note: If you are using the REPLACE option, you cannot change the channel type.

CLUSNL
The name of the namelist that specifies a list of clusters to which the channel belongs.

This parameter is valid only for channels with a channel type (CHLTYPE) of CLUSSDR and CLUSRCVR channels. Only one of the resultant values of CLUSTER or CLUSNL can be nonblank, the other must be blank.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

CLUSTER
The name of the cluster to which the channel belongs. The maximum length is 48 characters conforming to the rules for naming WebSphere MQ objects.

This parameter is valid only for channels with a channel type (CHLTYPE) of CLUSSDR and CLUSRCVR channels. Only one of the resultant values of CLUSTER or CLUSNL can be nonblank, the other must be blank.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows and z/OS.

CMDSCOPE
This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSG_DISP is set to GROUP.

```
' ' The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name other than the queue manager on which it was entered, only if you are using a shared queue environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

92 WebSphere MQ Script (MQSC) Command Reference
**CONNAME**(string)

Connection name.

For cluster-receiver channels (when specified) **CONNAME** relates to the local queue manager, and for other channels it relates to the target queue manager.

The maximum length of the string is 48 characters on z/OS, and 264 characters on other platforms.

This parameter is required for channels with a channel type (CHLTYPE) of SDR, RQSTR, CLNTCONN, and CLUSSDR. It is optional for SVR channels, and for CLUSRCVR channels of TRPTYPE(TCP), and is not valid for RCVR or SVRCONN channels.

**Note:** If you are using any of the special characters in your connection name (for example, parentheses) you must enclose the string in single quotes.

The value you specify depends on the transport type (TRPTYPE) to be used:

**DECnet**

The DECnet node name and the DECnet object name, in the form: 

```
CONNAME('node_name(object_name)')
```

This is valid only on Compaq OpenVMS.

**LU 6.2**

- On Compaq OpenVMS this is the SNA gateway node name, access name, and the tpname that is used by SNA to invoke the remote program. The format of this information is as follows: 

  ```
  CONNAME('gateway_node.access_name(tpname)')
  ```

- On z/OS there are two forms in which to specify the value:

  **Logical unit name**

  The logical unit information for the queue manager, comprising the logical unit name, TP name, and optional mode name. This can be specified in one of 3 forms:

<table>
<thead>
<tr>
<th>Form</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>luname</td>
<td>IGY12355</td>
</tr>
<tr>
<td>luname/TPname</td>
<td>IGY12345/APIING</td>
</tr>
<tr>
<td>luname/TPname/modename</td>
<td>IGY12345/APIINGD/#INTER</td>
</tr>
</tbody>
</table>

  For the first form, the TP name and mode name must be specified for the TPNAME and MODENAME parameters; otherwise these parameters must be blank.

  **Note:** For client-connection channels, only the first form is allowed.

  **Symbolic name**

  The symbolic destination name for the logical unit information for the queue manager, as defined in the side information data set. The TPNAME and MODENAME parameters must be blank.
Note: For cluster-receiver channels, the side information is on the other queue managers in the cluster. Alternatively, in this case it can be a name that a channel auto-definition exit can resolve into the appropriate logical unit information for the local queue manager.

The specified or implied LU name can be that of a VTAM® generic resources group.

- On OS/2 Warp it is the fully-qualified name of the partner LU, or an LU alias.
- On Compaq OpenVMS, OS/400, UNIX systems, and Windows, this is the name of the CPI-C communications side object or, if the TPNAME is not blank, this is the fully-qualified name of the partner logical unit.

See the information about configuration parameters for an LU 6.2 connection for your platform in the WebSphere MQ Intercommunication manual for more information.

- On Compaq NSK, the value of this depends on whether SNAX or ICE is used as the communications protocol:
  - If SNAX is used:
    - For sender, requester, and fully qualified server channels, this is the process name of the SNAX/APC process, the name of the local LU, and the name of the partner LU on the remote machine, for example:
      CONNAME('$$PPP.LOCALLU.REMOTELU')
    - For receiver and non fully qualified server channels, this is the process name of the SNAX/APC process and the name of the local LU, for example:
      CONNAME('$$PPP.LOCALLU')

      The name of the local LU can be an asterisk (*), indicating any name.
  - If ICE is used:
    - For sender, requester, and fully qualified server channels, this is the process name of the ICE process, the ICE open name, the name of the local LU, and the name of the partner LU on the remote machine, for example:
      CONNAME('$$PPP.#OPEN.LOCALLU.REMOTELU')

      For receiver and non fully qualified server channels, this is the process name of the SNAX/APC process, the ICE open name, and the name of the local LU, for example:
      CONNAME('$$PPP.#OPEN.LOCALLU')

      The name of the local LU can be an asterisk (*), indicating any name.

NetBIOS
A unique NetBIOS name (limited to 16 characters).

SPX The 4-byte network address, the 6-byte node address, and the 2-byte socket number. These values must be entered in
hexadecimal, with a period separating the network and node addresses. The socket number must be enclosed in brackets, for example:

CONNAME('0a0b0c0d.804abcde23a1(5e86)')

If the socket number is omitted, the WebSphere MQ default value (X'5e86') is assumed.

TCP

Either the host name, or the network address of the remote machine (or the local machine for cluster-receiver channels). This can be followed by an optional port number, enclosed in parentheses.

On z/OS the connection name can include the IP_name of an z/OS dynamic DNS group or a network dispatcher input port. Do not include this for channels with a channel type (CHLTYPE) of CLUSSDR.

On AIX, HP-UX, Linux, OS/400, Solaris, and Windows, when you define a channel with a channel type (CHLTYPE) of CLUSRCVR that is using TCP/IP, you do not need to specify the network address of your queue manager. WebSphere MQ generates a CONNAME for you, assuming the default port and using the current IP address of the system.

UDP

Either the host name, or the network address of the remote MQSeries for Windows V2.0 machine. This can be followed by an optional port number, enclosed in parentheses.

CONVERT

Specifies whether the sending message channel agent should attempt conversion of the application message data, if the receiving message channel agent cannot perform this conversion.

NO No conversion by sender

YES Conversion by sender

On z/OS, N and Y are accepted as synonyms of NO and YES.

This parameter is valid only for channels with a channel type (CHLTYPE) of SDR, SVR, CLUSSDR, or CLUSRCVR.

DESCR(string)

Plain-text comment. It provides descriptive information about the channel when an operator issues the DISPLAY CHANNEL command.

It should contain only displayable characters. The maximum length is 64 characters. In a DBCS installation, it can contain DBCS characters (subject to a maximum length of 64 bytes).

Note: If characters are used that are not in the coded character set identifier (CCSID) for this queue manager, they might be translated incorrectly if the information is sent to another queue manager.

DISCINT(integer)

The minimum time in seconds for which the channel waits for a message to arrive on the transmission queue, after a batch ends, before terminating the channel. A value of zero causes the message channel agent to wait indefinitely.

The value must be in the range zero through 999 999.
DEFINe CHANNEL

This parameter is valid only for channels with a channel type (CHLTYPE) of SDR, SVR, CLUSSDR, or CLUSRCVR.

HBINT(integer)
This parameter has a different interpretation depending upon the channel type, as follows:

- For channels with a channel type (CHLTYPE) of SDR, SVR, RCVR, RQSTR, CLUSSDR, or CLUSRCVR, this is the time, in seconds, between heartbeat flows passed from the sending MCA when there are no messages on the transmission queue. The heartbeat exchange gives the receiving MCA the opportunity to quiesce the channel.

  This type of heartbeat is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

  Note: You should set this value to be significantly less than the value of DISCINT. WebSphere MQ checks only that it is within the permitted range however.

- For channels with a channel type (CHLTYPE) of SVRCONN or CLNTCONN, this is the time, in seconds, between heartbeat flows passed from the server MCA when that MCA has issued an MQGET with WAIT on behalf of a client application. This allows the server to handle situations where the client connection fails during an MQGET with WAIT. This type of heartbeat is valid only for AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.

  The value must be in the range zero through 999 999. A value of zero means that no heartbeat exchange takes place. The value that is used is the larger of the values specified at the sending side and the receiving side.

KAINT(integer)
Specifies the value passed to the communications stack for KeepAlive timing for this channel. The value is applicable for the TCP/IP and SPX communications protocols, though not all implementations support this parameter.

If the TCPKEEP channel initiator parameter (set by CSQ6CHIP) is NO, the value is ignored and the KeepAlive facility is not used.

(integer)

  The KeepAlive interval to be used, in seconds, in the range 1 through 99 999.

0

  The value used is that specified by the KEEPALIVEOPTIONS statement in the TCP profile configuration data set.

AUTO

  The KeepAlive interval is calculated based upon the negotiated heartbeat value as follows:

  - If the negotiated HBINT is greater than zero, KeepAlive interval is set to that value plus 60 seconds.
  - If the negotiated HBINT is zero, the value used is that specified by the KEEPALIVEOPTIONS statement in the TCP profile configuration data set.

  The default value is AUTO.

This parameter is valid for all channel types. It is ignored for channels with a TRPTYPE other than TCP or SPX.
DEFINE CHANNEL

This parameter is valid only on z/OS.

LIKE(channel-name)

The name of a channel, whose parameters are used to model this definition.

This parameter applies only to the DEFINE CHANNEL command.

If this field is not filled in, and you do not complete the parameter fields related to the command, the values are taken from one of the following, depending upon the channel type:

- SYSTEM.DEF.SENDER  Sender channel
- SYSTEM.DEF.SERVER  Server channel
- SYSTEM.DEF.RECEIVER Receiver channel
- SYSTEM.DEF.REQUESTER Requester channel
- SYSTEM.DEF.SVRCONN  Server-connection channel
- SYSTEM.DEF.CLNTCONN  Client-connection channel
- SYSTEM.DEF.CLUSSDR  Cluster-sender channel
- SYSTEM.DEF.CLUSRCVR  Cluster-receiver channel

This is equivalent to defining the following object:
LIKE(SYSTEM.DEF.SENDER)

for a sender channel, and similarly for other channel types.

These default channel definitions can be altered by the installation to the default values required.

On WebSphere MQ for z/OS, the queue manager searches page set zero for an object with the name you specify. The disposition of the LIKE object is not copied to the object you are defining.

Notes:
1. QSGDISP (GROUP) objects are not searched.
2. LIKE is ignored if QSGDISP(COPY) is specified. However, the group object defined is used as a LIKE object.

LOCLADDR(string)

Local communications address for the channel. Use this parameter if you want a channel to use a particular IP address, port, or port range for outbound communications. This might be useful in recovery scenarios where a channel is restarted on a different TCP/IP stack.

This parameter is valid only for channels with a transport type (TRPTYPE) of TCP. If TRPTYPE is not TCP, the data is ignored and no error message is issued. The value is the optional IP address and optional port or port range used for outbound TCP/IP communications. The format for this information is as follows:

LOCLADDR([ip-addr][([low-port[,high-port]])])

where ip-addr is specified in dotted decimal or alphanumeric form, and low-port and high-port are port numbers enclosed in parentheses. All are optional.
Table 2 shows how the LOCLADDR parameter can be used:

<table>
<thead>
<tr>
<th>LOCLADDR</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.20.4.98</td>
<td>Channel binds to this address locally</td>
</tr>
<tr>
<td>9.20.4.98(1000)</td>
<td>Channel binds to this address and port 1000 locally</td>
</tr>
<tr>
<td>9.20.4.98(1000,2000)</td>
<td>Channel binds to this address and uses a port in the range 1000 to 2000 locally</td>
</tr>
<tr>
<td>(1000)</td>
<td>Channel binds to port 1000 locally</td>
</tr>
<tr>
<td>(1000,2000)</td>
<td>Channel binds to port in range 1000 to 2000 locally</td>
</tr>
</tbody>
</table>

This parameter is valid only for channels with a channel type (CHLTYPE) of SDR, SVR, RQSTR, CLNTCONN, CLUSSDR, or CLUSRCVR.

**Note:** Even though this parameter is similar in form to CONNAME, it should not be confused with it. The LOCLADDR parameter specifies the characteristics of the local communications, whereas the CONNAME parameter specifies how to reach a remote queue manager.

**LONGRTY**(integer)

When a sender, server, or cluster-sender channel is attempting to connect to the remote queue manager, and the count specified by SHORTRTY has been exhausted, this specifies the maximum number of further attempts that are made to connect to the remote queue manager, at intervals specified by LONGTMR.

If this count is also exhausted without success, an error is logged to the operator, and the channel is stopped. The channel must subsequently be restarted with a command (it is not started automatically by the channel initiator).

The value must be in the range zero through 999 999 999.

This parameter is valid only for channels with a channel type (CHLTYPE) of SDR, SVR, CLUSSDR, or CLUSRCVR.

**LONGTMR**(integer)

For long retry attempts, this is the maximum number of seconds to wait before reattempting connection to the remote queue manager.

The time is approximate; zero means that another connection attempt is made as soon as possible.

The interval between retries might be extended if the channel has to wait to become active.

The value must be in the range zero through 999 999 999.

**Note:** For implementation reasons, the maximum retry interval that can be used is 999 999; values exceeding this are treated as 999 999.

This parameter is valid only for channels with a channel type (CHLTYPE) of SDR, SVR, CLUSSDR, or CLUSRCVR.

**MAXMSGL**(integer)

Specifies the maximum message length that can be transmitted on the
channel. This is compared with the value for the partner and the actual maximum used is the lower of the two values.

The value zero means the maximum message length for the queue manager.

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows, specify a value in the range zero through to the maximum message length for the queue manager.

See the MAXMSGL parameter of the ALTER QMGR command for more information.

On z/OS, specify a value in the range zero through 100 MB.

On other platforms, specify a value in the range zero through 4,194,304 bytes (4 MB).

**MCANAME**(*string*)
Message channel agent name.
This is reserved, and if specified must only be set to blanks (maximum length 20 characters).

**MCATYPE**
Specifies whether the message-channel-agent program should run as a thread or a process.

**PROCESS**
The message channel agent runs as a separate process

**THREAD**
The message channel agent runs as a separate thread

In situations where a threaded listener is required to service a large number of incoming requests, resources can become strained. In this case, you should use multiple listener processes and target incoming requests at specific listeners though the port number specified on the listener.

This parameter is valid only for channels with a channel type (CHLTYPE) of SDR, SVR, RQSTR, CLUSSDR, or CLUSRCVR. It is supported only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.

On z/OS it is supported only for channels with a channel type of CLUSRCVR. When specified in a CLUSRCVR definition, MCATYPE is used by a remote machine to determine the corresponding CLUSSDR definition.

**MCAUSER**(*string*)
Message channel agent user identifier.
If it is nonblank, it is the user identifier that is to be used by the message channel agent for authorization to access WebSphere MQ resources, including (if PUTAUT is DEF) authorization to put the message to the destination queue for receiver or requester channels.

If it is blank, the message channel agent uses its default user identifier.

The default user identifier is derived from the user ID that started the receiving channel. The possible values are:
- On z/OS, the user ID assigned to the channel-initiator started task by the z/OS started-procedures table.
**DEFINE CHANNEL**

- For TCP/IP, other than z/OS, the user ID from the inetd.conf entry, or the user that started the listener.
- For SNA, other than z/OS, the user ID from the SNA server entry or, in the absence of this the incoming attach request, or the user that started the listener.
- For NetBIOS or SPX, the user ID that started the listener.

The maximum length of the string is 64 characters on Windows and 12 characters on other platforms. On Windows, you can optionally qualify a user identifier with the domain name in the format `user@domain`.

This parameter is not valid for channels with a channel type (CHLTYPE) of CLNTCONN.

**MODENAME**(string)

LU 6.2 mode name (maximum length 8 characters).

This parameter is valid only for channels with a transport type (TRPTYPE) of LU 6.2. If TRPTYPE is not LU 6.2, the data is ignored and no error message is issued.

If specified, this should be set to the SNA mode name unless the CONNAME contains a side-object name, in which case it should be set to blanks. The actual name is then taken from the CPI-C Communications Side Object, or APPC side information data set.

See the information about configuration parameters for an LU 6.2 connection for your platform in the WebSphere MQ Intercommunication manual for more information.

This parameter is not valid for channels with a channel type (CHLTYPE) of RCVR or SVRCONN.

**MRDATA**(string)

Channel message-retry exit user data. The maximum length is 32 characters.

This is passed to the channel message-retry exit when it is called.

This parameter is valid only for channels with a channel type (CHLTYPE) of RCVR, RQSTR, or CLUSRCVR. It is not supported on z/OS.

**MREXIT**(string)

Channel message-retry exit name.

The format and maximum length of the name is the same as for MSGEXIT.

This parameter is valid only for channels with a channel type (CHLTYPE) of RCVR, RQSTR, or CLUSRCVR. It is not supported on z/OS.

**MRRTY**(integer)

The number of times the channel retries before it decides it cannot deliver the message.

This parameter controls the action of the MCA only if the message-retry exit name is blank. If the exit name is not blank, the value of MRRTY is passed to the exit for the exit’s use, but the number of retries performed (if any) is controlled by the exit, and not by this parameter.

The value must be in the range zero through 999 999 999. A value of zero means that no retries are performed.

This parameter is valid only for channels with a channel type (CHLTYPE) of RCVR, RQSTR, or CLUSRCVR. It is not supported on z/OS.
**MRTMR**(integer)

The minimum interval of time that must pass before the channel can retry the MQPUT operation. This time interval is in milliseconds.

This parameter controls the action of the MCA only if the message-retry exit name is blank. If the exit name is not blank, the value of MRTMR is passed to the exit for the exit's use, but the retry interval is controlled by the exit, and not by this parameter.

The value must be in the range zero through 999 999 999. A value of zero means that the retry is performed as soon as possible (provided that the value of MRRTY is greater than zero).

This parameter is valid only for channels with a channel type (CHLTYPE) of RCVR, RQSTR, or CLUSRCVR. It is not supported on z/OS.

**MSGDATA**(string)

User data for the channel message exit. The maximum length is 32 characters.

This data is passed to the channel message exit when it is called.

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, Solaris, and Windows, you can specify data for more than one exit program by specifying multiple strings separated by commas. The total length of the field must not exceed 999 characters.

On OS/400, you can specify up to 10 strings, each of length 32 characters. The first string of data is passed to the first message exit specified, the second string to the second exit, and so on.

On z/OS, you can specify up to 8 strings, each of length 32 characters. The first string of data is passed to the first message exit specified, the second string to the second exit, and so on.

On other platforms you can specify only one string of message exit data for each channel.

**MSGEXIT**(string)

Channel message exit name.

On Compaq NSK, there is only one channel user exit program. If the MSGEXIT, MREXIT, SCYEXIT, SENDEXIT, and RCVEXIT parameters are all left blank, the channel user exit is not invoked. If any of these parameters is nonblank, the channel exit program is called. You can enter text string for these parameters. The maximum length of the string is 128 characters. This string is passed to the exit program, but it is not used to determine the program name.

On other platforms, if this name is nonblank, the exit is called at the following times:

- Immediately after a message has been retrieved from the transmission queue (sender or server), or immediately before a message is put to a destination queue (receiver or requester).
  
  The exit is given the entire application message and transmission queue header for modification.

- At initialization and termination of the channel.

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, Solaris, and Windows, you can specify the name of more than one exit program by
DEFINE CHANNEL

specifying multiple strings separated by commas. However, the total number of characters specified must not exceed 999.

On OS/400, you can specify the names of up to 10 exit programs by specifying multiple strings separated by commas.

On z/OS, you can specify the names of up to 8 exit programs by specifying multiple strings separated by commas.

On other platforms you can specify only one message exit name for each channel.

For channels with a channel type (CHLTYPE) of CLNTCONN or SVRCONN, this parameter is not relevant, because message exits are not invoked for such channels.

The format and maximum length of the name depends on the environment:
• On Compaq OpenVMS and UNIX systems, it is of the form:
  libraryname(functionname)
  The maximum length of the string is 128 characters.
• On OS/2 Warp and Windows, it is of the form:
  dllname(functionname)
  where dllname is specified without the suffix (.DLL). The maximum length of the string is 128 characters.
• On OS/400®, it is of the form:
  prograname libname
  where program name occupies the first 10 characters and libname the second 10 characters (both padded to the right with blanks if necessary). The maximum length of the string is 20 characters.
• On z/OS, it is a load module name, maximum length 8 characters (128 characters are allowed for exit names for client-connection channels, subject to a maximum total length including commas of 999).

NETPRTY(integer)
The priority for the network connection. Distributed queuing chooses the path with the highest priority if there are multiple paths available. The value must be in the range zero through 9; zero is the lowest priority.

This parameter is valid only for CLUSRCVR channels.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

NPMSSPEED
The class of service for nonpersistent messages on this channel:

FAST Fast delivery for nonpersistent messages; messages might be lost if the channel is lost. This is the default. Messages are retrieved using MQGMO_SYNCPOINT_IF_PERSISTENT and so are not included in the batch unit of work.

NORMAL Normal delivery for nonpersistent messages.
If the sending side and the receiving side do not agree about this parameter, or one does not support it, NORMAL is used.

This parameter is valid only for channels with a CHLTYPE of SDR, SVR, RCVR, RQSTR, CLUSSDR, or CLUSRCVR. It is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

**PASSWORD**(string)
Password used by the message channel agent when attempting to initiate a secure LU 6.2 session with a remote message channel agent. The maximum length is 12 characters.

This parameter is valid only for channels with a channel type (CHLTYPE) of SDR, SVR, RQSTR, CLNTCONN, or CLUSSDR. On z/OS, it is supported only for channels with a channel type (CHLTYPE) of CLNTCONN.

Although the maximum length of the parameter is 12 characters, only the first 10 characters are used.

**PUTAUT**
Specifies which user identifiers should be used to establish authority to put messages to the destination queue (for messages channels) or to execute an MQI call (for MQI channels).

**DEF**
The default user ID is used. On z/OS this might involve using both the user ID received from the network and that derived from MCAUSER.

**CTX**
The user ID from the *UserIdentifier* field of the message descriptor is used. On z/OS this might involve also using the user ID received from the network or that derived from MCAUSER, or both.

**ONLYMCA**
The default user ID is used. Any user ID received from the network is not used. This value is supported only on z/OS.

**ALTMCA**
The user ID from the *UserIdentifier* field of the message descriptor is used. Any user ID received from the network is not used. This value is supported only on z/OS.

This parameter is valid only for channels with a channel type (CHLTYPE) of RCVR, RQSTR, CLUSRCVR, or SVRCONN. CTX and ALTMCA are not valid for SVRCONN channels.

**QMNAME**(string)
Queue manager name.

For channels with a channel type (CHLTYPE) of CLNTCONN, this is the name of the queue manager to which an application running in the MQI client environment can request connection.

For channels of other types this parameter is not valid.

**QSGDISP**
This parameter applies to z/OS only.

Specifies the disposition of the object to which you are applying the command (that is, where it is defined and how it behaves).
### DEFINE CHANNEL

<table>
<thead>
<tr>
<th>QSGDISP</th>
<th>ALTER</th>
<th>DEFINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY</td>
<td>The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(COPY). Any object residing in the shared repository, or any object defined using a command that had the parameters QSGDISP(QMGR), is not affected by this command.</td>
<td>The object is defined on the page set of the queue manager that executes the command using the QSGDISP(GROUP) object of the same name as the 'LIKE' object.</td>
</tr>
<tr>
<td>GROUP</td>
<td>The object definition resides in the shared repository. The object was defined using a command that had the parameters QSGDISP(GROUP). Any object residing on the page set of the queue manager that executes the command (except a local copy of the object) is not affected by this command. If the command is successful, the following command is generated and sent to all active queue managers in the queue-sharing group to refresh local copies on page set zero:</td>
<td>The object definition resides in the shared repository. This is allowed only if the queue manager is in a queue-sharing group. If the definition is successful, the following command is generated and sent to all active queue managers in the queue-sharing group to make or refresh local copies on page set zero:</td>
</tr>
<tr>
<td>PRIVATE</td>
<td>The object resides on the page set of the queue manager that executes the command, and was defined with QSGDISP(QMGR) or QSGDISP(COPY). Any object residing in the shared repository is unaffected.</td>
<td>Not permitted.</td>
</tr>
<tr>
<td>QMGR</td>
<td>The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(QMGR). Any object residing in the shared repository, or any local copy of such an object, is not affected by this command. This is the default value.</td>
<td>The object is defined on the page set of the queue manager that executes the command. This is the default value.</td>
</tr>
</tbody>
</table>

**RCVDATA(string)**

Channel receive exit user data (maximum length 32 characters).

This is passed to the channel receive exit when it is called.

- On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, Solaris, and Windows, you can specify data for more than one exit program by specifying multiple strings separated by commas. The total length of the field must not exceed 999 characters.
- On OS/400, you can specify up to 10 strings, each of length 32 characters. The first string of data is passed to the first receive exit specified, the second string to the second exit, and so on.
- On z/OS, you can specify up to 8 strings, each of length 32 characters. The first string of data is passed to the first receive exit specified, the second string to the second exit, and so on.
- On other platforms you can specify only one string of receive exit data for each channel.
**DEFINE CHANNEL**

**RCVEXIT**(string)
Channel receive exit name.

On platforms other than Compaq NSK, if this name is nonblank, the exit is called at the following times:

- Immediately before the received network data is processed.
  The exit is given the complete transmission buffer as received. The contents of the buffer can be modified as required.
- At initialization and termination of the channel.

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, Solaris, and Windows, you can specify the name of more than one exit program by specifying multiple strings separated by commas. However, the total number of characters specified must not exceed 999.

On OS/400, you can specify the names of up to 10 exit programs by specifying multiple strings separated by commas.

On z/OS, you can specify the names of up to 8 exit programs by specifying multiple strings separated by commas.

On other platforms you can specify only one receive exit name for each channel.

The format and maximum length of the name is the same as for MSGEXIT.

**REPLACE** and **NOREPLACE**
Whether the existing definition (and on z/OS, with the same disposition) is to be replaced with this one. This is optional. The default is NOREPLACE. Any object with a different disposition is not changed.

This parameter applies only to the DEFINE CHANNEL command.

**REPLACE**
The definition should replace any existing definition of the same name. If a definition does not exist, one is created. Note that REPLACE does not alter the channel status.

**NOREPLACE**
The definition should not replace any existing definition of the same name.

**SCYDATA**(string)
Channel security exit user data (maximum length 32 characters).

This is passed to the channel security exit when it is called.

**SCYEXIT**(string)
Channel security exit name.

On platforms other than Compaq NSK, if this name is nonblank, the exit is called at the following times:

- Immediately after establishing a channel.
  Before any messages are transferred, the exit is given the opportunity to instigate security flows to validate connection authorization.
- Upon receipt of a response to a security message flow.
  Any security message flows received from the remote processor on the remote queue manager are given to the exit.
DEFINE CHANNEL

- At initialization and termination of the channel.

The format and maximum length of the name is the same as for MSGEXIT.

SENDDATA(string)
Channel send exit user data. The maximum length is 32 characters.
This is passed to the channel send exit when it is called.

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, Solaris, and Windows, you can specify data for more than one exit program by specifying multiple strings separated by commas. The total length of the field must not exceed 999 characters.

On OS/400, you can specify up to 10 strings, each of length 32 characters. The first string of data is passed to the first send exit specified, the second string to the second exit, and so on.

On z/OS, you can specify up to 8 strings, each of length 32 characters. The first string of data is passed to the first send exit specified, the second string to the second exit, and so on.

On other platforms you can specify only one string of send exit data for each channel.

SENDEXIT(string)
Channel send exit name.
On platforms other than Compaq NSK, if this name is nonblank, the exit is called at the following times:
- Immediately before data is sent out on the network.
  The exit is given the complete transmission buffer before it is transmitted. The contents of the buffer can be modified as required.
- At initialization and termination of the channel.

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, Solaris, and Windows, you can specify the name of more than one exit program by specifying multiple strings separated by commas. However, the total number of characters specified must not exceed 999.

On OS/400, you can specify the names of up to 10 exit programs by specifying multiple strings separated by commas.

On z/OS, you can specify the names of up to 8 exit programs by specifying multiple strings separated by commas.

On other platforms you can specify only one send exit name for each channel.

The format and maximum length of the name is the same as for MSGEXIT.

SEQWRAP(integer)
When this value is reached, sequence numbers wrap to start again at 1.
This value is nonnegotiable and must match in both the local and remote channel definitions.
The value must be in the range 100 through 999 999 999.
This parameter is valid only for channels with a channel type (CHLTYP) of SDR, SVR, RCVR, RQSTR, CLUSSDR, or CLUSRCVR.
SHORTRTY(integer)
The maximum number of attempts that are made by a sender, server, or cluster-sender channel to connect to the remote queue manager, at intervals specified by SHORTTMR, before the (normally longer) LONGRTY and LONGTMR are used.

Retry attempts are made if the channel fails to connect initially (whether it is started automatically by the channel initiator or by an explicit command), and also if the connection fails after the channel has successfully connected. However, if the cause of the failure is such that retry is unlikely to be successful, retries are not attempted.

The value must be in the range zero through 999 999 999.

This parameter is valid only for channels with a channel type (CHLTYPE) of SDR, SVR, CLUSSDR, or CLUSRCVR.

SHORTTMR(integer)
For short retry attempts, this is the maximum number of seconds to wait before reattempting connection to the remote queue manager.

The time is approximate; zero means that another connection attempt is made as soon as possible.

The interval between retries might be extended if the channel has to wait to become active.

The value must be in the range zero through 999 999 999.

Note: For implementation reasons, the maximum retry interval that can be used is 999 999; values exceeding this are treated as 999 999.

This parameter is valid only for channels with a channel type (CHLTYPE) of SDR, SVR, CLUSSDR, or CLUSRCVR.

SSLCAUTH
Defines whether WebSphere MQ requires a certificate from the SSL client or not. The initiating end of the channel acts as the SSL client, so this applies to the end of the channel that receives the initiation flow, which acts as the SSL server.

This parameter is supported only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.

This parameter is valid only for channels with a channel type (CHLTYPE) of RCVR, SVRCONN, CLUSRCVR, SVR, or RQSTR.

The parameter is used only for channels with SSLCIPH specified. If SSLCIPH is blank, the data is ignored and no error message is issued.

REQUIRED
WebSphere MQ requires and validates a certificate from the SSL client. This is the default.

OPTIONAL
The peer SSL client system might still send a certificate. If it does, the contents of this certificate are validated as normal.

SSLCIPH(string)
CipherSpec used on the channel. The maximum length is 32 characters.

This parameter is valid on all channel types.
This parameter is supported only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.

This parameter is valid only for channels with a transport type (TRPTYPE) of TCP. If the TRPTYPE is not TCP, the data is ignored and no error message is issued.

The SSLCIPH values must specify the same CipherSpec on both ends of the channel.

Specify the name of the CipherSpec you are using. The CipherSpecs that can be used with WebSphere MQ SSL support are shown in Table 3.

<table>
<thead>
<tr>
<th>CipherSpec name</th>
<th>Hash algorithm</th>
<th>Encryption algorithm</th>
<th>Encryption bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL_MD5</td>
<td>MD5</td>
<td>None</td>
<td>0</td>
</tr>
<tr>
<td>NULL_SHA</td>
<td>SHA</td>
<td>None</td>
<td>0</td>
</tr>
<tr>
<td>RC4_MD5_EXPORT</td>
<td>MD5</td>
<td>RC4</td>
<td>40</td>
</tr>
<tr>
<td>RC4_MD5_US</td>
<td>MD5</td>
<td>RC4</td>
<td>128</td>
</tr>
<tr>
<td>RC4_SHA_US</td>
<td>SHA</td>
<td>RC4</td>
<td>128</td>
</tr>
<tr>
<td>RC2_MD5.Export</td>
<td>MD5</td>
<td>RC2</td>
<td>40</td>
</tr>
<tr>
<td>DES_SHA.Export</td>
<td>SHA</td>
<td>DES</td>
<td>56</td>
</tr>
<tr>
<td>RC4_56_SHAExporter1024</td>
<td>SHA</td>
<td>RC4</td>
<td>56</td>
</tr>
<tr>
<td>DES_SHAExporter1024</td>
<td>SHA</td>
<td>DES</td>
<td>56</td>
</tr>
<tr>
<td>TRIPLE DES_SHA_US</td>
<td>SHA</td>
<td>3DES</td>
<td>128</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_128_CBC_SHA</td>
<td>SHA</td>
<td>AES</td>
<td>128</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_256_CBC_SHA</td>
<td>SHA</td>
<td>AES</td>
<td>256</td>
</tr>
</tbody>
</table>

Notes:
1. Not available for z/OS and OS/400 platforms
2. Specifies a 1024-bit handshake key size
3. Available for AIX platforms only

If the SSLCIPH parameter is blank, no attempt is made to use SSL on the channel.

SSLPEER(string)

Specifies the filter to use to compare with the Distinguished Name of the certificate from the peer queue manager or client at the other end of the channel. (A 'Distinguished Name' is the identifier of the SSL certificate.) If the Distinguished Name in the certificate received from the peer does not match the SSLPEER filter, the channel does not start.

This parameter is supported only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.

This parameter is optional; if it is not specified, the Distinguished Name of the peer is not checked at channel start up. (The Distinguished Name from the certificate is still written into the SSLPEER definition held in memory, and passed to the security exit). If SSLCIPH is blank, the data is ignored and no error message is issued.

This parameter is valid for all channel types.
The SSLPEER value is specified in the standard form used to specify a
Distinguished Name. For example:

```
SSLPEER('CN="xxx yyy zzz",O=xxx,C=xxx')
```

You can use a semi-colon as a separator instead of a comma.

The possible attribute types supported are:

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN</td>
<td>common name</td>
</tr>
<tr>
<td>T</td>
<td>title (not supported on z/OS, UNIX, or OS/400)</td>
</tr>
<tr>
<td>OU</td>
<td>organizational unit name</td>
</tr>
<tr>
<td>O</td>
<td>organization name</td>
</tr>
<tr>
<td>L</td>
<td>locality name</td>
</tr>
<tr>
<td>S, ST, or SP</td>
<td>state or province name</td>
</tr>
<tr>
<td>C</td>
<td>country</td>
</tr>
</tbody>
</table>

If any of the unsupported attribute types are specified in the SSLPEER
string, an error is output either when the attribute is defined or at run time
(depending on which platform you are running on), and the string is
deemed not to have matched the flowed certificate’s Distinguished Name.

If the flowed certificate’s Distinguished Name contains multiple OU
(organisational unit) attributes, and SSLPEER specifies these attributes to
be compared, they must match in the order they are found in the
certificate’s Distinguished Name, and must start with the first OU, or an
asterisk. For example, if the flowed certificate’s Distinguished Name
contains the OUs `OU=One,OU=Two,OU=Three`, specifying the following
SSLPEER values will work:

```
('OU=One,OU=Two')
('OU=*,OU=Two,OU=Three')
('OU=*,OU=Two')
```

but specifying the following SSLPEER values will fail:

```
('OU=Two,OU=Three')
('OU=One,OU=Three')
('OU=Two')
```

Any or all of the attribute values can be generic, either an asterisk (*) on its
own, or a stem with initiating or trailing asterisks. This allows the
SSLPEER to match any Distinguished Name value, or any value starting
with the stem for that attribute.

If an asterisk is specified at the beginning or end of any attribute value in
the Distinguished Name on the certificate, you can specify ‘\*’ to check for
an exact match in SSLPEER. For example, if you have an attribute of
`CN="Test\*"` in the Distinguished Name of the certificate, you can use the
following command:

```
SSLPEER('CN=Test\*')
```

The maximum length of the parameter is 1024 bytes on Windows and
UNIX platforms, and 256 bytes on z/OS.
DEFINE CHANNEL

TPNAME(string)
LU 6.2 transaction program name (maximum length 64 characters).
This parameter is valid only for channels with a transport type (TRPTYPE)
of LU 6.2.
On Compaq NSK, this should be set to the local TP name. This can be
followed by the name of the TP on the remote machine, for example:
TPNAME('localtp[.remotetp]')
Both names can be up to 16 characters in length.
The name can also be nonblank for client connection channels to be used
with OS/2 Warp.
On other platforms, this should be set to the SNA transaction program
name, unless the CONNAME contains a side-object name in which case it
should be set to blanks. The actual name is taken instead from the CPI-C
Communications Side Object, or the APPC side information data set.

See the information about configuration parameters for an LU 6.2
connection for your platform in the WebSphere MQ Intercommunication
manual for more information.

On Windows SNA Server, and in the side object on z/OS, the TPNAME is
wrapped to upper case.

This parameter is not valid for channels with a channel type (CHLTYPE) of
RCVR.

TRPTYPE
Transport type to be used.
On AIX, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS,
this parameter is optional because, if you do not enter a value, the value
specified in the SYSTEM.DEF.channel-type definition is used. However, no
check is made that the correct transport type has been specified if the
channel is initiated from the other end. On z/OS, if the
SYSTEM.DEF.channel-type definition does not exist, the default is LU62.
This is required on all other platforms.

DECNET
   DECnet (supported only on Compaq OpenVMS)
LU62  SNA LU 6.2

NETBIOS
NetBIOS (supported only on OS/2 Warp, Windows, and DOS; it
also applies to z/OS for defining client-connection channels that
connect to servers on the platforms supporting NetBIOS)

SPX  Sequenced packet exchange (supported only on OS/2 Warp,
Windows, and DOS; it also applies to z/OS for defining
client-connection channels that connect to servers on the platforms
supporting SPX)

TCP  Transmission Control Protocol - part of the TCP/IP protocol suite
UDP  User Datagram Protocol - part of the TCP/IP protocol suite
DEFINE CHANNEL

(supported only on AIX); this option is available only for connection to MQSeries for Windows, V2.0, with CSD02

USERID(string)
Task user identifier. The maximum length is 12 characters.

This is used by the message channel agent when attempting to initiate a secure LU 6.2 session with a remote message channel agent.

This parameter is valid only for channels with a channel type (CHLTYPE) of SDR, SVR, RQSTR, CLNTCONN, or CLUSSDR. On z/OS, it is supported only for CLNTCONN channels.

Although the maximum length of the parameter is 12 characters, only the first 10 characters are used.

XMITQ(string)
Transmission queue name.

The name of the queue from which messages are retrieved. See “Rules for naming WebSphere MQ objects” on page 4

This parameter is valid only for channels with a channel type (CHLTYPE) of SDR or SVR. For these channel types this parameter is required.
Use **DEFINE MAXSMSGS** to define the maximum number of messages that a task can get or put within a single unit of recovery.

**Notes:**

1. This command is valid only on z/OS and is retained for compatibility, although it can no longer be issued from the CSQINP1 initialization input data set. You should use the MAXUMSGS parameter of the ALTER QMGR command instead.
2. You can issue the DEFINE MAXSMSGS command to change the number of messages allowed. Once a value is set, it is preserved during a queue manager restart.

**Synonym:** DEF MAXSM

**DEFINE MAXSMSGS**

```
DEFINE MAXSMSGS(integer)
```

**Parameter descriptions**

**(integer)**

The maximum number of messages that a task can get or put within a single unit of recovery. This value must be an integer in the range 1 through 999 999 999. The default value is 10 000.

The number includes any trigger messages and report messages generated within the same unit of recovery.

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1 and must be blank or the local queue manager.

' ' The command is executed on the queue manager on which it was entered. This is the default value.
`DEFINE MAXSMSGS`
**DEFINE NAMELIST**

Use **DEFINE NAMELIST** to define a list of names. This is most commonly a list of cluster names or queue names.

**Notes:**

1. On UNIX systems, the command is valid only on AIX, HP-UX, Linux and Solaris.

**Synonym:** DEF NL

**DEFINE NAMELIST**

```
DEFINE NAMELIST (name) CMDSCOPE(' ') CMDSCOPE(qmgr-name) CMDSCOPE(*)
QSGDISP(QMGR) QSGDISP(COPY) QSGDISP(GROUP)
```

**Define attrs:**

```
LIKE(name-list-name) NOREPLACE REPLACE
```

**Namelist attrs:**

```
DESCR(' ') DESCR(string) NAMES(name) NLTYPE(NONE)
- - - - -
- NLTYPE(QUEUE) NLTYPE(Q) NLTYPE(CLUSTER) NLTYPE(AUTHINFO)
```

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.

2. Valid only on z/OS.

3. This is the default supplied with WebSphere MQ, but your installation might have changed it.
Parameter descriptions

The parameter descriptions also apply to the ALTER NAMELIST command, with the following exceptions:

- The LIKE parameter applies only to the DEFINE NAMELIST command.
- The REPLACE and NOREPLACE parameter applies only to the DEFINE NAMELIST command.
- The variations in the CMDSCOPE and QSGDISP parameters between the ALTER NAMELIST and DEFINE NAMELIST commands are described.

(name) Name of the list. This is required.

The name must not be the same as any other namelist name currently defined on this queue manager (unless REPLACE or ALTER is specified). See “Rules for naming WebSphere MQ objects” on page 4.

CMDSCOPE

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP.

```
'' The command is executed on the queue manager on which it was entered. This is the default value.
```

qmgr-name

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name other than the queue manager on which it was entered, only if you are using a shared queue environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

DESCR(string)

Plain-text comment. It provides descriptive information about the namelist when an operator issues the DISPLAY NAMELIST command (see “DISPLAY NAMELIST” on page 215).

It should contain only displayable characters. The maximum length is 64 characters. In a DBCS installation, it can contain DBCS characters (subject to a maximum length of 64 bytes).

Note: If characters are used that are not in the coded character set identifier (CCSID) for this queue manager, they might be translated incorrectly if the information is sent to another queue manager.

LIKE(namelist-name)

The name of a namelist, whose parameters will be used to model this definition.

This parameter applies only to the DEFINE NAMELIST command

If this field is not filled in, and you do not complete the parameter fields related to the command, the values are taken from the default definition for namelists on this queue manager.
**DEFINE NAMELIST**

This is equivalent to specifying:

LIKE(SYSTEM.DEFAULT.NAMELIST)

A default namelist definition is provided, but it can be altered by the installation to the default values required. See "Rules for naming WebSphere MQ objects" on page 4.

On WebSphere MQ for z/OS, the queue manager searches page set zero for an object with the name you specify. The disposition of the LIKE object is not copied to the object you are defining.

**Notes:**
1. QSGDISP (GROUP) objects are not searched.
2. LIKE is ignored if QSGDISP(COPY) is specified.

**NAMES(name, ...)**

List of names.

The names can be of any type, but must conform to the rules for naming WebSphere MQ objects, with a maximum length of 48 characters.

An empty list is valid: specify NAMES(). The maximum number of names in the list is 256.

**NLTYPE**

Indicates the type of names in the namelist.

This parameter is valid only on z/OS.

**NONE**

The names are of no particular type.

**QUEUE or Q**

A namelist that holds a list of queue names.

**CLUSTER**

A namelist that is associated with clustering, containing a list of the cluster names.

**AUTHINFO**

This namelist is associated with SSL and should contain a list of authentication information object names.

Namelists used for clustering must have NLTYPE(CLUSTER) or NLTYPE(NONE).

Namelists used for SSL must have NLTYPE(AUTHINFO).

**QSGDISP**

This parameter applies to z/OS only.

Specifies the disposition of the object to which you are applying the command (that is, where it is defined and how it behaves).
<table>
<thead>
<tr>
<th>QSGDISP</th>
<th>ALTER</th>
<th>DEFINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY</td>
<td>The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(COPY). Any object residing in the shared repository, or any object defined using a command that had the parameters QSGDISP(QMGR), is not affected by this command.</td>
<td>The object is defined on the page set of the queue manager that executes the command using the QSGDISP(GROUP) object of the same name as the 'LIKE' object.</td>
</tr>
<tr>
<td>GROUP</td>
<td>The object definition resides in the shared repository. The object was defined using a command that had the parameters QSGDISP(GROUP). Any object residing on the page set of the queue manager that executes the command (except a local copy of the object) is not affected by this command. If the command is successful, the following command is generated and sent to all active queue managers in the queue-sharing group so that they refresh local copies on page set zero: DEFINE NAMELIST(name) REPLACE QSGDISP(COPY)</td>
<td>The object definition resides in the shared repository. This is allowed only if the queue manager is in a queue-sharing group. If the definition is successful, the following command is generated and sent to all active queue managers in the queue-sharing group so that they make or refresh local copies on page set zero: DEFINE NAMELIST(name) REPLACE QSGDISP(COPY)</td>
</tr>
<tr>
<td>PRIVATE</td>
<td>The object resides on the page set of the queue manager that executes the command, and was defined with QSGDISP(QMGR) or QSGDISP(COPY). Any object residing in the shared repository is unaffected.</td>
<td>Not permitted.</td>
</tr>
<tr>
<td>QMGR</td>
<td>The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(QMGR). Any object residing in the shared repository, or any local copy of such an object, is not affected by this command. This is the default value.</td>
<td>The object is defined on the page set of the queue manager that executes the command. This is the default value.</td>
</tr>
</tbody>
</table>

**REPLACE** and **NOREPLACE**

Whether the existing definition (and on z/OS, with the same disposition) is to be replaced with this one. This is optional. The default is NOREPLACE. Any object with a different disposition is not changed.

This parameter applies only to the DEFINE NAMELIST command

**REPLACE**

The definition should replace any existing definition of the same name. If a definition does not exist, one is created.

**NOREPLACE**

The definition should not replace any existing definition of the same name.
USE DEFINE PROCESS TO DEFINE A NEW WEBSHARE MQ PROCESS DEFINITION, AND SET ITS PARAMETERS.

SYNONYM: DEF PRO

```
DEFINE PROCESS
```

Use DEFINE PROCESS to define a new WebSphere MQ process definition, and set its parameters.

Synonym: DEF PRO

**DEFINE PROCESS**

```
DEFINE PROCESS (process-name)
```

**Define attrs:**

```
LIKE (process-name)
```

**Process attrs:**

```
APPLTYPE
CICS
DEF
DOS
IMS
MVS
NOTESAGENT
NSK
OPENVMS
OS2
OS400
UNIX
WINDOWS
WINDOWS
```

```
APPLICID
APPLICID (string)
DESCR
DESCR (string)
```
Parameter descriptions

The parameter descriptions also apply to the ALTER PROCESS command, with the following exceptions:

- The LIKE parameter applies only to the DEFINE PROCESS command.
- The NOREPLACE and REPLACE parameter applies only to the DEFINE PROCESS command.
- The variations in the CMDSCOPE and QSGDISP parameters between the ALTER PROCESS and DEFINE PROCESS commands are described.

(process-name)

Name of the WebSphere MQ process definition (see "Rules for naming WebSphere MQ objects" on page 4). This is required.

The name must not be the same as any other process definition currently defined on this queue manager (unless REPLACE is specified).

APPLICID(string)

The name of the application to be started. This might typically be a fully-qualified file name of an executable object. The maximum length is 256 characters.

For a CICS application this is a CICS transaction ID, and for an IMS application it is an IMS transaction ID.

On z/OS, for distributed queuing using CICS it must be “CKSG”, and for distributed queuing without CICS®, it must be “CSQX START”.

APPLTYPE(string)

The type of application to be started. Valid application types are:

integer

A system-defined application type in the range zero through 65 535 or a user-defined application type in the range 65 536 through 999 999 999.

For certain values in the system range, a parameter from the following list can be specified instead of a numeric value:

CICS Represents a CICS transaction.
DOS Represents a DOS application.
IMS Represents an IMS transaction.
MVS Represents a z/OS application (batch or TSO).
DEF
This causes the default application type for the platform at which the command is interpreted to be stored in the process definition. This default cannot be changed by the installation. If the platform supports clients, this is interpreted as the default application type of the server.

Only application types (other than user-defined types) that are supported on the platform at which the command is executed should be used:

- On Compaq OpenVMS, OPENVMS is supported
- On z/OS, CICS (default), DOS, IMS, MVS, OS2, UNIX, WINDOWS, WINDOWSNT, and DEF are supported
- On OS/400, OS400 (default), CICS, and DEF are supported
- On OS/2 Warp, OS2 (default), DOS, WINDOWS, UNIX, CICS, and DEF are supported
- On Compaq NSK, NSK is supported.
- On UNIX systems, UNIX (default), OS2, DOS, WINDOWS, CICS, and DEF are supported
- On Windows, WINDOWSNT (default), DOS, WINDOWS, OS2, UNIX, CICS, and DEF are supported

CMDSCOPE
This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group. CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP.

'' The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name other than the queue manager on which it was entered, only if you are using a shared queue environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.
DEFINE PROCESS

DESCR(string)
Plain-text comment. It provides descriptive information about the object when an operator issues the DISPLAY PROCESS command.

It should contain only displayable characters. The maximum length is 64 characters. In a DBCS installation, it can contain DBCS characters (subject to a maximum length of 64 bytes).

Note: If characters are used that are not in the coded character set identifier (CCSID) for this queue manager, they might be translated incorrectly if the information is sent to another queue manager.

ENVRDATA(string)
A character string that contains environment information pertaining to the application to be started. The maximum length is 128 characters.

The meaning of ENVRDATA is determined by the trigger-monitor application. The trigger monitor provided by WebSphere MQ appends ENVRDATA to the parameter list passed to the started application. The parameter list consists of the MQTMC2 structure, followed by one blank, followed by ENVRDATA with trailing blanks removed.

Notes:
1. On z/OS, ENVRDATA is not used by the trigger-monitor applications provided by WebSphere MQ.
2. On UNIX systems, ENVRDATA can be set to the ampersand character to make the started application run in the background.

LIKE(process-name)
The name of an object of the same type, whose parameters will be used to model this definition.

If this field is not filled in, and you do not complete the parameter fields related to the command, the values are taken from the default definition for this object.

This is equivalent to specifying:
LIKE(SYSTEM.DEFAULT.PROCESS)

A default definition for each object type is provided, but these can be altered by the installation to the default values required. See “Rules for naming WebSphere MQ objects” on page 4.

On WebSphere MQ for z/OS, the queue manager searches page set zero for an object with the name you specify. The disposition of the LIKE object is not copied to the object you are defining.

Notes:
1. QSGDISP (GROUP) objects are not searched.
2. LIKE is ignored if QSGDISP(COPY) is specified.

QSGDISP
This parameter applies to z/OS only.

Specifies the disposition of the object to which you are applying the command (that is, where it is defined and how it behaves).
## DEFINE PROCESS

<table>
<thead>
<tr>
<th>QSGDISP</th>
<th>ALTER</th>
<th>DEFINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY</td>
<td>The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(COPY). Any object residing in the shared repository, or any object defined using a command that had the parameters QSGDISP(QMGR), is not affected by this command.</td>
<td>The object is defined on the page set of the queue manager that executes the command using the QSGDISP(GROUP) object of the same name as the 'LIKE' object.</td>
</tr>
<tr>
<td>GROUP</td>
<td>The object definition resides in the shared repository. The object was defined using a command that had the parameters QSGDISP(GROUP). Any object residing on the page set of the queue manager that executes the command (except a local copy of the object) is not affected by this command. If the command is successful, the following command is generated and sent to all active queue managers in the queue-sharing group to refresh local copies on page set zero: DEFINE PROCESS(process-name) REPLACE QSGDISP(COPY)</td>
<td></td>
</tr>
<tr>
<td>PRIVATE</td>
<td>The object resides on the page set of the queue manager that executes the command, and was defined with QSGDISP(QMGR) or QSGDISP(COPY). Any object residing in the shared repository is unaffected.</td>
<td>Not permitted.</td>
</tr>
<tr>
<td>QMGR</td>
<td>The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(QMGR). Any object residing in the shared repository, or any local copy of such an object, is not affected by this command. This is the default value.</td>
<td>The object is defined on the page set of the queue manager that executes the command. This is the default value.</td>
</tr>
</tbody>
</table>

### REPLACE and NOREPLACE

Whether the existing definition (and on z/OS, with the same disposition) is to be replaced with this one. This is optional. The default is NOREPLACE. Any object with a different disposition is not changed.

**REPLACE**

The definition should replace any existing definition of the same name. If a definition does not exist, one is created.

**NOREPLACE**

The definition should not replace any existing definition of the same name.

**USERDATA**(string)

A character string that contains user information pertaining to the application defined in the APPLICID that is to be started. The maximum length is 128 characters.

The meaning of USERDATA is determined by the trigger-monitor application. The trigger monitor provided by WebSphere MQ simply
passes USERDATA to the started application as part of the parameter list. The parameter list consists of the MQTMC2 structure (containing USERDATA), followed by one blank, followed by ENVRDATA with trailing blanks removed.

For WebSphere MQ message channel agents, the format of this field is a channel name of up to 20 characters. See the WebSphere MQ Intercommunication manual for information about what these need as APPLICID.

On Compaq NSK, a character string containing spaces must be enclosed in double quotation marks.
Use DEFINE PSID to define a page set and associated buffer pool.

**Notes:**

1. You can issue DEFINE PSID only from the CSQINP1 initialization data set. If more than one DEFINE PSID command is issued for the same page set, only the last one is actioned.

2. You can use the DISPLAY USAGE TYPE(PAGESET) command to display information about page sets (see “DISPLAY USAGE” on page 261).

**Synonym:** DEF PSID

**DEFINE PSID**

```sql
DEFINE PSID(psid-number)
```

**BUFFPOOL(integer)**

**Parameter descriptions**

**(psid-number)**

Identifier of the page set. This is required.

In WebSphere MQ for z/OS a one-to-one relationship exists between page sets and the VSAM data sets used to store the pages. The identifier consists of a number in the range 00 through 99. It is used to generate a *ddname*, which references the VSAM ESDS data set, in the range CSQP0000 through CSQP0099.

The identifier must not be the same as any other page set identifier currently defined on this queue manager.

**BUFFPOOL(integer)**

The buffer pool number (in the range zero through 15). This is optional.

The default is zero.

See “DEFINE BUFFPOOL” on page 66
DEFINE queues

This section contains the following commands:

- "DEFINE QALIAS"
- "DEFINE QLOCAL" on page 127
- "DEFINE QMODEL" on page 129
- "DEFINE QREMOTE" on page 131

These queues are supported on the following platforms:

<table>
<thead>
<tr>
<th>Compaq NSK</th>
<th>Compaq OpenVMS</th>
<th>OS/400</th>
<th>OS/2 Warp</th>
<th>UNIX systems</th>
<th>Windows</th>
<th>z/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
</tr>
</tbody>
</table>

DEFINE QALIAS

Use DEFINE QALIAS to define a new alias queue, and set its parameters.

**Note:** An alias queue provides a level of indirection to another queue. The queue to which the alias refers must be another local or remote queue, defined at this queue manager. It cannot be another alias queue.

**Synonym:** DEF QA

**DEFINE QALIAS**

```
DEFINE QALIAS(q-name)
```

**Define attrs:**

```
LIKE(qalias-name)
```

**Common q attrs:**

```
DEFPRTY(integer)
DEFPRTY(0)
```

```
DEFPSIST(YES)
DEFPSIST(NO)
```

```
DESCR('')
DESCR(string)
```

```
PUT(ENABLED)
PUT(DISABLED)
```
DEFINE QUEUES

Alias q attrs:

1. CLUSNL(' ') (3)
2. CLUSTER(' ') (3)
3. DEFBIND(OPEN) (4)
4. DEFBIND(NOTFIXED)
5. GET(ENABLED) (3)
6. GET(DISABLED)
7. SCOPE(QMGR) (5)
8. SCOPE(CELL)
9. TARGQ('') (3)
10. TARGQ(string)

Notes:

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.
3. This is the default supplied with WebSphere MQ, but your installation might have changed it.
4. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
5. Valid only on Compaq OpenVMS, OS/2 Warp, OS/400, UNIX systems, and Windows.
6. Valid only on Compaq OpenVMS, OS/2 Warp, UNIX systems, and Windows.
DEFINE QLOCAL

Use DEFINE QLOCAL to define a new local queue, and set its parameters.

**Synonym:** DEF QL

**DEFINE QLOCAL**

```
DEFINE QLOCAL(q-name)

   CMDSCOPE(' ')
   CMDSCOPE(qmgr-name)
   CMDSCOPE(*)

   QSGDISP(QMGR)
   QSGDISP(COPY)
   QSGDISP(GROUP)
   QSGDISP(SHARED)

   DEFINE attrs
   Common q attrs
   Local q attrs
```

**Define attrs:**

```
LIKE(qlocal-name)
NOREPLACE
REPLACE
```

**Common q attrs:**

```
DEFPRTY(0)
DEFPRTY(integer)
DEFPSIST(NO)
DEFPSIST(YES)
DESCR(' ')
DESCR(string)
PUT(ENABLED)
PUT(DISABLED)
```

**Local q attrs:**

```
BOQNAME(' ')
BOQNAME(string)
BOTHRESH(0)
BOTHRESH(integer)
CFSTRUCT(' ')
CFSTRUCT(structure-name)
CLUSNL(' ')
CLUSNL(nlname)
CLUSTER(' ')
CLUSTER(clustername)
DEFBIND(OPEN)
DEFBIND(NOTFIXED)
DEFSOPT(SHARED)
DEFSOPT(EXCL)
DISTL(NO)
DISTL(YES)
GET(ENABLED)
GET(DISABLED)
NOHARDENBO
HARDENBO
INDXTYPE(NONE)
INDXTYPE(MSGID)
INDXTYPE(CORRELID)
INDXTYPE(GROUPID)
INDXTYPE(MSGTOKEN)
MAXDEPTH(5000)
MAXDEPTH(integer)
```

Chapter 2. The MQSC commands 127
**DEFINE QLOCAL**

Notes:

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Used only on z/OS.
3. This is the default supplied with WebSphere MQ, but your installation might have changed it.
4. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
5. This is the default supplied with WebSphere MQ (except on z/OS, where it is EXCL), but your installation might have changed it.
6. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
7. This is the default supplied with WebSphere MQ (except on z/OS, where it is 999 999 999), but your installation might have changed it.
8. Valid only on Compaq OpenVMS, OS/2 Warp, OS/400, UNIX systems, and Windows.
9. Valid only on Compaq OpenVMS, OS/2 Warp, UNIX systems, and Windows.
10. This is the default supplied with WebSphere MQ (except on z/OS, where it is NOSHARE), but your installation might have changed it.
DEFINE QMODEL

Use DEFINE QMODEL to define a new model queue, and set its parameters.

A model queue is not a real queue, but a collection of attributes that you can use when creating dynamic queues with the MQOPEN API call.

When it has been defined, a model queue (like any other queue) has a complete set of applicable attributes, even if some of these are defaults.

**Synonym:** DEF QM

### DEFINE QMODEL

**Define attrs:**

- `LIKE(qmodel-name)`
- `NOREPLACE`
- `REPLACE`

**Common q attrs:**

- `DEFPRTY(0)`
- `DEFPRTY(integer)`
- `DEFPSIST(NO)`
- `DEFPSIST(YES)`
- `DESCR('')`
- `DESCR(string)`
- `PUT(ENABLED)`
- `PUT(DISABLED)`

**Local q attrs:**

- `BOQNAME('')`
- `BOQNAME(string)`
- `BOTHRESH(0)`
- `BOTHRESH(integer)`
- `CFSTRUCT('')`
- `CFSTRUCT(structure-name)`
- `DEFSOPT(EXCL)`
- `DEFSOPT(SHARED)`
- `DISTL(NO)`
- `DISTL(YES)`
- `GET(ENABLED)`
- `GET(DISABLED)`
- `NOHARDENBO`
- `HARDENBO`
- `INDXTYPE(NONE)`
- `INDXTYPE(MSGID)`
- `CORRELID`
- `GROUPID`
- `MSGTOKEN`
- `INITQ('')`
- `INITQ(string)`
- `MAXDEPTH(5000)`
- `MAXDEPTH(integer)`
DEFINE QMODEL

Model q attr:

Notes:
1 Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2 Used only on z/OS.
3 This is the default supplied with WebSphere MQ, but your installation might have changed it.
4 Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
5 This is the default supplied with WebSphere MQ (except on z/OS, where it is 999 999 999), but your installation might have changed it.
DEFINE QREMOTE

Use DEFINE QREMOTE to define a new local definition of a remote queue, a queue manager alias, or a reply-to queue alias, and to set its parameters.

A remote queue is one that is owned by another queue manager that application processes connected to this queue manager need to access.

**Synonym:** DEF QR

**DEFINE QREMOTE**

```
DEFINE QREMOTE(qm-name)
```

- **CMDSCOPE:** (1) Either a single queue manager name or an asterisk (*).
- **QSGDISP:** (2) Options include QSGDISP(QMGR), QSGDISP(COPY), or QSGDISP(GROUP).
- **Define attrs:**
  - **LIKE(name):** NOREPLACE or REPLACE.
  - **DEFPRTY:** DEFPRTY(0) or DEFPRTY(integer).
  - **DEFPSIST:** DEFPSIST(NO) or DEFPSIST(YES).
  - **DESCR:** DSCR('') or DSCR(string).
  - **PUT:** PUT(ENABLED) or PUT(DISABLED).

- **Remote q attrs:**
  - **CLUSNL:** CLUSNL('') or CLUSNL(name).
  - **CLUSTER:** CLUSTER('') or CLUSTER(clustername).
  - **DEFBIND:** DEFBIND(OPEN) or DEFBIND(NOTFIXED).
DEFINE QREMOTE

Notes:
1 Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2 Valid only on z/OS.
3 This is the default supplied with WebSphere MQ, but your installation might have changed it.
4 Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
5 Valid only on Compaq OpenVMS, OS/2 Warp, OS/400, UNIX systems, and Windows.
6 Valid only on Compaq OpenVMS, OS/2 Warp, UNIX systems, and Windows.

Parameter descriptions

The parameter descriptions also apply to the ALTER QUEUE commands, with the following exceptions:

- The FORCE parameter applies only to the ALTER QUEUE commands.
- The LIKE parameter applies only to the DEFINE QUEUE commands.
- The REPLACE and NOREPLACE parameter applies only to the DEFINE QUEUE commands.
- The variations in the CMDSCOPE and QSGDISP parameters between the ALTER QUEUE and DEFINE QUEUE commands are described.

(q-name)
Local name of the queue, except the remote queue where it is the local definition of the remote queue. This is required.

The name must not be the same as any other queue name (of whatever queue type) currently defined on this queue manager (unless REPLACE or ALTER is specified). See “Rules for naming WebSphere MQ objects” on page 4.

BOQNAME(string)
The excessive backout queue name.

This parameter is supported only on local and model queues.

Apart from maintaining a value for this parameter, the queue manager takes no action based on its value.

BOTHRESH(integer)
The backout threshold.

This parameter is supported only on local and model queues.

Apart from maintaining a value for this parameter, the queue manager takes no action based on its value.
Define queues

Specify a value in the range zero through 999,999,999.

**CFSTRUCT**(structure-name)

Specifies the name of the Coupling Facility structure where you want messages stored when you use shared queues.

This parameter is supported only on z/OS for local and model queues.

The name:
- Cannot have more than 12 characters
- Must start with an uppercase letter (A through Z)
- Can include only the characters A through Z and 0 through 9

The name of the queue-sharing group to which the queue manager is connected is prefixed to the name you supply. The name of the queue-sharing group is always four characters, padded with @ symbols if necessary. For example, if you use a queue-sharing group named NY03 and you supply the name PRODUCT7, the resultant Coupling Facility structure name is NY03PRODUCT7. Note that the administrative structure for the queue-sharing group (in this case NY03CSQ_ADMIN) cannot be used for storing messages.

For ALTER QLOCAL, ALTER QMODEL, DEFINE QLOCAL with REPLACE, and DEFINE QMODEL with REPLACE the following rules apply:
- On a local queue with QSGDISP(SHARED), CFSTRUCT cannot change. If you need to change either the CFSTRUCT or QSGDISP value you must delete and redefine the queue. To preserve any of the messages on the queue you must off-load the messages before you delete the queue and reload the messages after you have redefined the queue, or move the messages to another queue.
- On a model queue with DEFTYPE(SHAREDYN), CFSTRUCT cannot be blank.
- On a local queue with a QSGDISP other than SHARED, or a model queue with a DEFTYPE other than SHAREDYN, the value of CFSTRUCT does not matter.

For DEFINE QLOCAL with NOREPLACE and DEFINE QMODEL with NOREPLACE, the Coupling Facility structure:
- On a local queue with QSGDISP(SHARED) or a model queue with a DEFTYPE(SHAREDYN), CFSTRUCT cannot be blank.
- On a local queue with a QSGDISP other than SHARED, or a model queue with a DEFTYPE other than SHAREDYN, the value of CFSTRUCT does not matter.

**Note:** Before you can use the queue, the structure must be defined in the Coupling Facility Resource Management (CFRM) policy data set.

**CLUSNL**(nlname)

The name of the namelist that specifies a list of clusters to which the queue belongs.

This parameter is supported only on alias, local, and remote queues.

Changes to this parameter do not affect instances of the queue that are already open.
Define queues

Only one of the resultant values of CLUSTER or CLUSNL can be nonblank; you cannot specify a value for both.

On local queues, this parameter cannot be set for transmission, SYSTEM.CHANNEL.xx, SYSTEM.CLUSTER.xx, or SYSTEM.COMMAND.xx queues, and on z/OS only, for SYSTEM.QSG.xx queues.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, Solaris, Windows, and z/OS.

**CLUSTER(clustername)**

The name of the cluster to which the queue belongs.

This parameter is supported only on alias, local, and remote queues.

The maximum length is 48 characters conforming to the rules for naming WebSphere MQ objects. Changes to this parameter do not affect instances of the queue that are already open.

Only one of the resultant values of CLUSNL or CLUSTER can be nonblank; you cannot specify a value for both.

On local queues, this parameter cannot be set for transmission, SYSTEM.CHANNEL.xx, SYSTEM.CLUSTER.xx, or SYSTEM.COMMAND.xx queues, and on z/OS only, for SYSTEM.QSG.xx queues.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP or SHARED.

''

The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

*

The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

**DEFBIND**

Specifies the binding to be used when the application specifies MQOO_BIND_AS_Q_DEF on the MQOPEN call, and the queue is a cluster queue.

**OPEN**

The queue handle is bound to a specific instance of the cluster queue when the queue is opened.

**NOTFIXED**

The queue handle is not bound to any particular instance of the
Define queues

cluster queue. This allows the queue manager to select a specific queue instance when the message is put using MQPUT, and to change that selection subsequently should the need arise.

The MQPUT1 call always behaves as if NOTFIXED had been specified.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

DEFPRTY(integer)
The default priority of messages put on the queue. The value must be in the range zero, (the lowest priority) through to the MAXPRTY queue manager parameter. (MAXPRTY is 9.)

DEFPERSIST
Specifies the message persistence to be used when applications specify the MQPER_PERSISTENCE_AS_Q_DEF option.

NO Messages on this queue are lost across a restart of the queue manager. This is the default supplied with WebSphere MQ, but your installation might have changed it.

YES Messages on this queue survive a restart of the queue manager.

On z/OS, N and Y are accepted as synonyms of NO and YES.

DEFSOPT
The default share option for applications opening this queue for input:
EXCL The open request is for exclusive input from the queue
SHARED The open request is for shared input from the queue

DEFTYPE
Queue definition type.

This parameter is supported only on model queues.

PERMDYN
A permanent dynamic queue is created when an application issues an MQOPEN MQI call with the name of this model queue specified in the object descriptor (MQOD).

On z/OS, the dynamic queue has a disposition of QMGR.

SHAREDYN
This option is available on z/OS only.

A permanent dynamic queue is created when an application issues an MQOPEN API call with the name of this model queue specified in the object descriptor (MQOD).

The dynamic queue has a disposition of SHARED.

TEMPDYN
A temporary dynamic queue is created when an application issues an MQOPEN API call with the name of this model queue specified in the object descriptor (MQOD).

On z/OS, the dynamic queue has a disposition of QMGR.

Do not specify this value for a model queue definition with a DEFPERSIST parameter of YES.

If you specify this option, do not specify INDXTYPE(MSGTOKEN).
Define queues

**DESCR** *(string)*
Plain-text comment. It provides descriptive information about the object when an operator issues the DISPLAY QUEUE command.

It should contain only displayable characters. The maximum length is 64 characters. In a DBCS installation, it can contain DBCS characters (subject to a maximum length of 64 bytes).

*Note:* If characters are used that are not in the coded character set identifier (CCSID) for this queue manager, they might be translated incorrectly if the information is sent to another queue manager.

**DISTL**
Whether distribution lists are supported by the partner queue manager.

- **YES** Distribution lists are supported by the partner queue manager.
- **NO** Distribution lists are not supported by the partner queue manager.

*Note:* You should not normally change this parameter, because it is set by the MCA. However you can set this parameter when defining a transmission queue if the distribution list capability of the destination queue manager is known.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.

**FORCE**
This parameter applies only to the ALTER command on alias, local and remote queues.

Specify this parameter to force completion of the command in the following circumstances.

For an **alias** queue, if both of the following are true:
- The TARGQ parameter is specified
- An application has this alias queue open

For a **local** queue, if both of the following are true:
- The NOSHARE parameter is specified
- One or more applications have the queue open for input

FORCE is also needed if both of the following are true:
- The USAGE parameter is changed
- Either one or more messages are on the queue, or one or more applications have the queue open

Do not change the USAGE parameter while there are messages on the queue; the format of messages changes when they are put on a transmission queue.

For a **remote** queue if both of the following are true:
- The XMITQ parameter is changed
- One or more applications has this queue open as a remote queue

FORCE is also needed if both of the following are true:
- Any of the RNAME, RQMNAME, or XMITQ parameters are changed
Define queues

- One or more applications has a queue open that resolved through this definition as a queue manager alias

**Note:** FORCE is not required if this definition is in use as a reply-to queue alias only.

If FORCE is not specified in the circumstances described, the command is unsuccessful.

**GET** Whether applications are to be permitted to get messages from this queue:

**ENABLED**

Messages can be retrieved from the queue (by suitably authorized applications). This is the default supplied with WebSphere MQ, but your installation might have changed it.

**DISABLED**

Applications cannot retrieve messages from the queue.

This parameter can also be changed using the **MQSET** API call.

**HARDENBO** and **NOHARDENBO**

Whether hardening should be used to ensure that the count of the number of times that a message has been backed out is accurate.

This parameter is supported only on local and model queues.

**HARDENBO**

The count is hardened.

**NOHARDENBO**

The count is not hardened. This is the default supplied with WebSphere MQ, but your installation might have changed it.

**INDXTYPE**

The type of index maintained by the queue manager to expedite **MQGET** operations on the queue:

This parameter is supported only on local and model queues.

**NONE**

No index is maintained. Use this when messages are usually retrieved sequentially or use both the message identifier and the correlation identifier as a selection criterion on the **MQGET** call.

**MSGID**

An index of message identifiers is maintained. Use this when messages are usually retrieved using the message identifier as a selection criterion on the **MQGET** call with the correlation identifier set to NULL.

**CORRELID**

An index of correlation identifiers is maintained. Use this when messages are usually retrieved using the correlation identifier as a selection criterion on the **MQGET** call with the message identifier set to NULL.

**GROUPID**

An index of group identifiers is maintained. Use this when messages need to be retrieved using message grouping selection criteria.
**Define queues**

Note: You cannot set INDXTYPE to GROUPID if the queue is a transmission queue.

You can only specify a shared queue with INDXTYPE(GROUPID) if the queue uses a CF structure at CFLEVEL(3).

**MSGTOKEN**

An index of message tokens is maintained. Use this when the queue is a WLM-managed queue that you are using with the Workload Manager functions of z/OS.

Note: You cannot set INDXTYPE to MSGTOKEN if:
- The queue is a model queue with a definition type of SHAREDYN
- The queue is a temporary dynamic queue
- The queue is a transmission queue
- You specify QSGDISP(SHARED)

If you are altering or replacing an existing local queue, you can change the INDXTYPE parameter only in the cases indicated in the following table:
Define queues

<table>
<thead>
<tr>
<th>Queue type</th>
<th>NON-SHARED</th>
<th>SHAREDD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue state</td>
<td>Uncommitted activity</td>
<td>No uncommitted activity, messages present</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ChangeINDXTYPEfrom:</th>
<th>To:</th>
<th>Change allowed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE MSGID</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>NONE CORRELID</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>NONE MSGTOKEN</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>NONE GROUPID</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MSGID NONE</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>MSGID CORRELID</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>MSGID MSGTOKEN</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MSGID GROUPID</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>CORRELID NONE</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>CORRELID MSGID</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>CORRELID MSGTOKEN</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>CORRELID GROUPID</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MSGTOKEN NONE</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>MSGTOKEN MSGID</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>MSGTOKEN CORRELID</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>MSGTOKEN GROUPID</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>GROUPID NONE</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>GROUPID MSGID</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>GROUPID CORRELID</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>GROUPID MSGTOKEN</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

This parameter is supported only on z/OS. On other platforms, retrieval optimization might be provided, but it is not controlled by a queue parameter.

**INITQ(string)**

The local name of a local queue (known as the *initiation queue*) on this queue manager, to which trigger messages relating to this queue are written. See “Rules for naming WebSphere MQ objects” on page 4.

This parameter is supported only on local and model queues.

**LIKE(qtype-name)**

The name of a queue, whose parameters will be used to model this definition.

This parameter applies only to the appropriate DEFINE Queue command.

If this field is not filled in, and you do not complete the parameter fields related to the command, the values are taken from one of the following, depending upon the queue type:
Define queues

<table>
<thead>
<tr>
<th>Queue Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM.DEFAULT.ALIAS.QUEUE</td>
<td>Alias queue</td>
</tr>
<tr>
<td>SYSTEM.DEFAULT.LOCAL.QUEUE</td>
<td>Local queue</td>
</tr>
<tr>
<td>SYSTEM.DEFAULT.MODEL.QUEUE</td>
<td>Model queue</td>
</tr>
<tr>
<td>SYSTEM.DEFAULT.REMOTE.QUEUE</td>
<td>Remote queue</td>
</tr>
</tbody>
</table>

This is equivalent to defining the following object:
LIKE(SYSTEM.DEFAULT.ALIAS.QUEUE)

for an alias queue, and similarly for other queue types.

These default queue definitions can be altered by the installation to the default values required.

On WebSphere MQ for z/OS, the queue manager searches page set zero for an object with the name you specify. If the object is not found, the queue manager then searches the shared repository. The disposition of the LIKE object is not copied to the object you are defining.

Notes:
1. QSGDISP (GROUP) objects are not searched.
2. LIKE is ignored if QSGDISP(COPY) is specified.

**MAXDEPTH(integer)**

The maximum number of messages allowed on the queue.

This parameter is supported only on local and model queues.

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS, specify a value in the range zero through 999 999 999.

On any other WebSphere MQ platform, specify a value in the range zero through 640 000.

Other factors can still cause the queue to be treated as full, for example, if there is no further DASD space available.

If this value is reduced, any messages that are already on the queue that exceed the new maximum remain intact.

**MAXMSGL(integer)**

The maximum length (in bytes) of messages on this queue.

This parameter is supported only on local and model queues.

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows, specify a value in the range zero through to the maximum message length for the queue manager. See the MAXMSGL parameter of the ALTER QMGR command for more information.

On z/OS, specify a value in the range zero through 100 MB. However, if you also specify QSGDISP(SHARED), or DEFTYPE(SHAREDYN), the MAXMSGL must be in the range zero through 64 512 bytes.

On other platforms, specify a value in the range zero through 4 194 304 bytes (4 MB).

For a transmission queue, this value includes the space required for headers. It is recommended that the value should be at least 4000 bytes larger than the maximum expected length of user data in any message that could be put on a transmission queue.
If this value is reduced, any messages that are already on the queue whose length exceeds the new maximum are not affected.

Applications can use this parameter to determine the size of buffer they need to retrieve messages from the queue. Therefore, the value should only be reduced if it is known that this will not cause an application to operate incorrectly.

**MSGDLVSQ**

Message delivery sequence.

This parameter is supported only on local and model queues.

**PRIORITY**

Messages are delivered (in response to **MQGET** API calls) in first-in-first-out (FIFO) order within priority. This is the default supplied with WebSphere MQ, but your installation might have changed it.

**FIFO**

Messages are delivered (in response to **MQGET** API calls) in FIFO order. Priority is ignored for messages on this queue.

If the message delivery sequence is changed from **PRIORITY** to FIFO while there are messages on the queue, the order of the messages already on the queue is not changed. Messages added to the queue subsequently take the default priority of the queue, and so might be processed before some of the existing messages.

If the message delivery sequence is changed from FIFO to **PRIORITY**, the messages put on the queue while the queue was set to FIFO take the default priority.

**Note:** If **INDXTYPE(GROUPID)** is specified with **MSGDLVSQ**(PRIORITY), the priority in which groups are retrieved is based on the priority of the first message within each group. The priorities zero and one are used by the queue manager to optimize the retrieval of messages in logical order, thus the first message in each group should not use these priorities. If it does, the message is stored as if it was priority two.

**PROCESS**(*string*)

The local name of the WebSphere MQ process.

This parameter is supported only on local and model queues.

This is the name of a process instance that identifies the application started by the queue manager when a trigger event occurs. See “Rules for naming WebSphere MQ objects” on page 4.

The process does not have to be defined when the local queue is defined, but it must be available for a trigger event to occur.

If the queue is a transmission queue, the process gives the name of the channel to be started. This parameter is optional for transmission queues on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS; if you do not specify it, the channel name is taken from the value specified for the TRIGDATA parameter.

**PUT**

Whether messages can be put on the queue.

**ENABLED**

Messages can be added to the queue (by suitably authorized
Define queues

applications). This is the default supplied with WebSphere MQ, but your installation might have changed it.

**DISABLED**

Messages cannot be added to the queue.

This parameter can also be changed using the **MQSET** API call.

**QDEPTHHI**(integer)

The threshold against which the queue depth is compared to generate a Queue Depth High event.

This parameter is supported only on local and model queues. For more information about the effect that shared queues on z/OS have on this event, see the **WebSphere MQ Event Monitoring** book.

This event indicates that an application has put a message on a queue, and this has caused the number of messages on the queue to become greater than or equal to the queue depth high threshold. See the **QDPHIEV** parameter.

The value is expressed as a percentage of the maximum queue depth (MAXDEPTH parameter), and must be greater than or equal to zero, and less than or equal to 100.

**QDEPTHLO**(integer)

The threshold against which the queue depth is compared to generate a Queue Depth Low event.

This parameter is supported only on local and model queues. For more information about the effect that shared queues on z/OS have on this event, see the **WebSphere MQ Event Monitoring** book.

This event indicates that an application has retrieved a message from a queue, and this has caused the number of messages on the queue to become less than or equal to the queue depth low threshold. See the **QDPLOEV** parameter.

The value is expressed as a percentage of the maximum queue depth (MAXDEPTH parameter), and must be in the range zero through 100.

**QDPHIEV**

Controls whether Queue Depth High events are generated.

This parameter is supported only on local and model queues.

A Queue Depth High event indicates that an application has put a message on a queue, and this has caused the number of messages on the queue to become greater than or equal to the queue depth high threshold (see the **QDEPTHHI** parameter).

**Note:** The value of this parameter can change implicitly. For more information on this, and the effect that shared queues on z/OS have on this event, see the description of the Queue Depth High event in the **WebSphere MQ Event Monitoring** book.

**ENABLED**

Queue Depth High events are generated

**DISABLED**

Queue Depth High events are not generated

**QDPLOEV**

Controls whether Queue Depth Low events are generated.
This parameter is supported only on local and model queues.

A Queue Depth Low event indicates that an application has retrieved a message from a queue, and this has caused the number of messages on the queue to become less than or equal to the queue depth low threshold (see the QDEPTHLO parameter).

**Note:** The value of this parameter can change implicitly. For more information on this, and the effect that shared queues on z/OS have on this event, see the description of the Queue Depth Low event in the [WebSphere MQ Event Monitoring](#) book.

**ENABLED**
- Queue Depth Low events are generated

**DISABLED**
- Queue Depth Low events are not generated

**QDPMAXEV**
Controls whether Queue Full events are generated.

This parameter is supported only on local and model queues.

A Queue Full event indicates that a put to a queue has been rejected because the queue is full, that is, the queue depth has already reached its maximum value.

**Note:** The value of this parameter can change implicitly. For more information on this, and the effect that shared queues on z/OS have on this event, see the description of the Queue Full event in the [WebSphere MQ Event Monitoring](#) book.

**ENABLED**
- Queue Full events are generated

**DISABLED**
- Queue Full events are not generated

**QSGDISP**
This parameter applies to z/OS only.

Specifies the disposition of the object within the group.

<table>
<thead>
<tr>
<th>QSGDISP</th>
<th>ALTER</th>
<th>DEFINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY</td>
<td>The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(COPY). Any object residing in the shared repository, or any object defined using a command that had the parameters QSGDISP(QMGR), is not affected by this command.</td>
<td>The object is defined on the page set of the queue manager that executes the command using the QSGDISP(GROUP) object of the same name as the 'LIKE' object. For local queues, messages are stored on the page sets of each queue manager and are available only through that queue manager.</td>
</tr>
</tbody>
</table>
Define queues

<table>
<thead>
<tr>
<th>QSGDISP</th>
<th>ALTER</th>
<th>DEFINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP</td>
<td>The object definition resides in the shared repository. The object was defined using a command that had the parameters QSGDISP(GROUP). Any object residing on the page set of the queue manager that executes the command (except a local copy of the object), or any object defined using a command that had the parameters QSGDISP(SHARED), is not affected by this command. If the command is successful, the following command is generated and sent to all active queue managers in the queue-sharing group to refresh local copies on page set zero:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DEFINE QUEUE(q-name) REPLACE QSGDISP(COPY)</td>
<td>The object definition resides in the shared repository. This is allowed only if there is a shared queue manager environment. If the definition is successful, the following command is generated and sent to all active queue managers to make or refresh local copies on page set zero:</td>
</tr>
<tr>
<td></td>
<td>PRIVATE</td>
<td>Not permitted.</td>
</tr>
<tr>
<td>PRIVATE</td>
<td>The object resides on the page set of the queue manager that executes the command, and was defined with QSGDISP(QMGR) or QSGDISP(COPY). Any object residing in the shared repository is unaffected.</td>
<td></td>
</tr>
<tr>
<td>QMGR</td>
<td>The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(QMGR). Any object residing in the shared repository, or any local copy of such an object, is not affected by this command. This is the default value.</td>
<td>The object is defined on the page set of the queue manager that executes the command. This is the default value. For local queues, messages are stored on the page sets of each queue manager and are available only through that queue manager.</td>
</tr>
<tr>
<td>SHARED</td>
<td>This value applies only to local queues. The object definition resides in the shared repository. The object was defined using a command that had the parameters QSGDISP(SHARED). Any object residing on the page set of the queue manager that executes the command, or any object defined using a command that had the parameters QSGDISP(GROUP), is not affected by this command. If the queue is clustered, a command is generated and sent to all active queue managers in the queue-sharing group to notify them of this clustered, shared queue.</td>
<td>This option applies only to local queues. The object is defined in the shared repository. Messages are stored in the Coupling Facility and are available to any queue manager in the queue-sharing group. You can specify SHARED only if:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CFSTRUCT is nonblank</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• INDXTYPE is not MSGTOKEN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The queue is not one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– SYSTEM.CHANNEL.INITQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– SYSTEM.CHANNEL.SEQNO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– SYSTEM.CHANNEL.COMMAND</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– SYSTEM.COMMAND.INPUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the queue is clustered, a command is generated and sent to all active queue managers in the queue-sharing group to notify them of this clustered, shared queue.</td>
</tr>
</tbody>
</table>

**QSVCIEV**

Controls whether Service Interval High or Service Interval OK events are generated.

This parameter is supported only on local and model queues and has no effect if it is specified on a shared queue.
A Service Interval High event is generated when a check indicates that no messages have been retrieved from the queue for at least the time indicated by the QSVCINT parameter.

A Service Interval OK event is generated when a check indicates that messages have been retrieved from the queue within the time indicated by the QSVCINT parameter.

**Note:** The value of this parameter can change implicitly. For more information, see the description of the Service Interval High and Service Interval OK events in the WebSphere MQ Event Monitoring book.

- **HIGH** - Service Interval High events are generated
- **OK** - Service Interval OK events are generated
- **NONE** - No service interval events are generated

**QSVCINT(integer)**

The service interval used for comparison to generate Service Interval High and Service Interval OK events.

This parameter is supported only on local and model queues and has no effect if it is specified on a shared queue.

See the QSVCLIEV parameter.

The value is in units of milliseconds, and must be in the range zero through 999 999 999.

**REPLACE and NOREPLACE**

This option controls whether any existing definition (and on WebSphere MQ for z/OS of the same disposition) is to be replaced with this one. Any object with a different disposition is not changed. The default is NOREPLACE.

**REPLACE**

If the object does exist, the effect is similar to issuing the ALTER command without the FORCE option and with all the other parameters specified. In particular, note that any messages that are on the existing queue are retained.

(The difference between the ALTER command without the FORCE option, and the DEFINE command with the REPLACE option, is that ALTER does not change unspecified parameters, but DEFINE with REPLACE sets all the parameters. When you use REPLACE, unspecified parameters are taken either from the object named on the LIKE option, or from the default definition, and the parameters of the object being replaced, if one exists, are ignored.)

The command fails if both of the following are true:

- The command sets parameters that would require the use of the FORCE option if you were using the ALTER command
- The object is open

The ALTER command with the FORCE option succeeds in this situation.

If SCOPE(CELL) is specified on Compaq OpenVMS, UNIX systems, OS/2 Warp, or Windows, and there is already a queue with the same name in the cell directory, the command fails, whether or not REPLACE is specified.
Define queues

NOREPLACE
The definition should not replace any existing definition of the object.

RETINTVL(integer)
The number of hours from the queue creation date and time (the date and time at which the queue was defined), after which the queue is no longer needed. The value must be in the range zero through 999,999,999.

This parameter is supported only on local and model queues.

The CRDATE and CRTIME can be displayed using the DISPLAY QUEUE command.

This information is available for use by an operator or a housekeeping application to delete queues that are no longer required.

Note: The queue manager does not delete queues based on this value, nor does it prevent queues from being deleted if their retention interval has not expired. It is the user’s responsibility to take any required action.

RNAME(string)
The name of remote queue. This is the local name of the queue as defined on the queue manager specified by RQMNAME.

This parameter is supported only on remote queues.

- If this definition is used for a local definition of a remote queue, RNAME must not be blank when the open occurs.
- If this definition is used for a queue manager alias definition, RNAME must be blank when the open occurs.
- If this definition is used for a reply-to alias, this name is the name of the queue that is to be the reply-to queue.

The name is not checked to ensure that it contains only those characters normally allowed for queue names (see “Rules for naming WebSphere MQ objects” on page 4).

RQMNAME(string)
The name of the remote queue manager on which the queue RNAME is defined.

This parameter is supported only on remote queues.

- If an application opens the local definition of a remote queue, RQMNAME must not be blank or the name of the local queue manager. When the open occurs, if XMITQ is blank there must be a local queue of this name, which is to be used as the transmission queue.
- If this definition is used for a queue manager alias, RQMNAME is the name of the queue manager that is being aliased. It can be the name of the local queue manager. Otherwise, if XMITQ is blank, when the open occurs there must be a local queue of this name, which is to be used as the transmission queue.
- If this definition is used for a reply-to alias, this name is the name of the queue manager that is to be the reply-to queue manager.

The name is not checked to ensure that it contains only those characters normally allowed for WebSphere MQ object names (see “Rules for naming WebSphere MQ objects” on page 4).
Define queues

**SCOPE**

Specifies the scope of the queue definition.

This parameter is supported only on alias, local, and remote queues.

**QMGR**

The queue definition has queue manager scope. This means that the definition of the queue does not extend beyond the queue manager that owns it. To open the queue for output from some other queue manager, either the name of the owning queue manager must be specified, or the other queue manager must have a local definition of the queue.

**CELL**

The queue definition has cell scope. This means that the queue is known to all the queue managers in the cell, and can be opened for output merely by specifying the name of the queue; the name of the queue manager that owns the queue need not be specified.

If there is already a queue with the same name in the cell directory, the command fails. The REPLACE option has no effect on this.

This value is valid only if a name service supporting a cell directory (for example, the supplied DCE name service) has been configured.

This parameter is valid only on Compaq OpenVMS, OS/2 Warp, UNIX systems, and Windows.

**SHARE** and **NOSHARE**

Whether multiple applications can get messages from this queue.

This parameter is supported only on local and model queues.

**SHARE**

More than one application instance can get messages from the queue.

**NOSHARE**

A single application instance only can get messages from the queue.

**STGCLASS***(string)**

The name of the storage class.

This parameter is supported only on local and model queues.

This is an installation-defined name.

This parameter is valid on z/OS only. For more information, see the [WebSphere MQ for z/OS Concepts and Planning Guide](#).

The first character of the name must be uppercase A through Z, and subsequent characters either uppercase A through Z or numeric 0 through 9.

**Note:** You can change this parameter only if the queue is empty and closed.

On platforms other than z/OS, this parameter is ignored.

If you specify QSGDISP(SHARED) or DEFTYPE(SHAREDYN), this parameter is ignored.
Define queues

**TARGQ**(*string*)
The local name of the base queue being aliased. (See “Rules for naming WebSphere MQ objects” on page 4) The maximum length is 48 characters.

This parameter is supported only on alias queues.

This must be one of the following (although this is not checked until the alias queue is opened by an application):

- A local queue (not a model queue)
- A local definition of a remote queue

This queue need not be defined until an application process opens the alias queue.

**TRIGDATA**(*string*)
The data that is inserted in the trigger message. The maximum length of the string is 64 bytes.

This parameter is supported only on local and model queues.

For a transmission queue on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, or z/OS, you can use this parameter to specify the name of the channel to be started.

This parameter can also be changed using the **MQSET** API call.

**TRIGDPTH**(integer)
The number of messages that have to be on the queue before a trigger message is written, if TRIGTYPE is DEPTH. The value must be in the range 1 through 999 999 999.

This parameter is supported only on local and model queues.

This parameter can also be changed using the **MQSET** API call.

**TRIGGER** and **NOTRIGGER**
Whether trigger messages are written to the initiation queue (named by the INITQ parameter) to trigger the application (named by the PROCESS parameter):

**TRIGGER**
Triggering is active, and trigger messages are written to the initiation queue.

**NOTRIGGER**
Triggering is not active, and trigger messages are not written to the initiation queue.

This parameter is supported only on local and model queues.

This is the default supplied with WebSphere MQ, but your installation might have changed it.

This parameter can also be changed using the **MQSET** API call.

**TRIGMPRI**(integer)
The message priority number that triggers this queue. The value must be in the range zero through to the MAXPRTY queue manager parameter (see “DISPLAY QMGR” on page 222 for details).

This parameter can also be changed using the **MQSET** API call.
TRIGTYPE
Whether and under what conditions a trigger message is written to the
initiation queue (named by the INITQ parameter).
This parameter is supported only on local and model queues.
FIRST Whenever the first message of priority equal to or greater than that
specified by the TRIGMPRI parameter of the queue arrives on the
queue.
EVERY Every time a message arrives on the queue with priority equal to
or greater than that specified by the TRIGMPRI parameter of the
queue.
DEPTH When the number of messages with priority equal to or greater
than that specified by TRIGMPRI is equal to the number indicated
by the TRIGDPTH parameter.
NONE No trigger messages are written.
This parameter can also be changed using the MQSET API call.

USAGE
Queue usage.
This parameter is supported only on local and model queues.
NORMAL The queue is not a transmission queue.
XMITQ The queue is a transmission queue, which is used to hold messages
that are destined for a remote queue manager. When an application
puts a message to a remote queue, the message is stored on the
appropriate transmission queue whilst awaiting transmission to the
remote queue manager.
If you specify this option, do not specify values for CLUSTER and
CLUSNL and do not specify INDXTYPE(MSGTOKEN) or
INDXTYPE(GROUPID).

XMITQ(string)
The name of the transmission queue to be used for forwarding messages to
the remote queue, for either a remote queue or for a queue manager alias
definition.
This parameter is supported only on remote queues.
If XMITQ is blank, a queue with the same name as RQMNAME is used as
the transmission queue.
This parameter is ignored if the definition is being used as a queue
manager alias and RQMNAME is the name of the local queue manager.
It is also ignored if the definition is used as a reply-to queue alias
definition.

Usage notes
1. For alias queues:
Usage notes

a. DEFINE QALIAS(otherqname) TARGQ(aliasqueue) CLUSTER(c) has the effect of advertising queue aliasqueue by the name otherqname.
b. DEFINE QALIAS(otherqname) TARGQ(aliasqueue) has the effect of allowing a queue advertised by the name otherqname to be used on this queue manager by the name aliasqueue.

2. For remote queues:
   a. DEFINE QREMOTE(rqueue) RNAME(otherq) RQMNAME(otherqm) CLUSTER(cl) has the effect of advertising this queue manager as a store and forward gateway to which messages for queue rqueue can be sent. It has no effect as a reply-to queue alias, except on the local queue manager.
      DEFINE QREMOTE(otherqm) RNAME() RQMNAME(anotherqm) XMITQ(xq) CLUSTER has the effect of advertising this queue manager as a store and forward gateway to which messages for anotherqm can be sent.
   b. RQMNAME can itself be the name of a cluster queue manager within the cluster, thus (as with QALIAS definitions) you can map the advertised queue manager name to another name locally.
   c. It is possible for the values of RQMNAME and QREMOTE to be the same if RQMNAME is itself a cluster queue manager. If this definition is also advertised using a CLUSTER attribute, do not choose the local queue manager in the cluster workload exit because a cyclic definition will result.
   d. Remote queues do not have to be defined locally. The advantage of doing so is that applications can refer to the queue by a simple, locally-defined name, rather than by one that is qualified by the ID of the queue manager on which the queue resides. This means that applications do not need to be aware of the real location of the queue.
   e. A remote queue definition can also be used as a mechanism for holding a queue manager alias definition, or a reply-to queue alias definition. The name of the definition in these cases is:
      • The queue manager name being used as the alias for another queue manager name (queue manager alias), or
      • The queue name being used as the alias for the reply-to queue (reply-to queue alias).
Use DEFINE STGCLASS to define a storage class to page set mapping.

**Synonym:** DEF STC

**DEFINE STGCLASS**

```
DEFINE STGCLASS(storage-class)
    (1)
    DESCRIPTOR('')
    LIKE(stgclass-name)

    (2)
    CMDSCOPE(' ')
    CMDSCOPE(qmgr-name)
    CMDSCOPE(*)

    (2)
    PSID(integer)

    QSGDISP(QMGR)
    QSGDISP(COPY)
    QSGDISP(GROUP)

    NOREPLACE
    REPLACE

    (1)
    XCFGNAME(' ')
    XCFGNAME(gname)

    XCFMNAME(' ')
    XCFMNAME(mname)
```

**Notes:**
1. This is the default supplied with WebSphere MQ, but your installation might have changed it.
2. Valid only when the queue manager is a member of a queue-sharing group.

**Parameter descriptions**

The parameter descriptions also apply to the ALTER command, with the following exceptions:

- The LIKE parameter applies only to the DEFINE command.
- The NOREPLACE and REPLACE parameter applies only to the DEFINE command.
- The variations in the CMDSCOPE and QSGDISP parameters between the ALTER and DEFINE commands are described.

**(storage-class)**

Name of the storage class. This is required.

This is 1 through 8 characters. The first character is in the range A through Z; subsequent characters are A through Z or 0 through 9.

**Note:** Exceptionally, certain all numeric storage class names are allowed, but are reserved for the use of IBM service personnel.

The storage class must not be the same as any other storage class currently defined on this queue manager.
DEFINE STGCLASS

CMDSCOPE
This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP.

'' The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name other than the queue manager on which it was entered, only if you are using a shared queue environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

DESCR(description)
Plain-text comment. It provides descriptive information about the object when an operator issues the DISPLAY STGCLASS command.

It should contain only displayable characters. The maximum length is 64 characters. In a DBCS installation, it can contain DBCS characters (subject to a maximum length of 64 bytes).

Note: If characters are used that are not in the coded character set identifier (CCSID) for this queue manager, they might be translated incorrectly if the information is sent to another queue manager

LIKE(stgclass-name)
The name of an object of the same type, whose parameters will be used to model this definition.

If this field is not filled in, and you do not complete the parameter fields related to the command, the values are taken from the default definition for this object.

This is equivalent to specifying:
LIKE(SYSTEMST)

This default storage class definition can be altered by your installation to the default values required.

The queue manager searches page set zero for an object with the name you specify. The disposition of the LIKE object is not copied to the object you are defining.

Notes:
1. QSGDISP (GROUP) objects are not searched.
2. LIKE is ignored if QSGDISP(COPY) is specified.
**DEFINE STGCLASS**

**PSID(integer)**

The page set identifier that this storage class is to be associated with. If you do not specify this, the value is taken from the default storage class SYSTEMST.

**Note:** No check is made that the page set has been defined; an error is raised only when you try to put a message to a queue that specifies this storage class (MQRC_PAGESET_ERROR).

The string consists of two numeric characters, in the range 00 through 99. See "DEFINE PSID" on page 124.

**QSGDISP**

Specifies the disposition of the object in the group.

<table>
<thead>
<tr>
<th>QSGDISP</th>
<th>ALTER</th>
<th>DEFINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY</td>
<td>The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(COPY). Any object residing in the shared repository, or any object defined using a command that had the parameters QSGDISP(QMGR), is not affected by this command.</td>
<td>The object is defined on the page set of the queue manager that executes the command using the QSGDISP(GROUP) object of the same name as the 'LIKE' object.</td>
</tr>
<tr>
<td>GROUP</td>
<td>The object definition resides in the shared repository. The object was defined using a command that had the parameters QSGDISP(GROUP). Any object residing on the page set of the queue manager that executes the command (except a local copy of the object) is not affected by this command. If the command is successful, the following command is generated and sent to all active queue managers in the queue-sharing group to refresh local copies on page set zero: DEFINE STGCLASS(storage-class) REPLACE QSGDISP(COPY)</td>
<td>The object definition resides in the shared repository. This is allowed only if the queue manager is in a queue-sharing group. If the definition is successful, the following command is generated and sent to all active queue managers in the queue-sharing group to make or refresh local copies on page set zero: DEFINE STGCLASS(storage-class) REPLACE QSGDISP(COPY)</td>
</tr>
<tr>
<td>PRIVATE</td>
<td>The object resides on the page set of the queue manager that executes the command, and was defined with QSGDISP(QMGR) or QSGDISP(COPY). Any object residing in the shared repository is unaffected.</td>
<td>Not permitted.</td>
</tr>
<tr>
<td>QMGR</td>
<td>The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(QMGR). Any object residing in the shared repository, or any local copy of such an object, is not affected by this command. This is the default value.</td>
<td>The object is defined on the page set of the queue manager that executes the command. This is the default value.</td>
</tr>
</tbody>
</table>

**REPLACE and NOREPLACE**

Whether the existing definition, and with the same disposition, is to be
DEFINE STGCLASS

replaced with this one. This is optional. The default is NOREPLACE. Any object with a different disposition is not changed.

REPLACE
The definition should replace any existing definition of the same name. If a definition does not exist, one is created.

If you use the REPLACE option, all queues that use this storage class must be empty.

NOREPLACE
The definition should not replace any existing definition of the same name.

XCFGNAME(group name)
If you are using the IMS bridge, this is the name of the XCF group to which the IMS system belongs. (This is the group name specified in the IMS parameter list.)

This is 1 through 8 characters. The first character is in the range A through Z; subsequent characters are A through Z or 0 through 9.

XCFMNAME(member name)
If you are using the IMS bridge, this is the XCF member name of the IMS system within the XCF group specified in XCFGNAME. (This is the member name specified in the IMS parameter list.)

This is 1 through 16 characters. The first character is in the range A through Z; subsequent characters are A through Z or 0 through 9.

Usage notes
1. The resultant values of XCFGNAME and XCFMNAME must either both be blank or both be nonblank.
2. You can change a storage class only if it is not being used by any queues. To determine whether any queues are using the storage class, you can use the following command:

   DISPLAY QUEUE(+) STGCLASS(ABC) PSID(n)

   where ‘ABC’ is the name of the storage class, and n is the identifier of the page set that the storage class is associated with.

   This command gives a list of all queues that reference the storage class, and have an active association to page set n, and therefore identifies the queues that are actually preventing the change to the storage class. If you do not specify the PSID, you just get a list of queues that are potentially stopping the change.

   See the DISPLAY QUEUE PSID command on page 236 for more information about active association of a queue to a page set.
Use DELETE AUTHINFO to delete an authentication information object.

**Synonym**: None

**DELETE AUTHINFO**

```
  DELETE AUTHINFO(name)
```

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP.

' '  The command is executed on the queue manager on which it was entered. This is the default value.

```
  CMDSCOPE(qmgr-name)
```

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

`
  CMDSCOPE(*)
```

The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

**QSGDISP**

This parameter applies to z/OS only.

Notes:

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.

**Parameter descriptions**

(name)  Name of the authentication information object. This is required.

The name must be that of an existing authentication information object.

CMDSCOPE

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP.

' '  The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

*  The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.
DELETE AUTHINFO

Specifies the disposition of the object to which you are applying the command (that is, where it is defined and how it behaves).

COPY The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(COPY). Any object residing in the shared repository, or any object defined using a command that had the parameters QSGDISP(QMGR), is not affected by this command.

GROUP The object definition resides in the shared repository. The object was defined using a command that had the parameters QSGDISP(GROUP). Any object residing on the page set of the queue manager that executes the command (except a local copy of the object) is not affected by this command.

If the command is successful, the following command is generated and sent to all active queue managers in the queue-sharing group to delete local copies on page set zero:

```
DELETE AUTHINFO(name) QSGDISP(COPY)
```

QMGR The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(QMGR). Any object residing in the shared repository, or any local copy of such an object, is not affected by this command.

This is the default value.
Use **DELETE CFSTRUCT** to delete a CF application structure definition.

**Synonym**: None

**DELETE CFSTRUCT**

```plaintext
DELETE CFSTRUCT(structure-name)
```

### Keyword and parameter descriptions

**(structure-name)**

The name of the CF structure definition to be deleted. The name must be defined within the queue sharing group.

### Usage notes

1. The command fails if there are any queues in existence that reference this CF structure name that are not both empty and closed.
2. The command cannot specify the CF administration structure (CSQ_ADMIN).
3. The command deletes the DB2 CF structure record only. It does **not** delete the CF structure definition from the CFRM policy data set.
4. CF structures at CFLEVEL(1) are automatically deleted when the last queue on that structure is deleted.
Use DELETE CHANNEL to delete a channel definition.

Notes for z/OS users:
1. This is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, see the WebSphere MQ Intercommunication manual.
2. The command fails if the channel initiator and command server have not been started, or the channel status is RUNNING, except client-connection channels, which can be deleted without the channel initiator or command server running.
3. You can only delete cluster-sender channels that have been created manually.

Parameter descriptions
(channel-name)
The name of the channel definition to be deleted. This is required. The name must be that of an existing channel.

CHLTABLE
Specifies the channel definition table that contains the channel to be deleted. This is optional.

QMGRTBL
The channel table is that associated with the target queue manager. This table does not contain any channels of type CLNTCONN. This is the default.

CLNTTBL
The channel table for CLNTCONN channels. On Compaq OpenVMS, OS/2 Warp, OS/400, Compaq NSK, UNIX systems, and Windows, this is normally associated with a queue manager, but
can be a system-wide, queue manager independent channel table if you set up a number of environment variables. For more information about setting up environment variables, see the WebSphere MQ Clients manual.

On z/OS, this is associated with the target queue manager, but separate from the main channel table.

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group. CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP.

```
```

The command is executed on the queue manager on which it was entered. This is the default value.

`qmgr-name`

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

```
`

The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

**QSGDISP**

This parameter applies to z/OS only.

Specifies the disposition of the object to which you are applying the command (that is, where it is defined and how it behaves).

**COPY**  The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(COPY). Any object residing in the shared repository, or any object defined using a command that had the parameters QSGDISP(QMGR), is not affected by this command.

**GROUP**  The object definition resides in the shared repository. The object was defined using a command that had the parameters QSGDISP(GROUP). Any object residing on the page set of the queue manager that executes the command (except a local copy of the object) is not affected by this command.

If the command is successful, the following command is generated and sent to all active queue managers in the queue-sharing group to delete local copies on page set zero:

```
DELETE CHANNEL(channel-name) QSGDISP(COPY)
```

**QMGR**  The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(QMGR). Any object
residing in the shared repository, or any local copy of such an object, is not affected by this command.
This is the default value.
Use **DELETE NAMELIST** to delete a namelist definition.

**Notes:**
1. On UNIX systems, the command is valid only on AIX, HP-UX, Linux, and Solaris.

**Synonym:** **DELETE NL**

**DELETE NAMELIST**

You must specify which namelist definition you want to delete.

*(name)* The name of the namelist definition to be deleted. The name must be defined to the local queue manager.

If an application has this namelist open, the command fails.

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP.

'' The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group.
DELETE NAMELIST

The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

QSGDISP
This parameter applies to z/OS only.
Specifies the disposition of the object to which you are applying the command (that is, where it is defined and how it behaves).

COPY The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(COPY). Any object residing in the shared repository, or any object defined using a command that had the parameters QSGDISP(QMGR), is not affected by this command.

GROUP The object definition resides in the shared repository. The object was defined using a command that had the parameters QSGDISP(GROUP). Any object residing on the page set of the queue manager that executes the command (except a local copy of the object) is not affected by this command.

If the command is successful, the following command is generated and sent to all active queue managers in the queue-sharing group to delete local copies on page set zero:

DELETE NAMELIST(name) QSGDISP(COPY)

QMGR
The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(QMGR). Any object residing in the shared repository, or any local copy of such an object, is not affected by this command.

This is the default value.
Use DELETE PROCESS to delete a process definition.

**Synonym:** DELETE PRO

**DELETE PROCESS**

```
DELETE PROCESS(process-name) [CMDSCOPE(qmgr-name)] [CMDSCOPE(*)]
```

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.

2. Valid only on z/OS.

**Parameter descriptions**

You must specify which process definition you want to delete.

**(process-name)**

The name of the process definition to be deleted. The name must be defined to the local queue manager.

If an application has this process open, the command fails.

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP.

```
' '
```

The command is executed on the queue manager on which it was entered. This is the default value.

**qmgr-name**

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.
DELETE PROCESS

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

QSGDISP
This parameter applies to z/OS only.
Specifies the disposition of the object to which you are applying the command (that is, where it is defined and how it behaves).

COPY The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(COPY). Any object residing in the shared repository, or any object defined using a command that had the parameters QSGDISP(QMGR), is not affected by this command.

GROUP The object definition resides in the shared repository. The object was defined using a command that had the parameters QSGDISP(GROUP). Any object residing on the page set of the queue manager that executes the command (except a local copy of the object) is not affected by this command.

If the command is successful, the following command is generated and sent to all active queue managers in the queue-sharing group to delete local copies on page set zero:

```
DELETE PROCESS(process-name) QSGDISP(COPY)
```

QMGR
The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(QMGR). Any object residing in the shared repository, or any local copy of such an object, is not affected by this command.

This is the default value.
DELETE queues

This section contains the following commands:

- "DELETE QALIAS"
- "DELETE QLOCAL" on page 166
- "DELETE QMODEL" on page 167
- "DELETE QREMOTE" on page 167

These queues are supported on the following platforms:

<table>
<thead>
<tr>
<th></th>
<th>Compaq NSK</th>
<th>Compaq OpenVMS</th>
<th>OS/400</th>
<th>OS/2 Warp</th>
<th>UNIX systems</th>
<th>Windows</th>
<th>z/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

### DELETE QALIAS

Use DELETE QALIAS to delete an alias queue definition.

**Synonym:** DELETE QA

**DELETE QALIAS**

![Diagram of DELETE QALIAS command]

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group.
   You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.
DELETE QLOCAL

Use DELETE QLOCAL to delete a local queue definition. You can specify that the queue must not be deleted if it contains messages, or that it can be deleted even if it contains messages.

**Synonym:** DELETE QL

### DELETE QLOCAL

```
DELETE QLOCAL(q-name)

CMDSCOPE('',) (2) NOPURGE

CMDSCOPE(qmgr-name) (1)

CMDSCOPE(*)

/PURGE

QSGDISP(QMGR) (2)

QSGDISP(COPY) (1)

QSGDISP(GROUP) (1)

QSGDISP(SHARED)
```

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.
DELETE QMODEL

Use DELETE QMODEL to delete a model queue definition.

**Synonym:** DELETE QM

**DELETE QMODEL**

```plaintext
DELETE QMODEL(q-name)
```

**CMDSCOPE(' ')(2)**

**QSGDISP(QMGR)(2)**

**CMDSCOPE(qmgr-name)(1)**

**QSGDISP(COPY)(1)**

**CMDSCOPE(*) (1)**

**QSGDISP(GROUP)**

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.

DELETE QREMOTE

Use DELETE QREMOTE to delete a local definition of a remote queue. It does not affect the definition of that queue on the remote system.

**Synonym:** DELETE QM

**DELETE QREMOTE**

```plaintext
DELETE QREMOTE(q-name)
```

**CMDSCOPE(' ')(2)**

**QSGDISP(QMGR)(2)**

**CMDSCOPE(qmgr-name)(1)**

**QSGDISP(COPY)(1)**

**CMDSCOPE(*) (1)**

**QSGDISP(GROUP)**

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.

**Parameter descriptions**

**(q-name)**

The name of the queue must be defined to the local queue manager for all the queue types.

For an alias queue this is the local name of the alias queue to be deleted.

For a model queue this is the local name of the model queue to be deleted.

For a remote queue this is the local name of the remote queue to be deleted.
DELETE Queues

For a local queue this is the name of the local queue to be deleted. You must specify which queue you want to delete.

**Note:** A queue cannot be deleted if it contains uncommitted messages.

If an application has this queue open, or has open a queue that eventually resolves to this queue, the command fails. The command also fails if this queue is a transmission queue, and any queue that is, or resolves to, a remote queue that references this transmission queue, is open.

If this queue has a SCOPE attribute of CELL, the entry for the queue is also deleted from the cell directory.

**CMDSCOPE**
This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP or SHARED.

```
'
```
The command is executed on the queue manager on which it was entered. This is the default value.

```
qmgr-name
```
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

```
*
```
The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

**PURGE and NOPURGE**
Specifies whether or not any existing committed messages on the queue named by the DELETE command are to be purged for the delete command to work. The default is NOPURGE.

**PURGE**
The deletion is to go ahead even if there are committed messages on the named queue, and these messages are also to be purged.

**NOPURGE**
The deletion is not to go ahead if there are any committed messages on the named queue.

**QSGDISP**
Specifies the disposition of the object to which you are applying the command (that is, where it is defined and how it behaves). If the object definition is shared, you do not need to delete it on every queue manager that is part of a queue-sharing group. (Queue-sharing groups are available only on WebSphere MQ for z/OS.)

**COPY**
The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(COPY). Any object
residing in the shared repository, or any object defined using a command that had the parameters QSGDISP(QMGR), is not affected by this command.

GROUP
The object definition resides in the shared repository. The object was defined using a command that had the parameters QSGDISP(GROUP). Any object residing on the page set of the queue manager that executes the command, or any object defined using a command that had the parameters QSGDISP(SHARED), is not affected by this command.

If the deletion is successful, the following command is generated and sent to all active queue managers in the queue-sharing group to make, or delete, local copies on page set zero:

```
DELETE queue(q-name) QSGDISP(COPY)
```

or, for a local queue only:

```
DELETE QLOCAL(q-name) NOPURGE QSGDISP(COPY)
```

**Note:** You always get the NOPURGE option even if you specify PURGE. To delete messages on local copies of the queues, you must explicitly issue the command:

```
DELETE QLOCAL(q-name) QSGDISP(COPY) PURGE
```

for each copy.

QMGR
The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(QMGR). Any object residing in the shared repository, or any local copy of such an object, is not affected by this command.

This is the default value.

SHARED
This option applies only to local queues.

The object definition resides in the shared repository. The object was defined using a command that had the parameters QSGDISP(SHARED). Any object residing on the page set of the queue manager that executes the command, or any object defined using a command that had the parameters QSGDISP(GROUP), is not affected by this command.
Use DELETE STGCLASS to delete a storage class definition

**Synonym:** DELETE STC

DELETE STGCLASS

<table>
<thead>
<tr>
<th>Parameter descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>You must specify which storage class definition you want to delete.</td>
</tr>
<tr>
<td>All queues that use the storage class must be empty and closed.</td>
</tr>
<tr>
<td>(name) The name of the storage class definition to be deleted. The name must be defined to the local queue manager.</td>
</tr>
<tr>
<td>The command fails unless all queues referencing the storage class are empty and closed.</td>
</tr>
<tr>
<td><strong>CMDSCOPE</strong></td>
</tr>
<tr>
<td>This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.</td>
</tr>
<tr>
<td>CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP.</td>
</tr>
<tr>
<td>&quot;&quot; The command is executed on the queue manager on which it was entered. This is the default value.</td>
</tr>
<tr>
<td>qmgr-name The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.</td>
</tr>
<tr>
<td>You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.</td>
</tr>
<tr>
<td>&quot;&quot; The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.</td>
</tr>
</tbody>
</table>

Notes:

1. Valid only when the queue manager is a member of a queue-sharing group.
QSGDISP

Specifies the disposition of the object to which you are applying the command (that is, where it is defined and how it behaves).

COPY  The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(COPY). Any object residing in the shared repository, or any object defined using a command that had the parameters QSGDISP(QMGR), is not affected by this command.

GROUP  The object definition resides in the shared repository. The object was defined using a command that had the parameters QSGDISP(GROUP). Any object residing on the page set of the queue manager that executes the command (except a local copy of the object) is not affected by this command.

If the command is successful, the following command is generated and sent to all active queue managers in the queue-sharing group to delete local copies on page set zero:

    DELETE STGCLASS(name) QSGDISP(COPY)

QMGR  The object definition resides on the page set of the queue manager that executes the command. The object was defined using a command that had the parameters QSGDISP(QMGR). Any object residing in the shared repository, or any local copy of such an object, is not affected by this command.

This is the default value.
Use DISPLAY ARCHIVE to display archive information.

Synonym: DIS ARC

Notes:
1. Valid only when the queue manager is a member of a queue-sharing group.

Parameter descriptions

CMDSCOPE
This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

' ' The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

Usage notes
1. DISPLAY ARCHIVE returns a report that shows the initial values for the archiving parameters, and the current values as changed by the SET ARCHIVE command.

   - Units in which primary and secondary space allocations are made (ALCUNIT).
   - Prefix for first archive log data set name (ARCPFX1).
   - Prefix for second archive log data set name (ARCPFX2).
DISPLAY ARCHIVE

- The retention period of the archive log data set in days (ARCRETN).
- List of route codes for messages to the operator about archive log data sets (ARCWRTC).
- Whether to send message to operator and wait for reply before trying to mount an archive log data set (ARCWTOR).
- Block size of archive log data set (BLKSIZE).
- Whether archive log data sets are cataloged in the ICF (CATALOG).
- Whether archive log data sets should be compacted (COMPACT).
- Primary space allocation for DASD data sets (PRIQTY).
- Whether archive log data sets are protected by ESM profiles when the data sets are created (PROTECT).
- Maximum time, in seconds, allowed for quiesce when ARCHIVE LOG with MODE(QUIESCE) specified (QUIESCE).
- Secondary space allocation for DASD data sets. See the ALCUNIT parameter for the units to be used (SECQTY).
- Whether the archive data set name should include a time stamp (TSTAMP).
- Device type or unit name on which the first copy of archive log data sets is stored (UNIT).
- Device type or unit name on which the second copy of archive log data sets is stored (UNIT2).

It also reports the status of tape units used for archiving.

2. This command is issued internally by WebSphere MQ at the end of queue manager startup.
Use DISPLAY AUTHINFO to display the attributes of an authentication information object.

**Synonym**: DIS AUTHINFO

### DISPLAY AUTHINFO

```plaintext
DISPLAY AUTHINFO(generic-authentication-information-object-name) ALL
```

**Requested attrs:**

- ALTDATE
- ALTTIME
- AUTHTYPE
- CONNAME
- DESCR
- LDAPPWD
- LDAPUSER

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.

### Parameter descriptions

**generic-authentication-information-object-name**

The name of the authentication information object to be displayed (see “Rules for naming WebSphere MQ objects” on page 4). A trailing asterisk (*) matches all authentication information objects with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all authentication information objects. The names must all be defined to the local queue manager.
ALL Specify this to display all the parameters. If this parameter is specified, any parameters that are requested specifically have no effect; all parameters are still displayed.

This is the default if you do not specify a generic name and do not request any specific parameters.

CMDSCOPE
This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP.

'' The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

QSGDISP
Specifies the disposition of the objects for which information is to be displayed. Values are:

LIVE This is the default value and displays information for objects defined with QSGDISP(QMGR) or QSGDISP(COPY).

ALL Displays information for objects defined with QSGDISP(QMGR) or QSGDISP(COPY).

If there is a shared queue manager environment, and the command is being executed on the queue manager where it was issued, this option also displays information for objects defined with QSGDISP(GROUP).

If QSGDISP(LIVE) is specified or defaulted, or if QSGDISP(ALL) is specified in a shared queue manager environment, the command might give duplicated names (with different dispositions).

COPY Display information only for objects defined with QSGDISP(COPY).

GROUP Display information only for objects defined with QSGDISP(GROUP). This is allowed only if there is a shared queue manager environment.

PRIVATE Display information for objects defined with
DISPLAY AUTHINFO

QSGDISP(QMGR) or QSGDISP(COPY). Note that QSGDISP(PRIVATE) displays the same information as QSGDISP(LIVE).

QMGR Display information only for objects defined with QSGDISP(QMGR).

QSGDISP displays one of the following values if it is specified, or if there is a shared queue manager environment:

QMGR The object was defined with QSGDISP(QMGR).
GROUP The object was defined with QSGDISP(GROUP).
COPY The object was defined with QSGDISP(COPY).

Requested parameters
Specify one or more parameters that define the data to be displayed. The parameters can be specified in any order, but do not specify the same parameter more than once.

The default, if no parameters are specified (and the ALL parameter is not specified) is that the object names and their AUTHTYPEs are displayed.

ALTDATE The date on which the definition was last altered, in the form yyyy-mm-dd
ALTTIME The time at which the definition was last altered, in the form hh.mm.ss
AUTHTYPE The type of the authentication information
CONNAME The DNS name or IP address of the host on which the LDAP server is running
DESCR Description of the authentication information object
LDAPPWD Password associated with the Distinguished Name of the user on the LDAP server, displayed as xxxxxxxx on all platforms except z/OS.
LDAPUSER Distinguished Name of the user on the LDAP server

See “DEFINE AUTHINFO” on page 62 for more information about individual parameters.
DISPLAY CFSTATUS

Use DISPLAY CFSTATUS to display the status of one or more CF application structures.

**Synonym:** DIS CFSTATUS

**DISPLAY CFSTATUS**

`DISPLAY CFSTATUS(generic-structure-name) [TYPE(SUMMARY)] [TYPE(CONNECT)] [TYPE(BACKUP)]`

**Keyword and parameter descriptions**

The name of the application structure for the status information to be displayed must be specified. This can be a specific application structure name or a generic name. By using a generic name, it is possible to display either:

- status information for all application structure definitions
- status information for one or more application structures that match the specified name

The type of status information to be returned can also be specified. This can be:

- summary status information for the application structure in the queue-sharing group
- connection status information for each queue manager in the queue-sharing group for each matching application structure name
- backup status information for each backup taken for each matching application structure defined in the queue-sharing group

(generic-structure-name)

The 12-character name of the CF application structure to be displayed. A trailing asterisk (*) matches all structure names with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all structure names.

The CF structure name must be defined within the queue-sharing group.

The CFSTATUS generic name can be the administration CF structure name (CSQ_ADMIN) or any generic form of this name. Data for this structure, however, is only displayed when TYPE is set to SUMMARY.

**TYPE**

Specifies the type of status information required to be displayed. Values are:

**SUMMARY**

Display summary status information for the application structure. This is the default.

**CONNECT**

Display connection status information for each application structure for each active queue manager.
DISPLAY CFSTATUS

<table>
<thead>
<tr>
<th>BACKUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display backup status information for each application structure.</td>
</tr>
</tbody>
</table>

Summary status
For summary status, the following information is returned for each structure that satisfies the selection criteria:

- The name of the application structure matching the generic name.
- The type of information returned.

CFTYPE
The CF structure type. This is one of the following:

ADMIN
This is the CF administration structure.

APPL
This is a CF application structure.

STATUS
The status of the CF application structure. This is one of the following:

ACTIVE
The structure is active.

FAILED
The structure has failed.

NOTFOUND
The structure is not allocated in the CF, but has been defined to DB2.

INBACKUP
The structure is in the process of being backed-up.

INRECOVER
The structure is in the process of being recovered.

SIZEMAX(size)
The size in kilobytes of the application structure.

SIZEUSED(integer)
The percentage of the size of the application structure that is in use. Therefore SIZEUSED(25) would indicate that a quarter of the space allocated to this application structure is in use.

ENTSMAX(integer)
The number of CF list entries defined for this application structure.

ENTSUSED(integer)
The number of CF list entries for this application structure that are in use.

FAILTIME(time)
The time that this application structure failed. The format of this field is hh.mm.ss. This parameter is only applicable when the CF structure is in FAILED or INRECOVER state. If the structure is not in a failed state, this is displayed as FAILTIME().

FAILDATE(date)
The date that this application-structure failed. The format of this field is yyyy-mm-dd. This parameter is only applicable when the CF structure is in FAILED or INRECOVER state. If the structure is not in a failed state, then this is displayed as FAILDATE().
Connection status
For connection status, the following information is returned for each connection to each structure that satisfies the selection criteria:

- The name of the application structure matching the generic name.
- The type of information returned.

**QMNAME**(qmgrname)
The queue manager name.

**SYSNAME**(systemname)
The name of the z/OS image of the queue manager that last connected to the application structure. These can be different across queue managers depending on the customer configuration setup.

**STATUS**
A status indicating whether or not this queue manager is connected to this application structure. This is one of the following:

- **ACTIVE**
  The structure is connected to this queue manager.

- **FAILED**
  The queue manager connection to this structure has failed.

- **NONE**
  The structure has never been connected to this queue manager.

**FAILTIME**(time)
The time that this queue manager lost connectivity to this application structure. The format of this field is **hh.mm.ss**. This parameter is only applicable when the CF structure is in FAILED state. If the structure is not in a failed state, this is displayed as **FAILTIME()**.

**FAILDATE**(date)
The date that this queue manager lost connectivity to this application structure. The format of this field is **yyyy-mm-dd**. This parameter is only applicable when the CF structure is in FAILED state. If the structure is not in a failed state, this is displayed as **FAILDATE()**.

Backup status
For backup status, the following information is returned for each structure that satisfies the selection criteria:

- The name of the application structure matching the generic name.
- The type of information returned.

**STATUS**
The status of the CF application structure. This is one of the following:

- **ACTIVE**
  The structure is active.

- **FAILED**
  The structure has failed.

- **NONE**
  The structure is defined as RECOVER(YES), but has never been backed up.

- **INBACKUP**
  The structure is in the process of being backed-up.

- **INRECOVER**
  The structure is in the process of being recovered.
The name of the queue manager that took the last successful backup for this application structure.

The end time of the last successful backup taken for this application structure. The format of this field is hh:mm:ss.

The date of the last successful backup taken for this application structure. The format of this field is yyyy-mm-dd.

The size in megabytes of the last successful backup taken for this application structure.

This is the backup dataset start RBA for the start of the last successful backup taken for this application structure.

This is the backup dataset end RBA for the end of the last successful backup taken for this application structure.

The time that this CF structure failed. The format of this field is hh:mm:ss. This parameter is only applicable when the CF structure is in FAILED state. If the structure is not in a failed state, this is displayed as FAILTIME().

The date that this CF structure failed. The format of this field is yyyy-mm-dd. This parameter is only applicable when the CF structure is in FAILED state. If the structure is not in a failed state, this is displayed as FAILDATE().
DISPLAY CFSTRUCT

Use DISPLAY CFSTRUCT to display the attributes of one or more CF application structures.

**Synonym:** DIS CFSTRUCT

**DISPLAY CFSTRUCT**

```
>>>DISPLAY CFSTRUCT(generic-structure-name) [ALL] [requested attrs]
```

**Requested attrs:**

```
ALTDATE
ALTTIME
CFLEVEL
DESCR
RECOVER
```

**Keyword and parameter descriptions**

The name of the application structure to be displayed must be specified. This can be a specific application structure name or a generic name. By using a generic name, it is possible to display either:

- all application structure definitions
- one or more application structures that match the specified name

```
(generic-structure-name)
```

The 12-character name of the CF application structure to be displayed. A trailing asterisk (*) matches all structure names with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all structure names.

The CF structure name must be defined within the queue-sharing group.

```
ALL
```

Specify this to display all attributes. If this keyword is specified, any attributes that are requested specifically have no effect; all attributes are still displayed.

**Requested parameters**

Specify one or more attributes that define the data to be displayed. The attributes can be specified in any order. Do not specify the same attribute more than once.

The default, if no parameters are specified (and the ALL parameter is not specified) is that the structure names are displayed.

```
ALTDATE
```

The date on which the definition was last altered, in the form yyyy-mm-dd.
DISPLAY CFSTRUCT

<table>
<thead>
<tr>
<th></th>
<th>ALTTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The time at which the definition was last altered, in the form hh:mm:ss.</td>
</tr>
<tr>
<td></td>
<td>CFLEVEL</td>
</tr>
<tr>
<td></td>
<td>Indicates the functional capability level for this CF application structure.</td>
</tr>
<tr>
<td></td>
<td>DESCR</td>
</tr>
<tr>
<td></td>
<td>Descriptive comment.</td>
</tr>
<tr>
<td></td>
<td>RECOVER</td>
</tr>
<tr>
<td></td>
<td>Indicates whether CF recovery for the application structure is supported.</td>
</tr>
<tr>
<td></td>
<td>Values are:</td>
</tr>
<tr>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>CF application structure recovery is not supported.</td>
</tr>
<tr>
<td></td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>CF application structure recovery is supported.</td>
</tr>
</tbody>
</table>

**Usage notes**

1. The command cannot specify the CF administration structure (CSQ_ADMIN).
DISPLAY CHANNEL

Use DISPLAY CHANNEL to display a channel definition.

Notes:
1. On z/OS, this is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, see the [WebSphere MQ Intercommunication](#) manual.
2. You can only display cluster-sender channels if they were created manually.

Synonym: DIS CHL

DISPLAY CHANNEL

Requested attrs:
<table>
<thead>
<tr>
<th>DISPLAY CHANNEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTDATE</td>
</tr>
<tr>
<td>ALTTIME</td>
</tr>
<tr>
<td>AUTOSTART</td>
</tr>
<tr>
<td>BATCHID</td>
</tr>
<tr>
<td>BATCHINT</td>
</tr>
<tr>
<td>BATCHSZ</td>
</tr>
<tr>
<td>CHLTYPE</td>
</tr>
<tr>
<td>CLUSNL</td>
</tr>
<tr>
<td>CLUSTER</td>
</tr>
<tr>
<td>CONNAME</td>
</tr>
<tr>
<td>CONVERT</td>
</tr>
<tr>
<td>DESCR</td>
</tr>
<tr>
<td>DISCINT</td>
</tr>
<tr>
<td>HBINT</td>
</tr>
<tr>
<td>KAINT</td>
</tr>
<tr>
<td>LOCLADDR</td>
</tr>
<tr>
<td>LONGRTY</td>
</tr>
<tr>
<td>LONGTMR</td>
</tr>
<tr>
<td>MAXMSGL</td>
</tr>
<tr>
<td>MCANAME</td>
</tr>
<tr>
<td>MCTYPE</td>
</tr>
<tr>
<td>MCAUSER</td>
</tr>
<tr>
<td>MODEMNAME</td>
</tr>
<tr>
<td>MRDATA</td>
</tr>
<tr>
<td>MREXIT</td>
</tr>
<tr>
<td>MBRTY</td>
</tr>
<tr>
<td>MRTRTY</td>
</tr>
<tr>
<td>MSGDATA</td>
</tr>
<tr>
<td>MSGEXIT</td>
</tr>
<tr>
<td>NETPRTY</td>
</tr>
<tr>
<td>NPMSPEED</td>
</tr>
<tr>
<td>PASSWORD</td>
</tr>
<tr>
<td>PUTAUT</td>
</tr>
<tr>
<td>QNAME</td>
</tr>
<tr>
<td>RCVDATA</td>
</tr>
<tr>
<td>RCVEXIT</td>
</tr>
<tr>
<td>SCYDATA</td>
</tr>
<tr>
<td>SCYEXIT</td>
</tr>
<tr>
<td>SENDDATA</td>
</tr>
<tr>
<td>SENDEXIT</td>
</tr>
<tr>
<td>SEQGRAP</td>
</tr>
<tr>
<td>SHORTRTY</td>
</tr>
<tr>
<td>SHORTTMR</td>
</tr>
<tr>
<td>SSLCAUTH</td>
</tr>
<tr>
<td>SSLCIPHER</td>
</tr>
<tr>
<td>SSLPEER</td>
</tr>
<tr>
<td>TPNAME</td>
</tr>
<tr>
<td>TRPTYPE</td>
</tr>
<tr>
<td>USERID</td>
</tr>
<tr>
<td>XMITQ</td>
</tr>
</tbody>
</table>
DISPLAY CHANNEL

Notes:

1 Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.

2 Valid only on z/OS.

3 Valid only on AIX, Compaq OpenVMS, HP-UX, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

4 Valid only on Compaq NSK.

5 Not valid on z/OS.

6 Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.

Parameter descriptions

You must specify the name of the channel definition you want to display. This can be a specific channel name or a generic channel name. By using a generic channel name, you can display either:

• All channel definitions
• One or more channel definitions that match the specified name

            (generic-channel-name)

The name of the channel definition to be displayed (see “Rules for naming WebSphere MQ objects” on page 4). A trailing asterisk (*) matches all channel definitions with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all channel definitions. The names must all be defined to the local queue manager.

ALL Specify this to display all the parameters. If this parameter is specified, any parameters that are also requested specifically have no effect; all parameters are still displayed.

On AIX, Compaq OpenVMS, HP-UX, OS/2 Warp, OS/400, Solaris, Windows, and z/OS, this is the default if you do not specify a generic name and do not request any specific parameters.

If no parameters are specified (and the ALL parameter is not specified or defaulted), the default is that the channel names only are displayed. On z/OS, the CHLTYPE is also displayed.

CMDSCOPE

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

'' The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also
DISPLAY CHANNEL

passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

QSGDISP

Specifies the disposition of the objects for which information is to be displayed. Values are:

- **LIVE**
  This is the default value and displays information for objects defined with QSGDISP(QMGR) or QSGDISP(COPY).

- **ALL**
  Displays information for objects defined with QSGDISP(QMGR) or QSGDISP(COPY).
  
  If there is a shared queue manager environment, and the command is being executed on the queue manager where it was issued, this option also displays information for objects defined with QSGDISP(GROUP).

  If QSGDISP(ALL) is specified in a shared queue manager environment, the command might give duplicated names (with different dispositions).

  **Note:** In the QSGDISP(LIVE) case, this occurs only where a shared and a non-shared queue have the same name; such a situation should not occur in a well-managed system.

In a shared queue manager environment, use

```
DISPLAY CHANNEL(name) CMDSCOPE(*) QSGDISP(ALL)
```

to list ALL objects matching name

in the queue-sharing group without duplicating those in the shared repository.

- **COPY**
  Display information only for objects defined with QSGDISP(COPY).

- **GROUP**
  Display information only for objects defined with QSGDISP(GROUP). This is allowed only if there is a shared queue manager environment.

- **PRIVATE**
  Display information only for objects defined with QSGDISP(QMGR) or QSGDISP(COPY). Note that QSGDISP(PRIVATE) displays the same information as QSGDISP(LIVE).

- **QMGR**
  Display information only for objects defined with QSGDISP(QMGR).

QSGDISP displays one of the following values if it is specified, or if there is a shared queue manager environment:

- **QMGR**
  The object was defined with QSGDISP(QMGR).

- **GROUP**
  The object was defined with QSGDISP(GROUP).

- **COPY**
  The object was defined with QSGDISP(COPY).
TYPE  This is optional. It can be used to restrict the display to channels of one type.

The value is one of the following:

- **ALL** Channels of all types are displayed (this is the default).
- **SDR** Sender channels only are displayed.
- **SVR** Server channels only are displayed.
- **RCVR** Receiver channels only are displayed.
- **RQSTR** Requester channels only are displayed.
- **CLNCON** Client-connection channels only are displayed.
- **SVRCONN** Server-connection channels only are displayed.
- **CLUSSDR** Cluster-sender channels only are displayed (valid on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS only).
- **CLUSRCVR** Cluster-receiver channels only are displayed (valid on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS only).

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows, CHLTYPE(type) can be used as a synonym for this parameter.

**Requested parameters**

Specify one or more parameters that define the data to be displayed. You can specify the parameters in any order, but do not specify the same parameter more than once.

Some parameters are relevant only for channels of a particular type or types. Attributes that are not relevant for a particular type of channel cause no output, nor is an error raised.

- **ALTDATE** The date on which the definition was last altered, in the form `yyyy-mm-dd`.
- **ALTTIME** The time at which the definition was last altered, in the form `hh.mm.ss`.
- **AUTOSTART** Whether an LU 6.2 responder process should be started for the channel.
- **BATCHHB** The batch heartbeating value being used.
- **BATCHINT** Minimum batch duration.
- **BATCHSZ** Batch size.
- **CHLTYPE** Channel type.

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows the channel type is always displayed if you specify a generic channel name and do not request any other parameters. On z/OS, the channel type is always displayed.
## DISPLAY CHANNEL

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 WarpOS/2 Warp, OS/400, Solaris, and Windows, `TYPE(type)` can be used as a synonym for this parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLUSTER</td>
<td>The name of the cluster to which the channel belongs.</td>
</tr>
<tr>
<td>CLUSNL</td>
<td>The name of the namelist that specifies the list of clusters to which the channel belongs.</td>
</tr>
<tr>
<td>CONNAME</td>
<td>Connection name.</td>
</tr>
<tr>
<td>CONVERT</td>
<td>Whether sender should convert application message data.</td>
</tr>
<tr>
<td>DESCR</td>
<td>Description.</td>
</tr>
<tr>
<td>DISCINT</td>
<td>Disconnection interval.</td>
</tr>
<tr>
<td>HBINT</td>
<td>Heartbeat interval.</td>
</tr>
<tr>
<td>KAINTE</td>
<td>KeepAlive timing for the channel.</td>
</tr>
<tr>
<td>LOCLADDR</td>
<td>Local communications address for the channel.</td>
</tr>
<tr>
<td>LONGRTY</td>
<td>Long retry count.</td>
</tr>
<tr>
<td>LONGTMR</td>
<td>Long retry timer.</td>
</tr>
<tr>
<td>MAXMSGL</td>
<td>Maximum message length for channel.</td>
</tr>
<tr>
<td>MCANAME</td>
<td>Message channel agent name.</td>
</tr>
<tr>
<td>MCATYPE</td>
<td>Whether message channel agent runs as a separate process or a separate thread.</td>
</tr>
<tr>
<td>MCAUSER</td>
<td>Message channel agent user identifier.</td>
</tr>
<tr>
<td>MODENAME</td>
<td>LU 6.2 mode name.</td>
</tr>
<tr>
<td>MRDATA</td>
<td>Channel message-retry exit user data.</td>
</tr>
<tr>
<td>MREXIT</td>
<td>Channel message-retry exit name.</td>
</tr>
<tr>
<td>MRRTY</td>
<td>Channel message-retry exit retry count.</td>
</tr>
<tr>
<td>MRTMR</td>
<td>Channel message-retry exit retry time.</td>
</tr>
<tr>
<td>MSGDATA</td>
<td>Channel message exit user data.</td>
</tr>
<tr>
<td>MSGEXIT</td>
<td>Channel message exit names.</td>
</tr>
<tr>
<td>NETPRTY</td>
<td>The priority for the network connection.</td>
</tr>
<tr>
<td>NPMSPEED</td>
<td>Nonpersistent message speed.</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>Password for initiating LU 6.2 session (if nonblank, this is displayed as asterisks on all platforms except z/OS).</td>
</tr>
<tr>
<td>PUTAUT</td>
<td>Put authority.</td>
</tr>
<tr>
<td>QMNAME</td>
<td>Queue manager name.</td>
</tr>
<tr>
<td>RCVDATA</td>
<td>Channel receive exit user data.</td>
</tr>
<tr>
<td>RCVEXIT</td>
<td>Channel receive exit names.</td>
</tr>
<tr>
<td>SCYDATA</td>
<td>Channel security exit user data.</td>
</tr>
<tr>
<td>SCYEXIT</td>
<td>Channel security exit names.</td>
</tr>
<tr>
<td>SENDDATA</td>
<td>Channel send exit user data.</td>
</tr>
</tbody>
</table>
### DISPLAY CHANNEL

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SENEXIT</td>
<td>Channel send exit names.</td>
</tr>
<tr>
<td>SEQWRAP</td>
<td>Sequence number wrap value.</td>
</tr>
<tr>
<td>SHORTRTRY</td>
<td>Short retry count.</td>
</tr>
<tr>
<td>SHORTTMR</td>
<td>Short retry timer.</td>
</tr>
<tr>
<td>SSLCAUTH</td>
<td>Whether SSL client authentication is required.</td>
</tr>
<tr>
<td>SSLCIPH</td>
<td>Cipher specification for the SSL connection.</td>
</tr>
<tr>
<td>SSLPEER</td>
<td>Filter for the Distinguished Name from the certificate of the peer queue manager or client at the other end of the channel.</td>
</tr>
<tr>
<td>TPNAME</td>
<td>LU 6.2 transaction program name.</td>
</tr>
<tr>
<td>TRPTYPE</td>
<td>Transport type.</td>
</tr>
<tr>
<td>USERID</td>
<td>User identifier for initiating LU 6.2 session.</td>
</tr>
<tr>
<td>XMITQ</td>
<td>Transmission queue name.</td>
</tr>
</tbody>
</table>
DISPLAY CHSTATUS

Use DISPLAY CHSTATUS to display the status of one or more channels.

Note: On z/OS:
1. This is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, see the WebSphere MQ Intercommunication manual.
2. The command fails if the channel initiator has not been started.
3. The command server must be running.

Synonym: DIS CHS

DISPLAY CHSTATUS

Common status:

Current-only status:
DISPLAY CHSTATUS

Parameter descriptions

You must specify the name of the channel for which you want to display status information. This can be a specific channel name or a generic channel name. By using a generic channel name, you can display either:

- Status information for all channels, or
- Status information for one or more channels that match the specified name.

You can also specify whether you want:

- The current status data (of current channels only), or
- The saved status data of all channels.

Notes:

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.
3. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
4. Ignored if specified on z/OS.
5. Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
DISPLAY CHSTATUS

Status for all channels that meet the selection criteria is given, whether the channels were defined manually or automatically.

Before explaining the syntax and options for this command, it is necessary to describe the format of the status data that is available for channels and the states that channels can have.

There are three classes of data available for channel status. These are saved, current, and (on z/OS only) short.

The status fields available for saved data are a subset of the fields available for current data and are called common status fields. Note that although the common data fields are the same, the data values might be different for saved and current status. The rest of the fields available for current data are called current-only status fields.

- **Saved** data consists of the common status fields noted in the syntax diagram. This data is reset at the following times:
  - For all channels:
    - When the channel enters or leaves STOPPED or RETRY state
    - On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows, when the queue manager is ended
  - For a sending channel:
    - Before requesting confirmation that a batch of messages has been received
    - When confirmation has been received
  - For a receiving channel:
    - Just before confirming that a batch of messages has been received
  - For a server connection channel:
    - No data is saved

Therefore, a channel that has never been current cannot have any saved status.

**Note:** Status is not saved until a persistent message is transmitted across a channel, or a nonpersistent message is transmitted with a NPMITESPEED of NORMAL. Because status is saved at the end of each batch, a channel does not have any saved status until at least one batch has been transmitted.

- **Current** data consists of the common status fields and current-only status fields as noted in the syntax diagram. The data fields are continually updated as messages are sent/received.

- **Short** data consists of the STATUS current data item and the short status field as noted in the syntax diagram.

This method of operation has the following consequences:

- An inactive channel might not have any saved status – if it has never been current or has not yet reached a point where saved status is reset.
- The “common” data fields might have different values for saved and current status.
- A current channel always has current status and might have saved status.

Channels can be current or inactive:

**Current channels**

These are channels that have been started, or on which a client has connected, and that have not finished or disconnected normally. They might not yet have reached the point of transferring messages, or data, or
even of establishing contact with the partner. Current channels have current status and might also have saved status.

The term Active is used to describe the set of current channels that are not stopped.

Inactive channels
These are channels that either:
• Have not been started
• On which a client has not connected
• Have finished
• Have disconnected normally

(Note that if a channel is stopped, it is not yet considered to have finished normally – and is, therefore, still current.) Inactive channels have either saved status or no status at all.

There can be more than one instance of the same named receiver, requester, cluster-receiver, or server-connection channel current at the same time (the requester is acting as a receiver). This occurs if several senders, at different queue managers, each initiate a session with this receiver, using the same channel name. For channels of other types, there can only be one instance current at any time.

For all channel types, however, there can be more than one set of saved status information available for a given channel name. At most one of these sets relates to a current instance of the channel, the rest relate to previously-current instances. Multiple instances arise if different transmission queue names or connection names have been used in connection with the same channel. This can happen in the following cases:
• At a sender or server:
  – If the same channel has been connected to by different requesters (servers only)
  – If the transmission queue name has been changed in the definition
  – If the connection name has been changed in the definition
• At a receiver or requester:
  – If the same channel has been connected to by different senders or servers
  – If the connection name has been changed in the definition (for requester channels initiating connection)

The number of sets that are displayed for a given channel can be limited by using the XMITQ, CONNAME, and CURRENT parameters on the command.

(generic-channel-name)
The name of the channel definition for which status information is to be displayed. A trailing asterisk (*) matches all channel definitions with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all channel definitions. The channels must all be defined to the local queue manager.

XMITQ(q-name)
The name of the transmission queue for which status information is to be displayed, for the specified channel or channels.

This parameter can be used to limit the number of sets of status information that is displayed. If it is not specified, the display is not limited in this way.
DISPLAY CHSTATUS

CHLDISP
This parameter applies to z/OS only and specifies the disposition of the
channels for which information is to be displayed, as used in the START
and STOP CHANNEL commands, and not that set by QSGDISP for the
channel definition. Values are:

ALL This is the default value and displays requested status
information for private channels.
   If there is a shared queue manager environment and the
   command is being executed on the queue manager where
   it was issued, or if CURRENT is specified, this option also
displays the requested status information for shared
channels.

PRIVATE Display requested status information for private channels.

SHARED Display requested status information for shared channels.
   This is allowed only if there is a shared queue manager
   environment, and either:
   • CMDSCOPE is blank or the local queue manager
   • CURRENT is specified

CHLDISP displays the following values:
PRIVATE The status is for a private channel.
SHARED The status is for a shared channel.

CMDSCOPE
This parameter applies to z/OS only and specifies how the command is
executed when the queue manager is a member of a queue-sharing group.

'' The command is executed on the queue manager on which
it was entered. This is the default value.

qmgr-name The command is executed on the queue manager you
specify, providing the queue manager is active within the
queue-sharing group.
   You can specify a queue manager name, other than the
queue manager on which it was entered, only if you are
using a queue-sharing group environment and if the
command server is enabled.

* The command is executed on the local queue manager and
is also passed to every active queue manager in the
queue-sharing group. The effect of this is the same as
entering the command on every queue manager in the
queue-sharing group.

Note: See Table 4 on page 200, Table 5 on page 200 and Table 6 on page 200
for the permitted combinations of CHLDISP and CMDSCOPE.

CONNAME(connection-name)
The connection name for which status information is to be displayed, for
the specified channel or channels.
   This parameter can be used to limit the number of sets of status
information that is displayed. If it is not specified, the display is not
limited in this way.
   The value returned for CONNAME might not be the same as in the
channel definition, and might differ between the current channel status
and the saved channel status. (Using CONNAME for limiting the number of sets of status is therefore not recommended.)

For example, if CONNAME:

• Is blank in the channel definition or, when using TCP, is in “host name” format, the channel status value has the resolved IP address.
• Includes the port number, again when using TCP, the current channel status value includes the port number, but the saved channel status value does not.

For SAVED or SHORT status, this value could also be the queue manager name, or queue-sharing group name, of the remote system.

**CURRENT**
This is the default, and indicates that current status information as held by the channel initiator for current channels only is to be displayed.

Both common and current-only status information can be requested for current channels.

Short status information is not displayed if this parameter is specified.

**SAVED**
Specify this to display saved status information for both current and inactive channels.

Only common status information can be displayed. Short and current-only status information is not displayed for current channels if this parameter is specified. On z/OS, the STATUS item is not displayed.

**SHORT**
This indicates that short status information and the STATUS item for current channels only is to be displayed.

Other common status and current-only status information is not displayed for current channels if this parameter is specified.

**ALL**
Specify this to display all the status information for each relevant instance.

If SAVED is specified, this causes only common status information to be displayed, not current-only status information.

If this parameter is specified, any parameters requesting specific status information that are also specified have no effect; all the information is displayed.

The following information is always returned, for each set of status information:

• The channel name
• The transmission queue name (for sender and server channels)
• The connection name
• The remote queue-manager, or queue-sharing group, name (only for current status)
• The type of status information returned (CURRENT, SAVED, or on z/OS only, SHORT)
• STATUS (except SAVED on z/OS)
• On z/OS, CHLDISP

If no parameters requesting specific status information are specified (and the ALL parameter is not specified), no further information is returned.
If status information is requested that is not relevant for the particular channel type, this is not an error.

Common status
The following information applies to all sets of channel status, whether or not the set is current. The information applies to all channel types except server-connection.

CURLUWID
The logical unit of work identifier associated with the current batch, for a sending or a receiving channel.

For a sending channel, when the channel is in doubt it is the LUWID of the in-doubt batch.

For a saved channel instance, this parameter has meaningful information only if the channel instance is in doubt. However, the parameter value is still returned when requested, even if the channel instance is not in doubt.

It is updated with the LUWID of the next batch when this is known.

CURMSGS
For a sending channel, this is the number of messages that have been sent in the current batch. It is incremented as each message is sent, and when the channel becomes in doubt it is the number of messages that are in doubt.

For a saved channel instance, this parameter has meaningful information only if the channel instance is in doubt. However, the parameter value is still returned when requested, even if the channel instance is not in doubt.

For a receiving channel, it is the number of messages that have been received in the current batch. It is incremented as each message is received.

The value is reset to zero, for both sending and receiving channels, when the batch is committed.

CURSEQNO
For a sending channel, this is the message sequence number of the last message sent. It is updated as each message is sent, and when the channel becomes in doubt it is the message sequence number of the last message in the in-doubt batch.

For a saved channel instance, this parameter has meaningful information only if the channel instance is in doubt. However, the parameter value is still returned when requested, even if the channel instance is not in doubt.

For a receiving channel, it is the message sequence number of the last message that was received. It is updated as each message is received.

INDOUBT
Whether the channel is currently in doubt.

This is only YES while the sending Message Channel Agent is waiting for an acknowledgment that a batch of messages that it has sent has been successfully received. It is NO at all other times, including the period during which messages are being sent, but before an acknowledgment has been requested.

For a receiving channel, the value is always NO.

LSTLUWID
The logical unit of work identifier associated with the last committed batch of messages transferred.
LSTSEQNO
Message sequence number of the last message in the last committed batch. This number is not incremented by nonpersistent messages using channels with a NPM_SPEED of FAST.

STATUS
Current status of the channel. This is one of the following:

STARTING
A request has been made to start the channel but the channel has not yet begun processing. A channel is in this state if it is waiting to become active.

BINDING
Channel is performing channel negotiation and is not yet ready to transfer messages.

INITIALIZING
The channel initiator is attempting to start a channel. This is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS. On z/OS, this is displayed as INITIALIZI.

RUNNING
The channel is either transferring messages at this moment, or is waiting for messages to arrive on the transmission queue so that they can be transferred.

STOPPING
Channel is stopping or a close request has been received.

RETRYING
A previous attempt to establish a connection has failed. The MCA will retry the connection after the specified time interval.

PAUSED
The channel is waiting for the message-retry interval to complete before retrying an MQPUT operation. This is not valid on z/OS.

STOPPED
This state can be caused by one of the following:
- Channel manually stopped
  A user has entered a stop channel command against this channel.
- Retry limit reached
  The MCA has reached the limit of retry attempts at establishing a connection. No further attempt will be made to establish a connection automatically.

A channel in this state can be restarted only by issuing the START CHANNEL command, or starting the MCA program in an operating-system dependent manner.

REQUESTING
A local requester channel is requesting services from a remote MCA.

On z/OS, STATUS is not displayed if saved data is requested.
Note: For an inactive channel, CURMSGS, CURSEQNO, and CURLUWID have meaningful information only if the channel is INDOUBT. However they are still displayed and returned if requested.

**Current-only status**
The following information applies only to current channel instances. The information applies to all channel types, except where stated.

**BATCHES**
Number of completed batches during this session (since the channel was started).

**BATCHSZ**
The batch size being used for this session (valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS).
This parameter does not apply to server-connection channels, and no values are returned; if specified on the command, this is ignored.

**BUFSRCVD**
Number of transmission buffers received. This includes transmissions to receive control information only.

**BUFSSENT**
Number of transmission buffers sent. This includes transmissions to send control information only.

**BYTSRCVD**
Number of bytes received during this session (since the channel was started). This includes control information received by the message channel agent.

**BYTSSENT**
Number of bytes sent during this session (since the channel was started). This includes control information sent by the message channel agent.

**CHSTADA**
Date when this channel was started (in the form yyyy-mm-dd).

**CHSTATI**
Time when this channel was started (in the form hh.mm.ss).

**HBINT**
The heartbeat interval being used for this session.

**JOBNAME**
Name of job currently serving the channel.
- On Compaq OpenVMS, this is the process identifier, displayed in hexadecimal.
- On OS/2 Warp, OS/400, UNIX systems, and Windows, this is the concatenation of the process identifier and the thread identifier of the MCA program, displayed in hexadecimal.
- On Compaq NSK, this is the CPU ID and PID, displayed in hexadecimal.

This information is not available on z/OS. The parameter is ignored if specified.
DISPLAY CHSTATUS

LOCLADDR
Local communications address for the channel. The value returned
depends on the TRPRYPE of the channel (currently only TCP/IP is
supported).

LONGRTS
Number of long retry wait start attempts left. This applies only to sender
or server channels.

LSTMSGDA
Date when the last message was sent or MQI call was handled, see
LSTMSGTI.

LSTMSGTI
Time when the last message was sent or MQI call was handled.
For a sender or server, this is the time the last message (the last part of it if
it was split) was sent. For a requester or receiver, it is the time the last
message was put to its target queue. For a server-connection channel, it is
the time when the last MQI call completed.

MAXMSGL
The maximum message length being used for this session (valid only on
z/OS).

MCASTAT
Whether the Message Channel Agent is currently running. This is either
“running” or “not running”.
Note that it is possible for a channel to be in stopped state, but for the
program still to be running.
This information is not available on z/OS. The parameter is ignored if
specified.

MSGS
Number of messages sent or received (or, for server-connection channels,
the number of MQI calls handled) during this session (since the channel
was started).

NPMSPEED
The nonpersistent message handling technique being used for this session.

RQMNAME
The queue manager name, or queue-sharing group name, of the remote
system. This parameter does not apply to server-connection channels,
where no values are returned.

SHORTRTS
Number of short retry wait start attempts left. This applies only to sender
or server channels.

SSLPEER
Distinguished Name of the peer queue manager or client at the other end
of the channel.
The maximum length is 256 characters. This limit might mean that
exceptionally long Distinguished Names are truncated.

STOPREQ
Whether a user stop request is outstanding. This is either YES or NO.

Short status
The following information applies only to current channel instances.
DISPLAY CHSTATUS

QMNAME

The name of the queue manager that owns the channel instance.

Usage notes

The status information that is returned for various combinations of CHLDISP, CMDSCOPE, and status type are summarized in Table 4, Table 5, and Table 6.

Table 4. CHLDISP and CMDSCOPE for DISPLAY CHSTATUS CURRENT

<table>
<thead>
<tr>
<th>CHLDISP</th>
<th>CMDSCOPE( ) or CMDSCOPE (local-qmgr)</th>
<th>CMDSCOPE (qmgr-name)</th>
<th>CMDSCOPE(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIVATE</td>
<td>Common and current-only status for current private channels on the local queue manager</td>
<td>Common and current-only status for current private channels on the named queue manager</td>
<td>Common and current-only status for current private channels on all queue managers</td>
</tr>
<tr>
<td>SHARED</td>
<td>Common and current-only status for current shared channels on the local queue manager</td>
<td>Common and current-only status for current shared channels on the named queue manager</td>
<td>Common and current-only status for current shared channels on all queue managers</td>
</tr>
<tr>
<td>ALL</td>
<td>Common and current-only status for current private and shared channels on the local queue manager</td>
<td>Common and current-only status for current private and shared channels on the named queue manager</td>
<td>Common and current-only status for current private and shared channels on all active queue managers</td>
</tr>
</tbody>
</table>

Table 5. CHLDISP and CMDSCOPE for DISPLAY CHSTATUS SHORT

<table>
<thead>
<tr>
<th>CHLDISP</th>
<th>CMDSCOPE( ) or CMDSCOPE (local-qmgr)</th>
<th>CMDSCOPE (qmgr-name)</th>
<th>CMDSCOPE(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIVATE</td>
<td>STATUS and short status for current private channels on the local queue manager</td>
<td>STATUS and short status for current private channels on the named queue manager</td>
<td>STATUS and short status for current private channels on all active queue managers</td>
</tr>
<tr>
<td>SHARED</td>
<td>STATUS and short status for current shared channels on all active queue managers in the queue-sharing group</td>
<td>Not permitted</td>
<td>Not permitted</td>
</tr>
<tr>
<td>ALL</td>
<td>STATUS and short status for current private channels on the local queue manager and current shared channels in the queue-sharing group[i]</td>
<td>STATUS and short status for current private channels on the named queue manager</td>
<td>STATUS and short status for current private, and shared, channels on all active queue managers in the queue-sharing group[i]</td>
</tr>
</tbody>
</table>

Note:
1. In this case you get two separate sets of responses to the command on the queue manager where it was entered; one for PRIVATE and one for SHARED.

Table 6. CHLDISP and CMDSCOPE for DISPLAY CHSTATUS SAVED

<table>
<thead>
<tr>
<th>CHLDISP</th>
<th>CMDSCOPE( ) or CMDSCOPE (local-qmgr)</th>
<th>CMDSCOPE (qmgr-name)</th>
<th>CMDSCOPE(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIVATE</td>
<td>Common status for saved private channels on the local queue manager</td>
<td>Common status for saved private channels on the named queue manager</td>
<td>Common status for saved private channels on all active queue managers</td>
</tr>
<tr>
<td>SHARED</td>
<td>Common status for saved shared channels on all active queue managers in the queue-sharing group</td>
<td>Not permitted</td>
<td>Not permitted</td>
</tr>
</tbody>
</table>
Table 6. CHLDISP and CMDSCOPE for DISPLAY CHSTATUS SAVED (continued)

<table>
<thead>
<tr>
<th>CHLDISP</th>
<th>CMDSCOPE( ) or CMDSCOPE (local-qmgr)</th>
<th>CMDSCOPE (qmgr-name)</th>
<th>CMDSCOPE(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>Common status for saved private channels on the local queue manager and saved shared channels in the queue-sharing group</td>
<td>Common status for saved private channels on the named queue manager</td>
<td>Common status for saved private, and shared, channels on all active queue managers in the queue-sharing group</td>
</tr>
</tbody>
</table>
Use DISPLAY CLUSQMGR to display information about cluster channels for queue managers in a cluster.

Unlike the DISPLAY CHANNEL command, this command includes information about cluster channels that are auto-defined, and the status of cluster channels.

Notes:
1. On UNIX systems, the command is valid only on AIX, HP-UX, Linux, and Solaris.
2. On z/OS, the command fails if the channel initiator has not been started.

Synonym: DIS CLUSQMGR
Notes:

1  Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.

2  Valid only on z/OS.

3  Not valid on z/OS.

4  Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
DISPLAY CLUSQMGR

Parameter descriptions

\textit{(generic-qmgr-name)}

The name of the cluster queue manager for which information is to be displayed.

A trailing asterisk (*) matches all cluster queue managers with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all cluster queue managers.

\textbf{ALL}

Specify this to display all the parameters. If this parameter is specified, any parameters that are also requested specifically have no effect; all parameters are still displayed. This is the default if you do not specify a generic name and do not request any specific parameters.

\textbf{CHANNEL(generic-name)}

This is optional, and limits the information displayed to cluster channels with the specified channel name. The value can be a generic name.

\textbf{CLUSTER(generic-name)}

This is optional, and limits the information displayed to cluster queue managers with the specified cluster name. The value can be a generic name.

\textbf{CMDSCOPE}

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

\texttt{''}

The command is executed on the queue manager on which it was entered. This is the default value.

\texttt{qmgr-name}

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

\texttt{*}

The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

\textbf{Requested parameters}

Specify one or more parameters that define the data to be displayed. The parameters can be specified in any order, but do not specify the same parameter more than once.

Some parameters are relevant only for cluster channels of a particular type or types. Attributes that are not relevant for a particular type of channel cause no output, and do not cause an error.

\textbf{CLUSDATE}

The date on which the definition became available to the local queue manager, in the form \texttt{yyyy-mm-dd}.

\textbf{CLUSTIME}

The time at which the definition became available to the local queue manager, in the form \texttt{hh.mm.ss}.
DEFTYPE
How the cluster channel was defined:
CLUSSDR
As a cluster-sender channel from an explicit definition.
CLUSSDRA
As a cluster-sender channel by auto-definition alone.
CLUSSDRB
As a cluster-sender channel by auto-definition and an explicit definition.
CLUSRCVR
As a cluster-receiver channel from an explicit definition.

QMIID
The internally generated unique name of the cluster queue manager.

QMTYPE
The function of the cluster queue manager in the cluster:
REPOS
Provides a full repository service.
NORMAL
Does not provide a full repository service.

STATUS
The current status of the channel for this cluster queue manager. This is one of the following:
STARTING
A request has been made to start the channel but the channel has not yet begun processing. A channel is in this state if it is waiting to become active.
BINDING
The channel is performing channel negotiation and is not yet ready to transfer messages.
INACTIVE
The channel is not active.
INITIALIZING
The channel initiator is attempting to start a channel. On z/OS, this is displayed as INITIALIZ.
RUNNING
The channel is either transferring messages at this moment, or is waiting for messages to arrive on the transmission queue so that they can be transferred.
STOPPING
The channel is stopping, or a close request has been received.
RETRYING
A previous attempt to establish a connection has failed. The MCA will reattempt connection after the specified time interval.
PAUSED
The channel is waiting for the message-retry interval to complete before retrying an MQPUT operation.
STOPPED
This state can be caused by one of the following:
• Channel manually stopped.
A user has entered a stop channel command against this channel.

- Retry limit reached.
  The MCA has reached the limit of retry attempts at establishing a connection. No further attempt is made to establish a connection automatically.

A channel in this state can be restarted only by issuing the START CHANNEL command, or starting the MCA program in an operating-system dependent manner.

**REQUESTING**
A local requester channel is requesting services from a remote MCA.

**SUSPEND**
Whether this cluster queue manager is suspended from the cluster or not (as a result of the SUSPEND QMGR command). This is either YES or NO.

### Channel parameters

- **ALTDATE** The date on which the definition or information was last altered, in the form yyyy-mm-dd
- **ALTTIME** The time at which the definition or information was last altered, in the form hh.mm.ss
- **BATCHHB** The batch heartbeating value being used.
- **BATCHINT** Minimum batch duration
- **BATCHSZ** Batch size
- **CONNAME** Connection name
- **CONVERT** Whether the sender should convert application message data
- **DESCR** Description
- **DISCINT** Disconnection interval
- **HBINT** Heartbeat interval
- **KAINT** KeepAlive timing for the channel.
- **LOCLADDR** Local communications address for the channel.
- **LONGRTY** Long retry count
- **LONRTMR** Long retry timer
- **MAXMSGL** Maximum message length for channel
- **MCANAME** Message channel agent name
- **MCATYPE** Whether the message channel agent runs as a separate process or a separate thread
- **MCAUSER** Message channel agent user identifier
- **MODENAME** LU 6.2 mode name
- **MRDATA** Channel message-retry exit user data
- **MREXIT** Channel message-retry exit name
- **MRRTY** Channel message-retry exit retry count
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRTMR</td>
<td>Channel message-retry exit retry time</td>
</tr>
<tr>
<td>MSGDATA</td>
<td>Channel message exit user data</td>
</tr>
<tr>
<td>MSGEXIT</td>
<td>Channel message exit names</td>
</tr>
<tr>
<td>NETPRTY</td>
<td>The priority for the network connection</td>
</tr>
<tr>
<td>NPMSPRCE</td>
<td>Nonpersistent message speed</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>Password for initiating LU 6.2 session (if nonblank, this is displayed as asterisks)</td>
</tr>
<tr>
<td>PUTAUT</td>
<td>Put authority</td>
</tr>
<tr>
<td>RCVDATA</td>
<td>Channel receive exit user data</td>
</tr>
<tr>
<td>RCVEXIT</td>
<td>Channel receive exit names</td>
</tr>
<tr>
<td>SCYDATA</td>
<td>Channel security exit user data</td>
</tr>
<tr>
<td>SCYEXIT</td>
<td>Channel security exit name</td>
</tr>
<tr>
<td>SENDDATA</td>
<td>Channel send exit user data</td>
</tr>
<tr>
<td>SENDEXIT</td>
<td>Channel send exit names</td>
</tr>
<tr>
<td>SEQWRAP</td>
<td>Sequence number wrap value</td>
</tr>
<tr>
<td>SHORTRTRY</td>
<td>Short retry count</td>
</tr>
<tr>
<td>SHORTTMR</td>
<td>Short retry timer</td>
</tr>
<tr>
<td>SSLCAUTH</td>
<td>Whether SSL client authentication is required.</td>
</tr>
<tr>
<td>SSLCIPH</td>
<td>Cipher specification for the SSL connection.</td>
</tr>
<tr>
<td>SSLPEER</td>
<td>Filter for the Distinguished Name from the certificate of the peer queue manager or client at the other end of the channel.</td>
</tr>
<tr>
<td>TRPTYPE</td>
<td>Transport type</td>
</tr>
<tr>
<td>TPNAME</td>
<td>LU 6.2 transaction program name</td>
</tr>
<tr>
<td>USERID</td>
<td>User identifier for initiating LU 6.2 session</td>
</tr>
</tbody>
</table>
Use DISPLAY CMDSEVR to display the status of the command server.

**Synonym:** DIS CS

**DISPLAY CMDSEVR**

**Usage notes**

1. The command server takes messages from the system command input queue, and commands using CMDSCOPE, and processes them. DISPLAY CMDSEVR displays the status of the command server.

2. The response to this command is a message showing the current status of the command server, which is one of the following:
   - **ENABLED**: Available to process commands
   - **DISABLED**: Not available to process commands
   - **STARTING**: START CMDSEVR in progress
   - **STOPPING**: STOP CMDSEVR in progress
   - **STOPPED**: STOP CMDSEVR completed
   - **RUNNING**: Available to process commands, currently processing a message
   - **WAITING**: Available to process commands, currently waiting for a message
DISPLAY DQM

Use DISPLAY DQM to display information about the channel initiator.

Notes:
1. This is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, see the WebSphere MQ Intercommunication manual.
2. The command server must be running.

Synonym: DIS DQM

DISPLAY DQM

Parameter descriptions

CMDSCOPE
This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

''
The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

*
The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

Usage notes
1. The response to this command is a series of messages showing the current status of the channel initiator. This includes the following:
   - Whether the channel initiator is running or not
DISPLAY DQM

- Which listeners are started, and information about them.
- How many dispatchers are started, and how many were requested
- How many adapter subtasks are started, and how many were requested
- How many SSL subtasks are started, and how many were requested
- The TCP system name
- How many channel connections are current, and whether they are active, stopped, or retrying
- The maximum number of current connections
DISPLAY GROUP

Use DISPLAY GROUP to display information about the queue-sharing group to which the queue manager is connected.

**Synonym:** DIS GROUP

**DISPLAY GROUP**

**Usage notes**

1. The response to the DISPLAY GROUP command is a series of messages containing information about the queue-sharing group to which the queue manager is connected.

The following information is returned:
   • The name of the queue-sharing group
   • Whether all the queue managers that belong to the group are active or inactive
   • The names of all the queue managers that belong to the group
Use DISPLAY LOG to display archive log information.

**Synonym**: DIS LOG

[Diagram]

**Notes:**
1. Valid only when the queue manager is a member of a queue-sharing group.

**Parameter descriptions**

**CMDSCOPE**
This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

`''` The command is executed on the queue manager on which it was entered. This is the default value.

`qmgr-name` The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

`*` The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

**Usage notes**

1. DISPLAY LOG returns a report that shows the initial log parameters, and the current values as changed by the SET LOG command:
   - Length of time that an allowed archive read tape unit remains unused before it is deallocated (DEALLCT).
   - Size of input buffer storage for active and archive log data sets (INBUFF).
   - Size of output buffer storage for active and archive log data sets (OUTBUFF).
• Maximum number of dedicated tape units that can be set to read archive log tape volumes (MAXRTU).
• Maximum number of archive log volumes that can be recorded (MAXARCH).
• Whether archiving is on or off (OFFLOAD).
• Whether single or dual active logging is being used (TWOACTV).
• Whether single or dual archive logging is being used (TWOARCH).
• Whether single or dual BSDS is being used (TWOBSDS).
• Number of output buffers to be filled before they are written to the active log data sets (WRTHRSH).

It also returns a report about the status of the logs.

2. This command is issued internally by WebSphere MQ at the end of queue manager startup.
DISPLAY MAXSMSGS

Use DISPLAY MAXSMSGS to see the maximum number of messages that a task can get or put within a single unit of recovery.

**Note:** This command is valid only on z/OS and is retained for compatibility, although it can no longer be issued from the CSQINP1 initialization data set. You should use the MAXUMSGS parameter of the DISPLAY QMGR command instead.

**Synonym:** DIS MAXSM

**DISPLAY MAXSMSGS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMDSCOPE(’ ’)</td>
<td>The command is executed on the queue manager on which it was entered. This is the default value.</td>
</tr>
<tr>
<td>CMDSCOPE(qmgr-name)</td>
<td>The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group. You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.</td>
</tr>
<tr>
<td>CMDSCOPE(*)</td>
<td>The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.</td>
</tr>
</tbody>
</table>

Notes:
1. Valid only when the queue manager is a member of a queue-sharing group.

**Parameter descriptions**

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

‘ ’ The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.
Use DISPLAY NAMELIST to display the names in a namelist.

**Note:** On UNIX systems, the command is valid only on AIX, HP-UX, Linux and Solaris.

**Synonym:** DIS NL

**DISPLAY NAMELIST**

```
DISPLAY NAMELIST(generic-namelist-name)
```

**Requested attrs:**

```
ALTDATE
ALTTIME
DESCR
NAMCOUNT
NAMES
```

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.

**Parameter descriptions**

You must specify the name of the namelist definition you want to display. This can be a specific namelist name or a generic namelist name. By using a generic namelist name, you can display either:

- All namelist definitions
- One or more namelists that match the specified name

(generic-namelist-name)

The name of the namelist definition to be displayed (see “Rules for naming
DISPLAY NAMELIST

WebSphere MQ objects” on page 4). A trailing asterisk (*) matches all namelists with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all namelists. The namelists must all be defined to the local queue manager.

**ALL** Specify this to display all the parameters. If this parameter is specified, any parameters that are requested specifically have no effect; all the parameters are displayed.

This is the default if you do not specify a generic name, and do not request any specific parameters.

**CMDSCOPE**
This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

**QSGDISP**
Specifies the disposition of the objects for which information is to be displayed. Values are:

**LIVE**
This is the default value and displays information for objects defined with QSGDISP(QMGR) or QSGDISP(COPY).

**ALL**
Displays information for objects defined with QSGDISP(QMGR) or QSGDISP(COPY).

If there is a shared queue manager environment, and the command is being executed on the queue manager where it was issued, this option also displays information for objects defined with QSGDISP(GROUP).

If QSGDISP(ALL) is specified in a shared queue manager environment, the command might give duplicated names (with different dispositions).

In a shared queue manager environment, use

DISPLAY NAMELIST(name) CMDSCOPE(*) QSGDISP(ALL)

to list ALL objects matching name
in the queue-sharing group without duplicating those in the shared repository.

COPY Display information only for objects defined with QSGDISP(COPY).

GROUP Display information only for objects defined with QSGDISP(GROUP). This is allowed only if there is a shared queue manager environment.

PRIVATE Display information for objects defined with QSGDISP(QMGR) or QSGDISP(COPY). Note that QSGDISP(PRIVATE) displays the same information as QSGDISP(LIVE).

QMGR Display information only for objects defined with QSGDISP(QMGR).

QSGDISP displays one of the following values if it is specified, or if there is a shared queue manager environment:

QMGR The object was defined with QSGDISP(QMGR).

GROUP The object was defined with QSGDISP(GROUP).

COPY The object was defined with QSGDISP(COPY).

NLTYPE Indicates the type of namelist to be displayed.

This parameter is valid only on z/OS.

ALL Displays namelists of all types. This is the default.

NONE Displays namelists of type NONE.

QUEUE or Q Displays namelists that hold lists of queue names.

CLUSTER Displays namelists that are associated with clustering.

AUTHINFO Displays namelists that contain lists of authentication information object names.

Requested parameters Specify one or more parameters that define the data to be displayed. The parameters can be specified in any order, but do not specify the same parameter more than once.

The default, if no parameters are specified (and the ALL parameter is not specified) is that the object names and their NLTYPEs are displayed.

ALTDATE The date on which the definition was last altered, in the form yyyy-mm-dd

ALTTIME The time at which the definition was last altered, in the form hh.mm.ss

DESCR Description
DISPLAY NAMELIST

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NAMCOUNT</td>
<td>Number of names in the list</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NAMES</td>
<td>List of names</td>
<td></td>
</tr>
</tbody>
</table>

See “DEFINE NAMELIST” on page 114 for more information about the individual parameters.
Use `DISPLAY PROCESS` to display the attributes of one or more WebSphere MQ processes.

**Synonym:** DIS PRO

**DISPLAY PROCESS**

```
DISPLAY PROCESS(generic-process-name)
```

**Requested attrs:**

```
ALTDATE
ALTDATE
APPLICID
APPLTYPE
DESCR
ENVRDATA
USERDATA
```

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2. Valid only on z/OS.
3. Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

**Parameter descriptions**

You must specify the name of the process you want to display. This can be a specific process name or a generic process name. By using a generic process name, you can display either:

- All process definitions
DISPLAY PROCESS

- One or more processes that match the specified name

\( \text{(generic-process-name)} \)

The name of the process definition to be displayed (see “Rules for naming
WebSphere MQ objects” on page 4). A trailing asterisk (*) matches all
processes with the specified stem followed by zero or more characters. An
asterisk (*) on its own specifies all processes. The names must all be
deefined to the local queue manager.

ALL Specify this to display all the parameters. If this parameter is specified, any
parameters that are requested specifically have no effect; all parameters are
still displayed.

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris,
Windows, and z/OS, this is the default if you do not specify a generic
name and do not request any specific parameters.

On other platforms, if no parameters are specified (and the ALL parameter
is not specified), the default is that the process names are returned.

CMDSCOPE
This parameter applies to z/OS only and specifies how the command is
executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set
to GROUP.

'' The command is executed on the queue manager on which it was
entered. This is the default value.

\( qmgr-name \)

The command is executed on the queue manager you specify,
providing the queue manager is active within the queue-sharing
group.

You can specify a queue manager name, other than the queue
manager on which the command was entered, only if you are
using a queue-sharing group environment and if the command
server is enabled.

* The command is executed on the local queue manager and is also
passed to every active queue manager in the queue-sharing group.
The effect of this is the same as entering the command on every
queue manager in the queue-sharing group.

QSGDISP
Specifies the disposition of the objects for which information is to be
displayed. Values are:

LIVE This is the default value and displays information for
objects defined with QSGDISP(QMGR) or
QSGDISP(COPY).

ALL Displays information for objects defined with
QSGDISP(QMGR) or QSGDISP(COPY).

If there is a shared queue manager environment, and the
command is being executed on the queue manager where
it was issued, this option also displays information for
objects defined with QSGDISP(GROUP).
If QSGDISP(LIVE) is specified or defaulted, or if QSGDISP(ALL) is specified in a shared queue manager environment, the command might give duplicated names (with different dispositions).

**COPY**
Display information only for objects defined with QSGDISP(COPY).

**GROUP**
Display information only for objects defined with QSGDISP(GROUP). This is allowed only if there is a shared queue manager environment.

**PRIVATE**
Display information for objects defined with QSGDISP(QMGR) or QSGDISP(COPY). Note that QSGDISP(PRIVATE) displays the same information as QSGDISP(LIVE).

**QMGR**
Display information only for objects defined with QSGDISP(QMGR).

QSGDISP displays one of the following values if it is specified, or if there is a shared queue manager environment:

**QMGR** The object was defined with QSGDISP(QMGR).

**GROUP** The object was defined with QSGDISP(GROUP).

**COPY** The object was defined with QSGDISP(COPY).

**Requested parameters**
Specify one or more parameters that define the data to be displayed. The parameters can be specified in any order, but do not specify the same parameter more than once.

The default, if no parameters are specified (and the ALL parameter is not specified) is that the object names are displayed.

**ALTDATE** The date on which the definition was last altered, in the form yyyy-mm-dd

**ALTTIME** The time at which the definition was last altered, in the form hh.mm.ss

**APPLICID** Application identifier

**APPLTYPE** Application type

**DESCR** Description

**ENVRDATA** Environment data

**USERDATA** User data

See “DEFINE PROCESS” on page 118 for more information about individual parameters.
**DISPLAY QMGR**

Use **DISPLAY QMGR** to display the queue manager parameters for this queue manager.

**Synonym:** DIS QMGR

**DISPLAY QMGR.**

Requested attrs:
Chapter 2. The MQSC commands 223
DISPLAY QMGR

Notes:
1 Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2 Valid only on z/OS.
3 Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
4 Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
5 Valid only on AIX, HP-UX, Linux, OS/400, Solaris, Windows, and z/OS.
6 Valid only on AIX, HP-UX, Linux, and Solaris.

Parameter descriptions

ALL Specify this to display all the parameters. If this parameter is specified, any parameters that are requested specifically have no effect; all parameters are still displayed.

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows, this is the default if you do not request any specific parameters.

CMDSCOPE
This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP or SHARED.

'or' The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

Requested parameters
Specify one or more parameters that define the data to be displayed. The parameters can be specified in any order, but do not specify the same parameter more than once.

Note: If no parameters are specified (and the ALL parameter is not specified or defaulted), the queue manager name is returned.

You can request the following information for the queue manager:
**DISPLAY QMGR**

**ALTDATE**
The date on which the definition was last altered, in the form yyyy-mm-dd.
This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

**ALTTIME**
The time at which the definition was last altered, in the form hh.mm.ss.
This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

**AUTHOREV**
Whether authorization events are generated.

**CCSID**
Coded character set identifier. This applies to all character string fields defined by the application programming interface (API), including the names of objects, and the creation date and time of each queue. It does not apply to application data carried as the text of messages.

**CHAD**
Whether auto-definition of receiver and server-connection channels is enabled. This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.

**CHADEV**
Whether auto-definition events are enabled. This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.

**CHADEXIT**
The name of the channel auto-definition exit. This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.

**CLWLEXIT**
The name of the cluster workload exit.
This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

**CLWLDATA**
The data passed to the cluster workload exit.
This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

**CLWLLEN**
The maximum number of bytes of message data that is passed to the cluster workload exit.
This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

**CMDLEVEL**
Command level. This indicates the function level of the queue manager.

**COMMANDQ**
The name of the system-command input queue. Suitably authorized applications can put commands on this queue.

**CONFIGEV**
Whether configuration events are generated.
CPILEVEL
Reserved, this value has no significance.

DEADQ
The name of the queue to which messages are sent if they cannot be routed to their correct destination (the dead-letter queue or undelivered-message queue). The default is blanks.

For example, messages are put on this queue when:
- A message arrives at a queue manager, destined for a queue that is not yet defined on that queue manager
- A message arrives at a queue manager, but the queue for which it is destined cannot receive it because, possibly:
  - The queue is full
  - The queue is inhibited for puts
  - The sending node does not have authority to put the message on the queue
- An exception message needs to be generated, but the queue named is not known to that queue manager

Note: Messages that have passed their expiry time are not transferred to this queue when they are discarded.

If the dead-letter queue is not defined, or full, or unusable for some other reason, a message that would have been transferred to it by a message channel agent is retained instead on the transmission queue.

If a dead-letter queue or undelivered-message queue is not specified, all blanks are returned for this parameter.

DEFXMITQ
Default transmission queue name. This is the transmission queue on which messages, destined for a remote queue manager, are put if there is no other suitable transmission queue defined.

DESCR
Description.

DISTL
Whether distribution lists are supported by the queue manager. This is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.

EXPRYINT
On z/OS only, the approximate interval between scans for expired messages.

IGQ
On z/OS only, whether intra-group queuing is to be used.

IGQAUT
On z/OS only, displays the type of authority checking used by the intra-group queuing agent.

IGQUSER
On z/OS only, displays the user ID used by the intra-group queuing agent.

INHIBTEV
Whether inhibit events are generated.

LOCALEV
Whether local error events are generated.
MAXHANDS
The maximum number of open handles that any one task can have at any one time.

MAXMSGL
The maximum message length that can be handled by the queue manager. Individual queues or channels might have a smaller maximum than this.

MAXPRTY
The maximum priority. This is 9.

MAXUMSGS
Maximum number of uncommitted messages within one syncpoint.

PERFMEV
Whether performance-related events are generated.

PLATFORM
The architecture of the platform on which the queue manager is running. This is MVS (for z/OS platforms), OPENVMS, NSK, OS2, OS400, UNIX®, or WINDOWSNT.

QMIN
The internally generated unique name of the queue manager. This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

QMNAME
The name of the local queue manager. See “Rules for naming WebSphere MQ objects” on page 4.

QSGNAME
The name of the queue-sharing group to which the queue manager belongs, or blank if the queue manager is not a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.

REOMTEEV
Whether remote error events are generated.

REPOS
The name of a cluster for which this queue manager is to provide a repository manager service. This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

REPOSNL
The name of a list of clusters for which this queue manager is to provide a repository manager service. This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

SSLCRNLN(nlname)
Indicates the namelist of AUTHINFO objects being used for the queue manager for CRL checking.

SSLCRYP(string)
Indicates the name of the parameter string being used to configure the cryptographic hardware present on the system. Note that the PKCS #11 password appears as xxxxx.
SSLKEYR\(\text{string}\)
Indicates the name of the Secure Sockets Layer key repository.

SSLTASKS\(\text{integer}\)
On z/OS only, indicates the number of server subtasks to use for processing SSL calls. If you want to use SSL channels, you must have at least one of these tasks defined.

STRSTPEV
Whether start and stop events are generated.

SYNCPT
Whether syncpoint support is available with the queue manager. On Compaq OpenVMS, Linux, OS/2 Warp, OS/400, UNIX systems, Windows, and z/OS, it is always available.

TRIGINT
The trigger interval.
DISPLAY QSTATUS

<table>
<thead>
<tr>
<th>Compaq NSK</th>
<th>Compaq OpenVMS</th>
<th>OS/400</th>
<th>OS/2 Warp</th>
<th>UNIX systems</th>
<th>Windows</th>
<th>z/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

Use DISPLAY QSTATUS to display the status of one or more queues.

**Synonym:** DIS QS

**DISPLAY QSTATUS**

```
DISPLAY QSTATUS(generic-qname)
```

**Queue status:**

- CURDEPTH
- IPPROCS
- OPPROCS
- QSGDISP
- UNCOM

**Handle status:**
Notes:
1 Valid only when the queue manager is a member of a queue-sharing group.
2 Valid on z/OS only.
3 Channel initiator only
4 Not valid on Compaq NSK or z/OS.
5 IMS only
6 CICS only
7 RRSBATCH only

Parameter descriptions
You must specify the name of the queue for which you want to display status information. This can be a specific queue name or a generic queue name. By using a generic queue name you can display either:
• Status information for all queues, or
• Status information for one or more queues that match the specified name and other selection criteria

You must also specify whether you want status information about:
• Queues
• Handles that are accessing the queues

Note: You cannot use the DISPLAY QSTATUS command to display the status of an alias queue or remote queue. If you specify the name of one of these types
of queue, no data is returned. You can, however, specify the name of the
local queue or transmission queue to which the alias queue or remote queue
resolves.

(generic-qname)
The name of the queue for which status information is to be
displayed. A trailing asterisk (*) matches all queues with the
specified stem followed by zero or more characters. An asterisk (*)
on its own matches all queues. The queues must all be defined to
the local queue manager or queue-sharing group.

ALL
Display all the status information for each specified queue.
This is the default if you do not specify a generic name, and do not
request any specific parameters.

CMDSCOPE
This parameter specifies how the command is executed when the
queue manager is a member of a queue-sharing group. It is valid on
z/OS only.

'' The command is executed on the queue manager on which
it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you
specify, providing the queue manager is active within the
queue-sharing group.

You can specify a queue manager name, other than the
queue manager on which the command was entered, only if
you are using a queue-sharing group environment and if the
command server is enabled.

* The command is executed on the local queue manager and
is also passed to every active queue manager in the
queue-sharing group. The effect of this is the same as
entering the command on every queue manager in the
queue-sharing group.

OPENTYPE
Restricts the queues selected to those that have handles with the
specified type of access:

ALL Selects queues that are open with any type of access. This is
the default if the OPENTYPE parameter is not specified.

INPUT Selects queues that are open for input only. This option does
not select queues that are open for browse.

OUTPUT Selects queues that are open only for output.

The OPENTYPE parameter is valid only if TYPE(HANDLE) is also
specified.

TYPE Specifies the type of status information required:

QUEUE Status information relating to queues is displayed. This is
the default if the TYPE parameter is not specified.
DISPLAY QSTATUS

HANDLE
Status information relating to the handles that are accessing
the queues is displayed.

Queue status
For queue status, the following information is always returned for each queue that
satisfies the selection criteria, except where indicated:
• Queue name
• Type of information returned (TYPE parameter)
• Current queue depth (CURDEPTH parameter)
• Queue-sharing group disposition (QSGDISP parameter) - only if there is a
  queue-sharing group environment. (z/OS only)

The following parameters can be specified for TYPE(QUEUE) to request additional
information for each queue. If a parameter is specified that is not relevant for the
queue, operating environment, or type of status information requested, that
parameter is ignored.

CURDEPTH
The current depth of the queue, that is, the number of messages on the
queue. This includes both committed messages and uncommitted
messages.

IPPROCS
The number of handles that are currently open for input for the queue
(either input-shared or input-exclusive). This does not include handles that
are open for browse.

For shared queues, the number returned applies only to the queue
manager generating the reply. The number is not the total for all the queue
managers in the queue-sharing group.

OPPROCS
This is the number of handles that are currently open for output for the
queue.

For shared queues, the number returned applies only to the queue
manager generating the reply. The number is not the total for all the queue
managers in the queue-sharing group.

QSGDISP
Indicates the disposition of the queue. The value displayed is one of the
following:

  QMGR       The object was defined with QSGDISP(QMGR).
  COPY       The object was defined with QSGDISP(COPY).
  SHARED     The object was defined with QSGDISP(SHARED).

This parameter is valid on z/OS only.

This parameter is returned if requested, or if there is a queue-sharing
group environment.

For shared queues, if the CF structure used by the queue is unavailable or
has failed, the status information might be unreliable.
UNCOM
Indicates whether there are any uncommitted changes (puts and gets) pending for the queue. The value displayed is one of the following:
YES There are uncommitted changes pending.
NO There are no uncommitted changes pending.

For shared queues, the value returned applies only to the queue manager generating the reply. The value does not apply to all the queue managers in the queue-sharing group.

Handle status
For handle status, the following information is always returned for each queue that satisfies the selection criteria, except where indicated:
• Queue name
• Type of information returned (TYPE parameter)
• User identifier (USERID parameter) –not returned for APPLTYPE(SYSTEM)
• Process ID (PID parameter)
• Thread ID (TID parameter)
• Application tag (APPLTAG parameter)
• Application type (APPLTYPE parameter)
• Whether handle is providing input access (INPUT parameter)
• Whether handle is providing output access (OUTPUT parameter)
• Whether handle is providing browse access (BROWSE parameter)
• Whether handle is providing inquire access (INQUIRE parameter)
• Whether handle is providing set access (SET parameter)

The following parameters can be specified for TYPE(HANDLE) to request additional information for each queue. If a parameter that is not relevant is specified for the queue, operating environment, or type of status information requested, that parameter is ignored.

APPLTAG
A string containing the tag of the application connected to the queue manager. It is one of the following:
• z/OS batch job name
• TSO USERID
• CICS APPLID
• IMS region name
• Channel initiator job name
• OS/400 job name
• UNIX process
• Windows process

APPLTYPE
A string indicating the type of the application that is connected to the queue manager. It is one of the following:
• BATCH Application using a batch connection
• RRSBATCH RRS-coordinated application using a batch connection
• CICS CICS transaction
• IMS IMS transaction
• CHINIT Channel initiator
• SYSTEM Queue manager
• USER A user application
**DISPLAY QSTATUS**

**ASID**  A 4-character address-space identifier of the application identified by APPLTAG. It distinguishes duplicate values of APPLTAG.

This parameter is returned only when the queue manager owning the queue is running on z/OS, and the APPLTYPE parameter does not have the value SYSTEM.

**BROWSE**

Indicates whether the handle is providing browse access to the queue. The value is one of the following:

- **YES**  The handle is providing browse access.
- **NO**   The handle is not providing browse access.

**CHANNEL**

The name of the channel that owns the handle. If there is no channel associated with the handle, this parameter is blank.

This parameter is returned only when the handle belongs to the channel initiator.

**CONNAME**

The connection name associated with the channel that owns the handle. If there is no channel associated with the handle, this parameter is blank.

This parameter is returned only when the handle belongs to the channel initiator.

**INPUT**

Indicates whether the handle is providing input access to the queue. The value is one of the following:

- **SHARED**  The handle is providing shared-input access.
- **EXCL**   The handle is providing exclusive-input access.
- **NO**    The handle is not providing input access.

**INQUIRE**

Indicates whether the handle is providing inquire access to the queue. The value is one of the following:

- **YES**  The handle is providing inquire access.
- **NO**   The handle is not providing inquire access.

**OUTPUT**

Indicates whether the handle is providing output access to the queue. The value is one of the following:

- **YES**  The handle is providing output access.
- **NO**   The handle is not providing output access.

**PID**

Number specifying the process identifier of the application that has opened the specified queue.

This parameter is not valid on Compaq NSK and z/OS.

**PSBNAME**

The 8-character name of the program specification block (PSB) associated with the running IMS transaction. You can use the PSBNAME and PSTID to purge the transaction using IMS commands. It is valid on z/OS only.

This parameter is returned only when the APPLTYPE parameter has the value IMS.

**PSTID**

The 4-character IMS program specification table (PST) region identifier for the connected IMS region. It is valid on z/OS only.
This parameter is returned only when the APPLTYPE parameter has the value IMS.

**QSGDISP**

Indicates the disposition of the queue. It is valid on z/OS only. The value is one of the following:

- **QMGR** The object was defined with QSGDISP(QMGR).
- **COPY** The object was defined with QSGDISP(COPY).
- **SHARED** The object was defined with QSGDISP(SHARED).

This parameter is returned if requested, or if there is a queue-sharing group environment.

**SET**

Indicates whether the handle is providing set access to the queue. The value is one of the following:

- **YES** The handle is providing set access.
- **NO** The handle is not providing set access.

**TASKNO**

A 7-digit CICS task number. This number can be used in the CICS command "CEMT SET TASK(taskno) PURGE" to end the CICS task. This parameter is valid on z/OS only.

This parameter is returned only when the APPLTYPE parameter has the value CICS.

**TID**

Number specifying the thread identifier within the application process that has opened the specified queue.

This parameter is not valid on Compaq NSK and z/OS.

**TRANSID**

A 4-character CICS transaction identifier. This parameter is valid on z/OS only.

This parameter is returned only when the APPLTYPE parameter has the value CICS.

**URID**

The 32-character hexadecimal form of the 16-byte RRS unit-of-recovery identifier associated with the handle. This parameter is valid on z/OS only.

This parameter is returned only when the APPLTYPE parameter has the value RRSBATCH.

**USERID**

The user identifier associated with the handle.

This parameter is not returned when APPLTYPE has the value SYSTEM.
Display Queue

Use DISPLAY QUEUE to display the attributes of one or more queues of any type.

Notes:

1. On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS, you can use the following commands (or their synonyms) as an alternative way to display these attributes.
   - DISPLAY QALIAS
   - DISPLAY QCLUSTER
   - DISPLAY QLOCAL
   - DISPLAY QMODEL
   - DISPLAY QREMOTE

   These commands produce the same output as the DISPLAY QUEUE TYPE(queue-type) command. If you enter the commands this way, do not use the TYPE parameter because this causes an error.

2. On z/OS, the channel initiator must be running before you can display information about cluster queues (using TYPE(QCLUSTER) or the CLUSINFO parameter).

3. The command might not show every clustered queue in the cluster when issued on a partial repository, because the partial repository only knows about a queue once it has tried to use it.

Synonym: DIS Q

Display Queue

**Syntax**

```
DISPLAY QUEUE(generic-qname) [ALL] [CFSTRUCT] (generic-name) (1)

DISPLAY QUEUE CLUSINFO (2) CLUSNL (2)

DISPLAY QUEUE CLUSTER (2) CMDSCOPE( ' ' ) (1)
CMDSCOPE(qmgr-name) (3)
CMDSCOPE(+) (3)
```
DISPLAY QUEUE

Requested attrs:

Chapter 2. The MQSC commands

237
DISPLAY QUEUE

Notes:
1 Valid only on z/OS.
2 Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.
3 Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
4 Valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
5 Not valid on z/OS or OS/400.

Parameter descriptions

You must specify the name of the queue definition you want to display. This can be a specific queue name or a generic queue name. By using a generic queue name, you can display either:

- All queue definitions
- One or more queues that match the specified name

\[(\text{generic-q-name})\]

The local name of the queue definition to be displayed (see “Rules for naming WebSphere MQ objects” on page 4). A trailing asterisk (*) matches all queues with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all queues. The names must all be defined to the local queue manager.

ALL Specify this to display all the attributes. If this parameter is specified, any attributes that are also requested specifically have no effect; all attributes are still displayed.

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS, this is the default if you do not specify a generic name and do not request any specific attributes.

CFSTRUCT(\text{generic-name})

This parameter is optional and limits the information displayed to those queues where the value of the coupling facility structure is specified in brackets.

The value can be a generic name. If you do not enter a value for this parameter, CFSTRUCT is treated as a requested parameter.

CLUSINFO

This requests that, in addition to information about attributes of queues defined on this queue manager, information about these and other queues in the cluster that match the selection criteria is displayed. In this case, there might be multiple queues with the same name displayed. The cluster information is obtained from the repository on this queue manager.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

CLUSNL(\text{generic-name})

This is optional, and limits the information displayed if entered with a value in brackets:

- For queues defined on the local queue manager, only those with the specified cluster list. The value can be a generic name. Only queue types for which CLUSNL is a valid parameter are restricted in this way; other queue types that meet the other selection criteria are displayed.
DISPLAY QUEUE

- For cluster queues, only those belonging to clusters in the specified cluster list if the value is not a generic name. If the value is a generic name, no restriction is applied to cluster queues.

If you do not enter a value to qualify this parameter, it is treated as a requested parameter, and cluster list information is returned about all the queues displayed.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

Note: If the disposition requested is SHARED, CMDSCOPE must be blank or the local queue manager.

**CLUSTER(generic-name)**

This is optional, and limits the information displayed to queues with the specified cluster name if entered with a value in brackets. The value can be a generic name. Only queue types for which CLUSTER is a valid parameter are restricted in this way by this parameter; other queue types that meet the other selection criteria are displayed.

If you do not enter a value to qualify this parameter, it is treated as a requested parameter, and cluster name information is returned about all the queues displayed.

This parameter is valid only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP or SHARED.

'' The command is executed on the queue manager on which it was entered. This is the default value.

*qmgr-name*

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

**PSID(integer)**

The identifier of the page set where a queue resides. This is optional.

Specifying a value limits the information displayed to queues that have an active association to the specified page set. The value consists of two numeric characters, in the range 00 through 99. An asterisk (*) on its own specifies all page set identifiers. If you do not enter a value, page set information is returned about all the queues displayed.
The page set identifier is displayed only if there is an active association of the queue to a page set, that is, after the queue has been the target of an MQPUT request. The association of a queue to a page set is not active when:

- the queue has just been defined
- the queue’s STGCLASS attribute has been changed, and there has been no subsequent MQPUT request to the queue
- the queue manager has been restarted and there are no messages on the queue

This parameter is valid only on z/OS.

**QSGDISP**  
Specifies the disposition of the objects for which information is to be displayed. Values are:

- **LIVE**  
  This is the default value and displays information for objects defined with QSGDISP(QMGR) or QSGDISP(COPY). If there is a shared queue manager environment, and the command is being executed on the queue manager where it was issued, also display information for objects defined with QSGDISP(SHARED).

- **ALL**  
  Display information for objects defined with QSGDISP(QMGR) or QSGDISP(COPY). If there is a shared queue manager environment, and the command is being executed on the queue manager where it was issued, this option also displays information for objects defined with QSGDISP(GROUP) or QSGDISP(SHARED).

  In a shared queue manager environment, use
  ```
  DISPLAY QUEUE(name) CMDSCOPE(*) QSGDISP(ALL)
  ```
  to list ALL objects matching name in the queue-sharing group without duplicating those in the shared repository.

- **COPY**  
  Display information only for objects defined with QSGDISP(COPY).

- **GROUP**  
  Display information only for objects defined with QSGDISP(GROUP). This is allowed only if there is a shared queue manager environment.

- **PRIVATE**  
  Display information only for objects defined with QSGDISP(QMGR) or QSGDISP(COPY).

- **QMGR**  
  Display information only for objects defined with QSGDISP(QMGR).

- **SHARED**  
  Display information only for objects defined with QSGDISP(SHARED). This is allowed only in a shared queue manager environment.
**DISPLAY QUEUE**

**Note:** For cluster queues, this is always treated as a requested parameter. The value returned is the disposition of the real queue that the cluster queue represents.

If QSGDISP(LIVE) is specified or defaulted, or if QSGDISP(ALL) is specified in a shared queue manager environment, the command might give duplicated names (with different dispositions).

**Note:** In the QSGDISP(LIVE) case, this occurs only where a shared and a non-shared queue have the same name; such a situation should not occur in a well-managed system.

QSGDISP displays one of the following values if it is specified, or if there is a shared queue manager environment:

- **QMGR** The object was defined with QSGDISP(QMGR).
- **GROUP** The object was defined with QSGDISP(GROUP).
- **COPY** The object was defined with QSGDISP(COPY).
- **SHARED** The object was defined with QSGDISP(SHARED).

**STGCLASS(generic-name)**

This is optional, and limits the information displayed to queues with the storage class specified if entered with a value in brackets. The value can be a generic name.

If you do not enter a value to qualify this parameter, it is treated as a requested parameter, and storage class information is returned about all the queues displayed.

This parameter is valid only on z/OS.

**TYPE(queue-type)**

This is optional, and specifies the type of queues you want to be displayed. The default is to display all queue types; this includes cluster queues if CLUSINFO is also specified.

You can specify any of the queue types allowed for a DEFINE command (QLocal, QAlias, QRemote, or their synonyms). A queue type of QCluster can be specified to display only cluster queue information on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows, and z/OS.

If QCluster is specified, any selection criteria specified by the CFSTRUCT, STGCLASS, or PSID parameters are ignored.

On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows, QTYPE(type) can be used as a synonym for this parameter.

If no parameters are specified (and the ALL parameter is not specified or defaulted), the queue name and queue type are displayed.

**Requested parameters**

Specify one or more parameters that define the data to be displayed. The parameters can be specified in any order, but do not specify the same parameter more than once.

Most parameters are relevant only for queues of a particular type or types. Parameters that are not relevant for a particular type of queue cause no output, nor is an error raised.
Table 7 shows the parameters that are relevant for each type of queue. There is a brief description of each parameter after the table, but for more information, see the DEFINE command for each queue type.

Table 7. Parameters that can be returned by the DISPLAY QUEUE command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Local queue</th>
<th>Model queue</th>
<th>Alias queue</th>
<th>Remote queue</th>
<th>Cluster queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTDATE¹</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ALTTIME¹</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>BOQNAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BOTHRESH</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CFSTRUCT³</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLUSDATETIME</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLUSNL¹</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CLUSQMGR¹</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLUSQT¹</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLUSTER¹</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>CLUSTIME¹</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRDATE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRTIME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CURDEPTH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEFBIND¹</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DEFPRTY</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DEFPRTY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEFPRTY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEFPRTY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEFPRTY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEFPSIST</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DEFSOPT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEFTYPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESCRIPTOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISTL²</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GET</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>HARDENBO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDXTYPE³</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INITQ</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPROCS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAXDEPTH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>MAXMSGL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>MSGDLVSQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPPROCS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROCESS</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSID⁴</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>QDEPTHHI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QDEPTHLO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QDPHIEV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 7. Parameters that can be returned by the DISPLAY QUEUE command (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Local queue</th>
<th>Model queue</th>
<th>Alias queue</th>
<th>Remote queue</th>
<th>Cluster queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>QDPLOEV</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QDPMAXEV</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QMID¹</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QSGDISP³</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QSVCIEV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QSVCINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QTYPE</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>RETINTVL</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>RNAME</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>RQMNAME</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>SCOPE⁵</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>SHARE</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>STGCLASS³</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>TARGQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>TRIGDATA</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>TRIGDPTH</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>TRIGGER</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>TRIGMPRI</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>TRIGTYPE</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>USAGE</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>XMITQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

**Notes:**

1. Supported only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, Windows and z/OS.
2. Supported only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.
3. Supported only on z/OS.
4. Supported only on z/OS, returned only for queues with a disposition of QMGR or COPY.
5. Not supported on z/OS.

**ALTDATE**

The date on which the definition or information was last altered, in the form `yyyy-mm-dd`.

**ALTTIME**

The time at which the definition or information was last altered, in the form `hh.mm.ss`.

**BOQNAME**

Backout requeue name.
**DISPLAY QUEUE**

**BOTHRESH**
Backout threshold.

**CLUSDATE**
The date on which the definition became available to the local queue manager, in the form yyyy-mm-dd.

**CLUSNL**
The namelist that defies the cluster that the queue is in.

**CLUSQMGR**
The name of the queue manager that hosts the queue.

**CLUSQT**
Cluster queue type. This can be:
- **QALIAS**
  The cluster queue represents an alias queue.
- **QLOCAL**
  The cluster queue represents a local queue.
- **QMGR**
  The cluster queue represents a queue manager alias.
- **QREMOTE**
  The cluster queue represents a remote queue.

**CLUSTER**
The name of the cluster that the queue is in.

**CLUSTIME**
The time at which the definition became available to the local queue manager, in the form hh.mm.ss.

**CRDATE**
The date on which the queue was defined (in the form yyyy-mm-dd).

**CRTIME**
The time at which the queue was defined (in the form hh.mm.ss).

**CURDEPTH**
Current depth of queue.
On z/OS, CURDEPTH is returned as zero for queues defined with a disposition of GROUP. It is also returned as zero for queues defined with a disposition of SHARED if the CF structure that they use is unavailable or has failed.

**DEFBIND**
Default message binding.

**DEFPRTY**
Default priority of the messages put on the queue.

**DEFPSIST**
Whether the default persistence of messages put on this queue is set to NO or YES. NO means that messages are lost across a restart of the queue manager.

**DEFSOPT**
Default share option on a queue opened for input.

**DEFTYPE**
Queue definition type. This can be:
- **PREDEFINED** (Predefined)
DISPLAY QUEUE

The queue was created with a DEFINE command, either by an operator or by a suitably authorized application sending a command message to the service queue.

- **PERMDYN** (Permanent dynamic)
  Either the queue was created by an application issuing `MQOPEN` with the name of a model queue specified in the object descriptor (MQOD), or (if this is a model queue) this determines the type of dynamic queue that can be created from it.

  On z/OS the queue was created with `QSGDISP(QMGR)`.

- **TEMPDYN** (Temporary dynamic)
  Either the queue was created by an application issuing `MQOPEN` with the name of a model queue specified in the object descriptor (MQOD), or (if this is a model queue) this determines the type of dynamic queue that can be created from it.

  On z/OS the queue was created with `QSGDISP(QMGR)`.

- **SHAREDYN**
  A permanent dynamic queue was created when an application issued an `MQOPEN` API call with the name of this model queue specified in the object descriptor (MQOD).

  On z/OS, in a queue-sharing group environment, the queue was created with `QSGDISP(SHARED)`.

**DESCR**
Descriptive comment.

**DISTL**
Whether distribution lists are supported by the partner queue manager.
(Supported only on AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows.)

**GET**
Whether the queue is enabled for gets.

**HARDENBO**
Whether to harden the get back out count.

**INDXTYPE**
Index type (supported only on z/OS).

**INITQ**
Initiation queue name.

**IPPROCS**
Number of handles indicating that the queue is open for input.

On z/OS, IPPROCS is returned as zero for queues defined with a disposition of GROUP. With a disposition of SHARED, only the handles for the queue manager sending back the information are returned, not the information for the whole group.

**MAXDEPTH**
Maximum depth of queue.

**MAXMSGL**
Maximum message length.

**MSGDLVSQ**
Message delivery sequence.

**OPPROCS**
Number of handles indicating that the queue is open for output.
On z/OS, OPPROCS is returned as zero for queues defined with a disposition of GROUP. With a disposition of SHARED, only the handles for the queue manager sending back the information are returned, not the information for the whole group.

**PROCESS**
Process name.

**PUT** Whether the queue is enabled for puts.

**QDEPTHHI** Queue Depth High event generation threshold.

**QDEPTHLO** Queue Depth Low event generation threshold.

**QDPHIEV** Whether Queue Depth High events are generated.

**QDPLOEV** Whether Queue Depth Low events are generated.

**QDPMAXEV** Whether Queue Full events are generated.

**QMID** The internally generated unique name of the queue manager that hosts the queue.

**QSVCIEV** Whether service interval events are generated.

**QSVCINT** Service interval event generation threshold.

**QTYPE** Queue type.
- On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows, the queue type is always displayed if you specify a generic queue name and do not request any other parameters. On z/OS, the queue type is always displayed.
- On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows, TYPE(type) can be used as a synonym for this parameter.

**RETINTVL** Retention interval.

**RNAME** Name of the local queue, as known by the remote queue manager.

**RQMNAME** Remote queue manager name.

**SCOPE** Scope of queue definition (not supported on z/OS).

**SHARE** Whether the queue can be shared.

**STGCLASS** Storage class.

**TARGQ** Local name of aliased queue.
DISPLAY QUEUE

TRIGDATA
Trigger data.

TRIGDPTH
Trigger depth.

TRIGGER
Whether triggers are active.

TRIGMPRI
Threshold message priority for triggers.

TRIGTYPE
Trigger type.

USAGE
Whether or not the queue is a transmission queue.

XMITQ
Transmission queue name.
Use DISPLAY SECURITY to display the current settings for the security parameters.

**Synonym:** DIS SEC

**DISPLAY SECURITY**

```plaintext
CMDSCOPE('') (1)
CMDSCOPE(qmgr-name) (1)
CMDSCOPE(*)
```

**Requested attrs:**

```
ALL

- INTERVAL
- SWITCHES
- TIMEOUT
```

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group.

**Parameter descriptions**

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

**CMDSCOPE** cannot be used for commands issued from the first initialization input data set CSQINP1.

''

The command is executed on the queue manager on which it was entered. This is the default value.

$qmgr-name$

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group.
DISPLAY SECURITY

The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

ALL Display the TIMEOUT, INTERVAL, and SWITCHES parameters. This is the default if no requested parameters are specified.

INTERVAL
Time interval between checks.

SWITCHES
Display the current setting of the switch profiles.

If the subsystem security switch is off, no other switch profile settings are displayed.

TIMEOUT
Timeout value.

See “ALTER SECURITY” on page 50 for details of the TIMEOUT and INTERVAL parameters.
Use DISPLAY STGCLASS to display information about storage classes.

**Synonym:** DIS STC

**DISPLAY STGCLASS**

```
DISPLAY STGCLASS(generic-class)
```

**CMDSCOPE(' ')

**CMDSCOPE(qmgr-name)**

**CMDSCOPE(*)

**PSID(integer)**

**QSGDISP(LIVE)**

**QSGDISP(ALL)**

**QSGDISP(QMGR)**

**QSGDISP(COPY)**

**QSGDISP(GROUP)**

**QSGDISP(PRIVATE)**

**Requested attrs:**

```
ALTDATE
ALTTIME
DESCR
XCFGNAME
XCFMNAME
```

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.

**Parameter descriptions**

You use DISPLAY STGCLASS to show the page set identifiers that are associated with each storage class.

**(generic-class)**

Name of the storage class. This is required.

This is 1 through 8 characters. The first character is in the range A through Z; subsequent characters are A through Z or 0 through 9.

A trailing asterisk (*) matches all storage classes with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all storage classes.
DISPLAY STGCLASS

CMDSCOPE
This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

If QSGDISP is set to GROUP, CMDSCOPE must be blank or the local queue manager.

```
The command is executed on the queue manager on which it was entered. This is the default value.
```

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

```
The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.
```

PSID(integer)
The page set identifier that a storage class maps to. This is optional.

The string consists of two numeric characters, in the range 00 through 99. An asterisk (*) on its own specifies all page set identifiers. See "DEFINE PSID" on page 124.

QSGDISP
Specifies the disposition of the objects for which information is to be displayed. Values are:

LIVE
This is the default value and displays information for objects defined with QSGDISP(QMGR) or QSGDISP(COPY).

ALL
Displays information for objects defined with QSGDISP(QMGR) or QSGDISP(COPY).

If there is a shared queue manager environment, and the command is being executed on the queue manager where it was issued, this option also displays information for objects defined with QSGDISP(GROUP).

If QSGDISP(ALL) is specified in a shared queue manager environment, the command might give duplicated names (with different dispositions).

In a shared queue manager environment, use DISPLAY STGCLASS(generic-class) CMDSCOPE(*) QSGDISP(ALL)
to list ALL objects matching name

in the queue-sharing group without duplicating those in the shared repository.

COPY
Display information only for objects defined with QSGDISP(COPY).
DISPLAY STGCLASS

GROUP  Display information only for objects defined with QSGDISP(GROUP). This is allowed only if there is a shared queue manager environment.

PRIVATE  Display information only for objects defined with QSGDISP(QMGR) or QSGDISP(COPY).

QMGR  Display information only for objects defined with QSGDISP(QMGR).

QSGDISP displays one of the following values if it is specified, or if there is a shared queue manager environment:

QMGR  The object was defined with QSGDISP(QMGR).
GROUP  The object was defined with QSGDISP(GROUP).
COPY  The object was defined with QSGDISP(COPY).

ALL  Specify this to display all the parameters. If this parameter is specified, any parameters that are also requested specifically have no effect; all parameters are still displayed.

This is the default if you do not specify a generic name, and do not request any specific parameters.

Requested parameters
Specify one or more parameters that define the data to be displayed. The parameters can be specified in any order, but do not specify the same parameter more than once.

The default, if no parameters are specified (and the ALL parameter is not specified) is the storage class names and their page set identifiers are displayed.

ALTDATE  The date on which the definition was last altered, in the form yyyy-mm-dd.

ALTTIME  The time at which the definition was last altered, in the form hh.mm.ss.

DESCR  Descriptive comment.

XCFGNAME  The name of the XCF group that WebSphere MQ is a member of.

XCFMNAME  The XCF member name of the IMS system within the XCF group specified in XCFGNAME.
Use DISPLAY SYSTEM to display system information.

**Synonym:** DIS SYSTEM

### Parameter descriptions

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

```
''
```
The command is executed on the queue manager on which it was entered. This is the default value.

```
qmgr-name
```
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

```
*
```
The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

### Usage notes

1. DISPLAY SYSTEM returns a report that shows the initial values of the system parameters and the current values as changed by the SET SYSTEM command:
   - Default user ID for command security checks (CMDUSER).
   - Maximum number of connections from batch, CICS, IMS, and TSO tasks to a single instance of WebSphere MQ (CTHREAD).
   - Time in seconds for which queue manager exits can execute during each invocation (EXITLIM).
• How many started server tasks to use to run queue manager exits (EXITTCB).
• Maximum number of connections to a single instance of WebSphere MQ from batch or TSO background tasks (IDBACK).
• Maximum number of connections to a single instance of WebSphere MQ from TSO foreground tasks (IDFORE).
• Number of log records written by WebSphere MQ between the start of one checkpoint and the next (LOGLOAD).
• The OTMA connection parameters (OTMACON).
• Whether queue manager restart waits until all indexes are built, or completes before all indexes are built (QINDBLD).
• Coded character set identifier for the queue manager (QMCCSID).
• The queue-sharing group parameters (QSGDATA).
• The RESLEVEL auditing parameter (RESAUDIT).
• The message routing code assigned to messages not solicited from a specific console (ROUTCDE).
• Whether SMF accounting data is collected when WebSphere MQ is started (SMFACCT).
• Whether SMF statistics are collected when WebSphere MQ is started (SMFSTAT).
• Default time, in minutes, between each gathering of statistics (STATIME).
• Whether tracing is started automatically (TRACSTR).
• Size of trace table, in 4 KB blocks, to be used by the global trace facility (TRACTBL).
• Time in minutes between scanning the queue index for WLM-managed queues (WLMTIME).

2. This command is issued internally by WebSphere MQ at the end of queue manager startup.
Use DISPLAY THREAD to display information about active and in-doubt threads. Threads shown as in doubt on one invocation of this command will probably be resolved for subsequent invocations.

**Synonym:** DIS THD

**DISPLAY THREAD**

Notes:
1. Valid only when the queue manager is a member of a queue-sharing group.

**Parameter descriptions**

(\texttt{connection-name})

List of one or more connection-names (of 1 through 8 characters each).
- For batch connections, this name is the batch job name
- For CICS connections, this name is the CICS applid
- For IMS connections, this name is the IMS job name
- For TSO connections, this name is the TSO user ID
- For RRS connections, this is RRSBATCH

Threads are selected from the address spaces associated with these connections only.

\texttt{(*)} Displays threads associated with all connections to WebSphere MQ.

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

\texttt{''} The command is executed on the queue manager on which it was entered. This is the default value.
DISPLAY THREAD

qmgr-name

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

** TYPE **
The type of thread to display. This parameter is optional.

ACTIVE
Display only active threads.

An active thread is one for which a unit of recovery has started but not completed. Resources are held in WebSphere MQ on its behalf.

This is the default if TYPE is omitted.

INDOUBT
Display only in-doubt threads.

An in-doubt thread is one that is in the second phase of the two-phase commit operation. Resources are held in WebSphere MQ on its behalf. External intervention is needed to resolve the status of in-doubt threads. You might only have to start the recovery coordinator (CICS, IMS, or RRS), or you might need to do more. They might have been in doubt at the last restart, or they might have become in doubt since the last restart.

REGIONS
Display a summary of active threads for each active connection.

** Note:** Threads used internally by WebSphere MQ are excluded.

* Display both active and in-doubt threads, but not regions.

If, during command processing, an active thread becomes in doubt, it might appear twice: once as active and once as in doubt.

** QMNAME **
Specifies that WebSphere MQ should check whether the designated queue manager is INACTIVE, and if so, report any shared units of work that were in progress on the designated and inactive queue manager.

This option is valid only for TYPE(INDOUBT).

For more information about the DISPLAY THREAD command and in-doubt recovery, see the WebSphere MQ for z/OS System Administration Guide. Also, see messages CSQV401I through CSQV406I, and CSQV432I, in the WebSphere MQ for z/OS Messages and Codes manual.

** Note:** This command is issued internally by WebSphere MQ when taking a checkpoint, and when the queue manager is starting and stopping, so that a list of threads that are in doubt at the time is written to the z/OS console log.
Use DISPLAY TRACE to display a list of active traces.

Synonym: DIS TRACE

**Parameter descriptions**

All parameters are optional. Each option that is used limits the effect of the command to active traces that were started using the same option, either explicitly or by default, with exactly the same parameter values.
DISPLAY TRACE

*    Does not limit the list of traces. This is the default. The CLASS option
cannot be used with DISPLAY TRACE(*).

Each remaining parameter in this section limits the list to traces of the
corresponding type:

ACCTG  Accounting data (the synonym is A)

GLOBAL  Service data from the entire queue manager (the synonym is G)

STAT    Statistical data (the synonym is S)

COMMENT(string)  Specifies a comment. This does not appear in the display, but it might be
recorded in trace output.

DETAIL(output-type)  Limits the information that a trace displays based on the output-type
specified.

Possible values for output-type are:
1    Display summary trace information: TNO, TYPE, CLASS, and DEST
2    Display qualification trace information: TNO and RMID. Refer to
message CSQW127I (in the WebSphere MQ for z/OS Messages and
Codes manual) for more information about trace qualification.
1,2   Display both summary and qualification information
*    Display both summary and qualification information

If no parameter follows DETAIL (either DETAIL() or just DETAIL is used),
type 1 trace information is displayed.

CMDSCOPE  This parameter specifies how the command is executed when the queue
manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first
initialization input data set CSQINP1.

''    The command is executed on the queue manager on which it was
entered. This is the default value.

qmgr-name  The command is executed on the queue manager you specify,
providing the queue manager is active within the queue-sharing

group.

You can specify a queue manager name, other than the queue
manager on which the command was entered, only if you are
using a queue-sharing group environment and if the command
server is enabled.

Destination block

DEST  Limits the list to traces started for particular destinations. More than one
value can be specified, but do not use the same value twice. If no value is
specified, the list is not limited.

Possible values and their meanings are:

GTF    The Generalized Trace Facility
DISPLAY TRACE

RES  A wraparound table residing in the ECSA (extended common service area)
SMF  The System Management Facility
SRV  A serviceability routine designed for IBM® for problem diagnosis

See “START TRACE” on page 319 for a list of allowed destinations for each trace type.

Constraint block

CLASS(integer)
Limits the list to traces started for particular classes. See “START TRACE” on page 319 for a list of allowed classes.

The default is CLASS(*), which does not limit the list.

RMID(integer)
Limits the list to traces started for particular resource managers. See “START TRACE” on page 319 for a list of allowed resource manager identifiers. Do not use this option with STAT.

The default is RMID(*), which does not limit the list.

Note: Information about RMID 231 might be inaccurate if the trace has been altered using the ALTER TRACE command, or if the channel initiator has been stopped.

TNO(integer)
Limits the list to particular traces, identified by their trace number (1 to 32). Up to 8 trace numbers can be used. If more than one number is used, only one value for USERID can be used. The default is TNO(*), which does not limit the list.

USERID(string)
Limits the list to traces started for particular user IDs. Up to 8 user IDs can be used. If more than one user ID is used, only one value can be used for TNO. Do not use this option with STAT. The default is USERID(*), which does not limit the list.
Use DISPLAY USAGE to display information about the current state of a page set, or to display information about the log data sets.

**Synonym:** DIS USAGE

**DISPLAY USAGE**

<table>
<thead>
<tr>
<th>Compaq NSK</th>
<th>Compaq OpenVMS</th>
<th>OS/400</th>
<th>OS/2 Warp</th>
<th>UNIX systems</th>
<th>Windows</th>
<th>z/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

### Notes:

1. Valid only when the queue manager is a member of a queue-sharing group.

**Parameter descriptions**

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

- **''** The command is executed on the queue manager on which it was entered. This is the default value.

- **qmgr-name**

  The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

  You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

- *** **

  The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

**PSID(integer)**

The page-set identifier that a storage class maps to. This is optional.

This is a number, in the range 00 through 99. An asterisk (*) on its own specifies all page set identifiers.

The command fails if PSID has been specified together with **TYPE(DATASET).**
DISPLAY USAGE

<table>
<thead>
<tr>
<th>TYPE</th>
<th>Defines the type of information to be displayed. Values are:</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAGESET</td>
<td>Display page set and buffer pool information. This is the default.</td>
</tr>
<tr>
<td>DATASET</td>
<td>Display data set information (for log and backup data sets). This returns messages containing 44-character data set names for the following:</td>
</tr>
<tr>
<td></td>
<td>• The log data set containing the BEGIN_UR record for the oldest incomplete unit of work for this queue manager, or if there are no incomplete units of work, the log data set containing the current highest written RBA.</td>
</tr>
<tr>
<td></td>
<td>• The log data set containing the oldest restart_RBA of any pageset owned by this queue manager.</td>
</tr>
<tr>
<td></td>
<td>• The log data set whose timestamp range includes the timestamp of the last successful backup of any application structure known within the queue-sharing group.</td>
</tr>
<tr>
<td>ALL</td>
<td>Display both page set and data set information.</td>
</tr>
</tbody>
</table>

Note: This command is issued internally by WebSphere MQ during queue manager shutdown so that the restart RBA is recorded on the z/OS console log.
Use MOVE QLOCAL to move all the messages from one local queue to another.

**Synonym:** MOVE QL

**MOVE QLOCAL**

```plaintext
MOVE QLOCAL(source) CMDSCOPE(' ') QSGDISP(PRIVATE)

CMDSCOPE(qmgr-name) QSGDISP(SHARED)

TYPE(MOVE) TOQLOCAL(target)
```

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group.

**Parameter descriptions**

You must specify the names of two local queues: the one you want to move messages from (the source queue) and the one you want to move the messages to (the target queue).

**source**  The name of the local queue from which messages are moved. The name must be defined to the local queue manager.

The command fails if the queue contains uncommitted messages.

If an application has this queue open, or has open a queue that eventually resolves to this queue, the command fails. For example, the command fails if this queue is a transmission queue, and any queue that is, or resolves to, a remote queue that references this transmission queue, is open.

An application can open this queue while the command is in progress but the application waits until the command has completed.

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

' '  The command is executed on the queue manager on which it was entered. This is the default value.

**qmgr-name**  The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.
MOVE QLOCAL

QSGDISP
Specifies the disposition of the source queue.

PRIVATE
The queue is defined with QSGDISP(QMGR) or QSGDISP(COPY). This is the default value.

SHARED
The queue is defined with QSGDISP(SHARED). This is valid only in a queue-sharing group environment.

TYPE Specifies how the messages are moved.

MOVE
Move the messages from the source queue to the empty target queue.

The command fails if the target queue already contains one or more messages. The messages are deleted from the source queue. This is the default value.

ADD
Move the messages from the source queue and add them to any messages already on the target queue.

The messages are deleted from the source queue.

target The name of the local queue to which messages are moved. The name must be defined to the local queue manager.

The name of the target queue can be the same as that of the source queue only if the queue exists as both a shared and a private queue. In this case, the command moves messages to the queue that has the opposite disposition (shared or private) from that specified for the source queue on the QSGDISP parameter.

If an application has this queue open, or has open a queue that eventually resolves to this queue, the command fails. The command also fails if this queue is a transmission queue, and any queue that is, or resolves to, a remote queue that references this transmission queue, is open.

No application can open this queue while the command is in progress.

If you specify TYPE(MOVE), the command fails if the target queue already contains one or more messages.

The DEFTYPE, HARDENBO, INDXTYPE, and USAGE parameters of the target queue must be the same as those of the source queue.

Usage notes

1. A typical use of the MOVE QLOCAL command is to move messages from a private queue to a shared queue when you are setting up a queue-sharing group environment.
2. The MOVE QLOCAL command moves messages; it does not copy them.
3. The MOVE QLOCAL command moves messages in a similar way to an application performing successive MQGET and MQPUT calls. However, the MOVE QLOCAL command does not physically delete logically-expired messages and, therefore, no expiration reports are generated.
4. The priority, context, and persistence of each message are not changed.
5. The command performs no data conversion and calls no exits.
6. Confirm-on-delivery (COD) report messages are not generated but confirm-on-arrival (COA) report messages are. This means that more than one COA report message can be generated for a message.

7. The MOVE QLOCAL command transfers the messages in batches. At COMMIT time, if the trigger conditions are met, trigger messages are produced. This might be at the end of the move operation.

   **Note:** Before the transfer of messages begins, this command verifies that the number of messages on the source queue, when added to the number of messages on the target queue, does not exceed MAXDEPTH on the target queue.

   If the MAXDEPTH of the target queue were to be exceeded, no messages are moved.

8. The MOVE QLOCAL command can change the sequence in which messages can be retrieved. The sequence remains unchanged only if:
   - You specify `TYPE(MOVE)` and
   - The MSGDLVSQ parameter of the source and target queues is the same.

9. Messages are moved within one or more syncpoints. The number of messages in each syncpoint is determined by the queue manager.

10. If anything prevents the moving of one or more messages, the command stops processing. This can mean that some messages have already been moved, while others remain on the source queue. Some of the reasons that prevent a message being moved are:
    - The target queue is full.
    - The message is too long for the target queue.
    - The message is persistent, but the target queue cannot store persistent messages (for example, because the queue is shared).
    - The page set is full.
PING CHANNEL

Use PING CHANNEL to test a channel by sending data as a special message to the remote queue manager, and checking that the data is returned. The data is generated by the local queue manager.

Notes:
1. On z/OS:
   a. This is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, see the *WebSphere MQ Intercommunication* manual.
   b. The command server and the channel initiator must be running.
2. Where there is both a locally defined channel and an auto-defined cluster-sender channel of the same name, the command applies to the locally defined channel. If there is no locally defined channel but more than one auto-defined cluster-sender channel, the command applies to the channel that was last added to the local queue manager’s repository.

This command can be used only for sender (SDR), server (SVR), and cluster-sender (CLUSSDR) channels (including those that have been defined automatically). It is not valid if the channel is running; however, it is valid if the channel is stopped or in retry mode.

Synonym: PING CHL

PING CHANNEL

Parameter descriptions

*(channel-name)*
The name of the channel to be tested. This is required.
CMDSCOPE
This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

If CHLDISP is set to SHARED, CMDSCOPE must be blank or the local queue manager.

- The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name only if you are using a queue-sharing group environment and if the command server is enabled.

- The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

Note: The ‘*’ option is not permitted if CHLDISP is FIXSHARED.

CHLDISP
This parameter applies to z/OS only and can take the values of:
- PRIVATE
- SHARED
- FIXSHARED

In conjunction with the various values of the CMDSCOPE parameter, this parameter controls two types of channel:

SHARED
A receiving channel is shared if it was started in response to an inbound transmission directed to the queue-sharing group.

A sending channel is shared if its transmission queue has a disposition of SHARED.

PRIVATE
A receiving channel is private if it was started in response to an inbound transmission directed to the queue manager.

A sending channel is private if its transmission queue has a disposition other than SHARED.

Note: This disposition is not related to the disposition set by the disposition of the queue-sharing group of the channel definition.

The combination of the CHLDISP and CMDSCOPE parameters also controls from which queue manager the channel is operated. The possible options are:
- On the local queue manager where the command is issued.
- On another specific named queue manager in the group.
- On the most suitable queue manager in the group, determined automatically by the queue manager itself.
PING CHANNEL

The various combinations of CHLDISP and CMDSCOPE are summarized in Table 8.

Table 8. CHLDISP and CMDSCOPE for PING CHANNEL

<table>
<thead>
<tr>
<th>CHLDISP</th>
<th>CMDSCOPE( ) or CMDSCOPE (local-qmgr)</th>
<th>CMDSCOPE (qmgr-name)</th>
<th>CMDSCOPE(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIVATE</td>
<td>Ping private channel on the local queue manager</td>
<td>Ping private channel on the named queue manager</td>
<td>Ping private channel on all active queue managers</td>
</tr>
<tr>
<td>SHARED</td>
<td>Ping a shared channel on the most suitable queue manager in the group</td>
<td>Not permitted</td>
<td>Not permitted</td>
</tr>
</tbody>
</table>

This might automatically generate a command using CMDSCOPE and send it to the appropriate queue manager. If there is no definition for the channel on the queue manager to which the command is sent, or if the definition is unsuitable for the command, the command fails.

The definition of a channel on the queue manager where the command is entered might be used to determine the target queue manager where the command is actually run. Therefore, it is important that channel definitions are consistent. Inconsistent channel definitions might result in unexpected command behavior.

| FIXSHARED | Ping a shared channel on the local queue manager | Ping a shared channel on the named queue manager | Not permitted |

DATALEN(integer)

The length of the data, in the range 16 through 32 768. This is optional.
Use PING QMGR to test whether the queue manager is responsive to commands.

**Note:** If commands are issued to the queue manager by sending messages to the command server queue, this command causes a special message to be sent to it, consisting of a command header only, and checking that a positive reply is returned.

**Synonym:** PING QMGR

PING QMGR
Use RECOVER BSDS to reestablish a dual bootstrap data set (BSDS) after one has been disabled by a data set error.

**Note:** Command processing consists of allocating a data set with the same name as the one that encountered the error and copying onto the new data set the contents of the BSDS that does not have an error.

**Synonym:** REC BSDS

**RECOVER BSDS**

Use the command in the form:

```
RECOVER BSDS
```

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group.

**Parameter descriptions**

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

```
CMDSCOPE('')  (1)
```

The command is executed on the queue manager on which it was entered. This is the default value.

```
CMDSCOPE(qmgr-name)
```

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.
Use RECOVER CFSTRUCT to initiate recovery of CF application structures.

**Synonym**: REC CFSTRUCT

**RECOVER CFSTRUCT**

```
RECOVER CFSTRUCT(structure-name ...)
```

A list of names of up to 256 CF application structures to be recovered.

See “BACKUP CFSTRUCT” on page 58 for the rules for CF application structure names.

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

```
CB
```

The command is executed on the queue manager on which it was entered. This is the default value.

```
qmgr-name
```

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

**TYPE**

Specifies which variant of the RECOVER command is to be issued. Values are:

**NORMAL**

Perform a true recovery on the CF structures. This is the default.

**PURGE**

Recover to empty CF structures. Any messages in the CF structures are lost.

**Usage notes**

1. The command fails if TYPE(NORMAL) is specified and the application structure has not been flagged as in a FAILED state.
RECOVER CFSTRUCT

2. The command fails if any one of the specified structure names is not defined in the CFRM policy data set.

3. The recover process is both I/O and CPU intensive, and can only run on a single z/OS image. It should therefore be run on the most powerful or least busy system in the queue-sharing group.

4. The most likely failure is the loss of a complete CF and hence the simultaneous loss of all the application structures therein. If backup date and times are similar for each failed application structure, it is more efficient to recover them in a single RECOVER CFSTRUCT command.

5. This command fails if any of the specified CF structures is defined with either a CFLEVEL of less than 3, or with RECOVER set to NO.

6. The RECOVER CFSTRUCT command cannot be run during queue manager startup, that is, from the initialization input data sets.
Use REFRESH CLUSTER to discard all locally held cluster information (including any autodefined channels that are in doubt), and force it to be rebuilt. This enables you to perform a “cold-start” on the cluster.

Notes:

1. It is not normally necessary to issue a REFRESH CLUSTER command except in one of the following circumstances:
   - Messages have been removed from either the SYSTEM.CLUSTER.COMMAND.QUEUE, or from another queue manager’s SYSTEM.CLUSTER.TRANSMIT.QUEUE, where the destination queue is SYSTEM.CLUSTER.COMMAND.QUEUE on the queue manager in question.
   - Issuing a REFRESH CLUSTER command has been recommended by IBM Service.
   - The CLUSRCVR channels were removed from a cluster, or their CONNAMEs were altered on two or more full repository queue managers while they could not communicate.
   - The same name has been used for a CLUSRCVR channel on more than one queue manager in a cluster, and as a result, messages destined for one of the queue managers have been delivered to another. In this case, the duplicates should be removed, and then a REFRESH CLUSTER command should be issued on the single remaining queue manager that has the CLUSRCVR definition.
   - RESET CLUSTER ACTION(FORCEREMOVE) was issued in error.
   - The queue manager has been restarted from an earlier point in time than it last finished, (for example, by restoring backed up data.)

2. Issuing REFRESH CLUSTER does not correct mistakes in cluster definitions, nor is it necessary to issue the command after such mistakes have been corrected.

3. On UNIX systems, the command is valid only on AIX, HP-UX, Linux, and Solaris.

4. On z/OS, the command fails if the channel initiator has not been started.

5. On z/OS, any errors are reported to the console on the system where the channel initiator is running; they are not reported to the system that issued the command.

Synonym: REF CLUSTER

REFRESH CLUSTER

```sql
REFRESH CLUSTER(generic-clustername)
```
Notes:
1  Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.
2  Valid only on z/OS.

Parameter descriptions

(generic-clustername)
The name of the cluster to be refreshed. Alternatively the clustername can be specified as ‘*’. In this case the queue manager is refreshed in all the clusters that it is a member of. If used in conjunction with REPOS(YES), this has the additional effect of forcing the queue manager to restart its search for full repositories from the information in the local CLUSSDR definitions, even if the CLUSSDR connects the queue manager to several clusters.

This is required.

CMDSCOPE
This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

  ‘’  The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

REPOS
Specifies whether objects representing full repository cluster queue managers are also refreshed.

NO  The queue manager retains knowledge of all cluster queue manager and cluster queues marked as locally defined, and all cluster queue managers that are marked as full repositories. In addition, if the queue manager is a full repository for the cluster, it retains knowledge of the other cluster queue managers in the cluster. Everything else is removed from the local copy of the repository and rebuilt from the other full repositories in the cluster. Cluster channels are not stopped if REPOS(NO) is used, a full repository uses its CLUSSDR channels to inform the rest of the cluster that it has completed its refresh.

This is the default.

YES  Specifies that in addition to the REPOS(NO) behavior, objects representing full repository cluster queue managers are also refreshed. This option must not be used if the queue manager is itself a full repository, if it is a full repository, you must first alter it.
so that it is not a full repository for the cluster in question. The full repository location is recovered from the manually defined CLUSSDR definitions. After the refresh with REPOS(YES) has been issued, the queue manager can be altered so that it is once again a full repository, if required.

On z/OS, N and Y are accepted synonyms of NO and YES.

Usage Notes

1. Issuing REFRESH CLUSTER is disruptive to the cluster. It might make cluster objects invisible for a short period of time until the REFRESH processing completes. Specifically, if an application is using a queue that it has opened with MQOO_BIND_NOT_FIXED in effect, it might receive the return code MQRC_NO_DESTINATIONS_AVAILABLE. If the command is issued on a full repository queue manager, REFRESH CLUSTER might make a large volume of messages flow.

2. If cluster sender channels are running at the time REFRESH CLUSTER is issued, the refresh might not be complete until the channels have stopped and restarted. It is strongly recommended that all cluster sender channels for the cluster are stopped before the REFRESH CLUSTER command is issued. During the processing of the REFRESH CLUSTER command, the channel state might be recreated if the channel is not in doubt.

3. If you select REPOS(YES), you should check that all cluster sender channels in the relevant cluster are inactive or stopped before you issue the REFRESH CLUSTER command. If there are cluster sender channels running at the time the REFRESH CLUSTER command is processed, and they are used exclusively by the clusters being refreshed, and REPOS(YES) is used, the channels are stopped, by using the STOP CHANNEL command with MODE(FORCE) if necessary. This ensures that the refresh can remove the channel state, and that the channel runs with the refreshed version after the refresh has completed. If a channel’s state cannot be deleted, for example because it is in doubt, or because it is also running as part of another cluster, its state will not be new after the refresh, and it does not automatically restart if it was stopped.

Selecting REPOS(YES) on the sole working full repository queue manager in a cluster requires you to alter it to be a partial repository, with the result that there would be no full repository left. After the queue manager is refreshed and then restored to its status of a full repository, the other partial repositories would also have to be refreshed to restore a working cluster. If there is another working full repository in the cluster, it informs the other members of the cluster that the full repository executing the REFRESH CLUSTER command has resumed its role as a full repository, and no further action is needed.
Use REFRESH QMGR to perform special operations on queue managers.

**Synonym:** None

**REFRESH QMGR**

<table>
<thead>
<tr>
<th>CMDSCOPE</th>
<th>CMDSCOPE(qmgr-name)</th>
<th>CMDSCOPE(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMDSOCOE(' ')</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMDSCOPE(qmgr-name)</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>CMDSCOPE(*)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| INCLINT()              | NAME(*)            | OBJECT(ALL) |
| INCLINT(integer)       | NAME(generic-object-name) | OBJECT(QUEUE) |
|                       |                    | OBJECT(QLOCAL) |
|                       |                    | OBJECT(QMODEL) |
|                       |                    | OBJECT(QALIAS) |
|                       |                    | OBJECT(QREMOTE) |
|                       |                    | OBJECT(CHANNEL) |
|                       |                    | OBJECT(NAMELIST) |
|                       |                    | OBJECT(PROCESS) |
|                       |                    | OBJECT(QMGR) |
|                       |                    | OBJECT(STGCLASS) |
|                       |                    | OBJECT(CFSTRUCT) |
|                       |                    | OBJECT(AUTHINFO) |

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group.
   You can use queue-sharing groups only on z/OS.

**Parameter descriptions**

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE must be blank, or the local queue manager, if QSGDISP is set to GROUP or SHARED.

```
' '

The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.
*The command is executed on the local queue manager and is also
passed to every active queue manager in the queue-sharing group.
The effect of this is the same as entering the command on every
queue manager in the queue-sharing group.

This parameter is not valid with TYPE(EARLY).

**INCLINT**(integer)
Specifies a value in minutes defining a period immediately before the
current time, and requests that only objects that have been created or
changed within that period (as defined by the ALTDATE and ALTTIME
attributes) are included. The value must be in the range zero through 999
999. A value of zero means there is no time limit (this is the default).

This parameter is valid only with TYPE(CONFIGEV).

**NAME**(generic-object-name)
Requests that only objects whose names match the one specified are
included. A trailing asterisk (*) matches all object names with the specified
stem followed by zero or more characters. An asterisk (*) on its own
specifies all objects (this is the default). NAME is ignored if
OBJECT(QMGR) is specified.

This parameter is not valid with TYPE(EARLY).

**OBJECT**(objtype)
Requests that only objects of the specified type are included. (Synonyms
for object types, such as QL, can also be specified.) The default is ALL, to
include objects of every type.

This parameter is valid only with TYPE(CONFIGEV).

**TYPE**
This is required. Values are:

**CONFIGEV**
Requests that the queue manager generates a configuration event
message for every object that matches the selection criteria
specified by the OBJECT, NAME and INCLINT parameters.
Matching objects defined with QSGDISP(QMGR) or
QSGDISP(COPY) are always included. Matching objects defined
with QSGDISP(GROUP) or QSGDISP(SHARED) are included only
if the command is being executed on the queue manager where it
is entered.

**EARLY**
Requests that the subsystem function routines (generally known as
early code) for the queue manager replace themselves with the
corresponding routines in the linkpack area (LPA).

You only need to use this command after you install new
subsystem function routines (provided as corrective maintenance
or with a new version or release of WebSphere MQ). This
command instructs the queue manager to use the new routines.

See the [WebSphere MQ for z/OS System Setup Guide](#) for more
information about WebSphere MQ early code routines.

**EXPIRY**
Requests that the queue manager performs an expired message
scan for every queue that matches the selection criteria specified by
the NAME parameter. (The scan is performed regardless of the
setting of the EXPRYINT queue manager attribute.)
Note: If TYPE(EARLY) is specified, no other keywords are allowed.

**Usage Notes**

Issue this command with TYPE(CONFIGEV) after setting the CONFIGEV queue manager attribute to ENABLED, to bring the queue manager configuration up-to-date. To ensure that complete configuration information is generated, include all objects; if you have many objects, it might be preferable to use several commands, each with a different selection of objects, but such that all are included.

You can also use the command with TYPE(CONFIGEV) to recover from problems such as errors on the event queue. In such cases, use appropriate selection criteria, to avoid excessive processing time and event messages generation.

Issue the the command with TYPE(EXPIRY) at any time when you believe that a queue could contain numbers of expired messages.
Use REFRESH SECURITY to carry out a security refresh.

**Synonym:** REF SEC

REBUILD SECURITY is another synonym for REFRESH SECURITY.

**REFRESH SECURITY**

```
<--REFRESH SECURITY-->(
  -MQADMIN
  -MQNLIST
  -MQPROC
  -MQQUEUE

  CMDSCOPE('')

  CMDSCOPE(qmgr-name) (1)

  CMDSCOPE(+)
```

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group.

**Parameter descriptions**

This command causes WebSphere MQ to refresh in-storage ESM (external security manager, for example RACF) profiles. The in-storage profiles for the resources being requested are deleted. New entries are created when security checks for them are performed, and are validated when the user next requests access.

See the *WebSphere MQ for z/OS System Setup Guide* for more information about RACF® commands you have to issue when you issue this command.

You must specify the resource class for which the security refresh is to be performed. The classes are:

* All resource classes

**MQADMIN**

Administration type resources

**MQNLIST**

Namelist resources

**MQPROC**

Process resources

**MQQUEUE**

Queue resources

**Note:** If, when refreshing this class, it is determined that a security switch relating to one of the other classes has been changed, a refresh for that class also takes place.

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.
CMSDCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

''

The command is executed on the queue manager on which it was entered. This is the default value.

$qmgr-name$

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

*  
The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.
Use RESET CHANNEL to reset the message sequence number for a WebSphere MQ channel with, optionally, a specified sequence number to be used the next time that the channel is started.

Notes:

1. On z/OS:
   a. This is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, see the *WebSphere MQ Intercommunication* manual.
   b. The command server and channel initiator must be running.

2. This command can be issued to a channel of any type except SVRCONN and CLNTCONN channels, (including those that have been defined automatically). However, if it is issued to a sender, server or cluster-sender channel, then in addition to resetting the value at the end at which the command is issued, the value at the other (receiver, requester, or cluster-receiver) end is also reset to the same value the next time this channel is initiated (and resynchronized if necessary).

3. If the command is issued to a receiver, requester, or cluster-receiver channel, the value at the other end is *not* reset as well; this must be done separately if necessary.

4. Where there is both a locally defined channel and an auto-defined cluster-sender channel of the same name, the command applies to the locally defined channel. If there is no locally defined channel but more than one auto-defined cluster-sender channel, the command applies to the channel that was last added to the local queue manager’s repository.

*Synonym:* RESET CHL

**RESET CHANNEL**

<table>
<thead>
<tr>
<th>Compaq NSK</th>
<th>Compaq OpenVMS</th>
<th>OS/400</th>
<th>OS/2 Warp</th>
<th>UNIX systems</th>
<th>Windows</th>
<th>z/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Notes:

1. Valid only when the queue manager is a member of a queue-sharing group. You can use queue-sharing groups only on WebSphere MQ for z/OS.

2. Valid only on z/OS.
RESET CHANNEL

Parameter descriptions

(channel-name)
   The name of the channel to be reset. This is required.

CHLDISP
   This parameter applies to z/OS only and can take the values of:
   • PRIVATE
   • SHARED

In conjunction with the various values of the CMDSCOPE parameter, this parameter controls two types of channel:

SHARE
   A receiving channel is shared if it was started in response to an inbound transmission directed to the queue-sharing group.
   A sending channel is shared if its transmission queue has a disposition of SHARED.

PRIVATE
   A receiving channel is private if it was started in response to an inbound transmission directed to the queue manager.
   A sending channel is private if its transmission queue has a disposition other than SHARED.

Note: This disposition is not related to the disposition set by the disposition of the queue-sharing group of the channel definition.

The combination of the CHLDISP and CMDSCOPE parameters also controls from which queue manager the channel is operated. The possible options are:
   • On the local queue manager where the command is issued.
   • On another specific named queue manager in the group.

The various combinations of CHLDISP and CMDSCOPE are summarized in Table 9:

Table 9. CHLDISP and CMDSCOPE for RESET CHANNEL

<table>
<thead>
<tr>
<th>CHLDISP</th>
<th>CMDSCOPE ( ) or CMDSCOPE (local-qmgr)</th>
<th>CMDSCOPE (qmgr-name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIVATE</td>
<td>Reset private channel on the local queue manager</td>
<td>Reset private channel on the named queue manager</td>
</tr>
</tbody>
</table>
**Table 9. CHLDISP and CMDSCOPE for RESET CHANNEL (continued)**

<table>
<thead>
<tr>
<th>CHLDISP</th>
<th>CMDSCOPE( ) or CMDSCOPE (local-qmgr)</th>
<th>CMDSCOPE (qmgr-name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHARED</td>
<td>Reset a shared channel on all active queue managers. This might automatically generate a command using CMDSCOPE and send it to the appropriate queue managers. If there is no definition for the channel on the queue managers to which the command is sent, or if the definition is unsuitable for the command, the action fails there. The definition of a channel on the queue manager where the command is entered might be used to determine the target queue manager where the command is actually run. Therefore, it is important that channel definitions are consistent. Inconsistent channel definitions might result in unexpected command behavior.</td>
<td>Not permitted</td>
</tr>
</tbody>
</table>

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

If CHLDISP is set to SHARED, CMDSCOPE must be blank or the local queue manager.

' ' The command is executed on the queue manager on which it was entered. This is the default value.

**qmgr-name**

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name only if you are using a queue-sharing group environment and if the command server is enabled.

**SEQNUM(integer)**

The new message sequence number, which must be in the range 1 through 999 999 999. This is optional.
Use RESET CLUSTER to perform special operations on clusters.

Notes:
1. On UNIX systems, the command is valid only on AIX, HP-UX, Linux, and Solaris.
2. On z/OS, the command fails if the channel initiator has not been started.
3. On z/OS, any errors are reported to the console on the system where the channel initiator is running; they are not reported to the system that issued the command.

Synonym: None

Parameter descriptions

(clustername)
The name of the cluster to be reset. This is required.

ACTION(FORCEREMOVE)
Requests that the queue manager is forcibly removed from the cluster. This might be needed to ensure proper clean up after a queue manager has been deleted.

This action can be requested only by a repository queue manager.

CMDSCOPE
This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

' ' The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.
RESET CLUSTER

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

QMID(qmid)

The identifier of the queue manager to be forcibly removed.

QMNAME(qmname)

The name of the queue manager to be forcibly removed.

QUEUES

Specifies whether cluster queues owned by the queue manager being force removed are removed from the cluster.

NO Cluster queues owned by the queue manager being force removed are not removed from the cluster. This is the default.

YES Cluster queues owned by the queue manager being force removed are removed from the cluster in addition to the cluster queue manager itself. The cluster queues are removed even if the cluster queue manager is not visible in the cluster, perhaps because it was previously force removed without the QUEUES option.

On z/OS, N and Y are accepted synonyms of NO and YES.

Usage notes

1. To avoid any ambiguity, it is preferable to use QMID rather than QMNAME. The queue manager identifier can be found by commands such as DISPLAY QMGR and DISPLAY CLUSQMGR.

   If QMNAME is used, and there is more than one queue manager in the cluster with that name, the command is not actioned.

2. If you use characters other than those listed in “Rules for naming WebSphere MQ objects” on page 4 in your object or variable names, for example in QMID, you must enclose the name in quotes.

3. If you remove a queue manager from a cluster using this command, you can rejoin it to the cluster by issuing a REFRESH CLUSTER command. Wait at least 10 seconds before issuing a REFRESH CLUSTER command, because the repository ignores any attempt to rejoin the cluster within 10 seconds of a RESET CLUSTER command.
Use **RESET QSTATS** to report performance data for a queue and then to reset that data.

**Synonym:** None

### Parameter descriptions

**generic-qname**

The name of the local queue with a disposition of QMGR, COPY, or SHARED, but not GROUP, whose performance data is to be reset.

A trailing asterisk (*) matches all queues with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all queues.

The names must all be defined to the local queue manager or queue-sharing group.

The performance data is returned in the same format as parameters returned by DISPLAY commands. The data is:

**QSTATS**

The name of the queue

**QSGDISP**

The disposition of the queue, that is, QMGR, COPY, or SHARED.

**RESETINT**

The number of seconds since the statistics were last reset.

**HIQDEPTH**

The peak queue depth since the statistics were last reset.

**MSGSIN**

The number of messages that have been added to the queue by MQPUT and MQPUT1 calls since the statistics were last reset.

The count includes messages added to the queue in units of work that have not yet been committed, but the count is not decremented if the units of work are subsequently backed out.

**MSGSOUT**

The number of messages removed from the queue by destructive (non-browse) MQGET calls since the statistics were last reset.
RESET QSTATS

The count includes messages removed from the queue in units of work that have not yet been committed, but the count is not decremented if the units of work are subsequently backed out.

CMDSCOPE

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

''    The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

*    The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

Usage notes

1. If there is more than one queue with a name that satisfies the generic q-name, all those queues are reset.
2. Issue this command from an application, and not the z/OS console or its equivalent, to ensure that the statistical information is recorded.
3. Each queue manager in a queue-sharing group maintains its copy of the following performance statistics independently:

   MSGIN
   
   Incremented each time a message is put to the shared queue

   MSGOUT
   
   Incremented each time a message is removed from the shared queue

   HIQDEPTH
   
   Calculated by comparing its current value for HIQDEPTH with the new queue depth it obtains from the coupling facility during every put operation.

To obtain full statistics for a shared queue, you should specify CMDSCOPE(*) on RESET QSTATS to broadcast the command to all queue managers in the queue-sharing group.

The peak queue depth approximates to the maximum of all the returned HIQDEPTH values, and the total MQPUT and MQGET counts approximates to the sum of all the returned MSGIN and MSGOUT values respectively.

4. If the PERFMEV attribute of the queue manager is DISABLED, the command fails.
Use RESET TPIPE to reset the recoverable sequence numbers for an IMS Tpipe used by the WebSphere MQ-IMS bridge.

Notes:
1. This command is used in response to the resynchronization error reported in message CSQ2020E, and initiates resynchronization of the Tpipe with IMS.
2. The command fails if the queue manager is not connected to the specified XCF member.
3. The command fails if the queue manager is connected to the specified XCF member, but the Tpipe is open.
4. RESET TPIPE cannot be issued from the CSQINP1 and CSQINP2 initialization data sets.

Synonym: There is no synonym for this command.

Parameter descriptions

(tpipe-name)
The name of the Tpipe to be reset. This is required.

CMDSCOPE
This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

' ' The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.
You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

**ACTION**

Specifies whether to commit or back out any unit of recovery associated with this Tpipe. This is required if there is such a unit of recovery reported in message CSQ2020E; otherwise it is ignored.

**COMMIT**

The messages from WebSphere MQ are confirmed as having already transferred to IMS; that is, they are deleted from the WebSphere MQ-IMS bridge queue.

**BACKOUT**

The messages from WebSphere MQ are backed out; that is, they are returned to the WebSphere MQ-IMS bridge queue.

**SENDSEQ**(integer)

The new recoverable sequence number to be set in the Tpipe for messages sent by WebSphere MQ and to be set as the partner’s receive sequence number. It must be hexadecimal and can be up to 8 digits long, and can optionally be enclosed by X’ ’. It is optional; if omitted, the sequence number is not changed but the partner’s receive sequence is set to the WebSphere MQ send sequence number.

**RCVSEQ**(integer)

The new recoverable sequence number to be set in the Tpipe for messages received by WebSphere MQ and to be set as the partner’s send sequence number. It must be hexadecimal and can be up to 8 digits long, and can optionally be enclosed by X’ ’. It is optional; if omitted, the sequence number is not changed but the partner’s send sequence is set to the WebSphere MQ receive sequence number.

**XCFGNAME**(group-name)

The name of the XCF group to which the Tpipe belongs. This is 1 through 8 characters long. It is optional; if omitted, the group name used is that specified in the OTMACON system parameter.

**XCFMNAME**(member-name)

The name of the XCF member within the group specified by XCFGNAME to which the Tpipe belongs. This is 1 through 16 characters long, and is required.
Use RESOLVE CHANNEL to request a channel to commit or back out in-doubt messages.

Notes:
1. On z/OS:
   a. This is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, see the WebSphere MQ Intercommunication manual.
   b. The command server and the channel initiator must be running.
2. This command can be used only for sender (SDR), server (SVR), and cluster-sender (CLUSSDR) channels (including those that have been defined automatically).
3. Where there is both a locally defined channel and an auto-defined cluster-sender channel of the same name, the command applies to the locally defined channel. If there is no locally defined channel but more than one auto-defined cluster-sender channel, the command applies to the channel that was last added to the local queue manager’s repository.

Synonym: RESOLVE CHL (RES CHL on z/OS)

Parameter descriptions
(channel-name)
The name of the channel for which in-doubt messages are to be resolved. This is required.

ACTION
Specifies whether to commit or back out the in-doubt messages (this is required):
The messages are committed, that is, they are deleted from the transmission queue.

The messages are backed out, that is, they are restored to the transmission queue.

This parameter applies to z/OS only and can take the values of:
- PRIVATE
- SHARED

In conjunction with the various values of the CMDSCOPE parameter, this parameter controls two types of channel:

**SHARED**
- A receiving channel is shared if it was started in response to an inbound transmission directed to the queue-sharing group.
- A sending channel is shared if its transmission queue has a disposition of SHARED.

**PRIVATE**
- A receiving channel is private if it was started in response to an inbound transmission directed to the queue manager.
- A sending channel is private if its transmission queue has a disposition other than SHARED.

**Note:** This disposition is **not** related to the disposition set by the disposition of the queue-sharing group of the channel definition.

The combination of the CHLDISP and CMDSCOPE parameters also controls from which queue manager the channel is operated. The possible options are:
- On the local queue manager where the command is issued.
- On another specific named queue manager in the group.

The various combinations of CHLDISP and CMDSCOPE are summarized in Table 10.

**Table 10. CHLDISP and CMDSCOPE for RESOLVE CHANNEL**

<table>
<thead>
<tr>
<th>CHLDISP</th>
<th>CMDSCOPE (local-qmgr)</th>
<th>CMDSCOPE (qmgr-name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIVATE</td>
<td>Resolve private channel on the local queue manager</td>
<td>Resolve private channel on the named queue manager</td>
</tr>
</tbody>
</table>
Table 10. CHLDISP and CMDSCOPE for RESOLVE CHANNEL (continued)

<table>
<thead>
<tr>
<th>CHLDISP</th>
<th>CMDSCOPE( ) or CMDSCOPE (local-qmgr)</th>
<th>CMDSCOPE (qmgr-name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHARED</td>
<td>Resolve a shared channel on all active queue managers. This might automatically generate a command using CMDSCOPE and send it to the appropriate queue manager. If there is no definition for the channel on the queue manager to which the command is sent, or if the definition is unsuitable for the command, the command fails. The definition of a channel on the queue manager where the command is entered might be used to determine the target queue manager where the command is actually run. Therefore, it is important that channel definitions are consistent. Inconsistent channel definitions might result in unexpected command behavior.</td>
<td>Not permitted</td>
</tr>
</tbody>
</table>

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

If CHLDISP is set to SHARED, CMDSCOPE must be blank or the local queue manager.

```
''
```

The command is executed on the queue manager on which it was entered. This is the default value.

$qmgr$-name

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name only if you are using a queue-sharing group environment and if the command server is enabled.

**Usage notes**

This command is used when the other end of a link fails during the confirmation period, and for some reason it is not possible to reestablish the connection.

In this situation the sending end remains in doubt, as to whether or not the messages were received. Any outstanding units of work need to be resolved by being backed out or committed.

Care must be exercised in the use of this command. If the resolution specified is not the same as the resolution at the receiving end, messages can be lost or duplicated.
Use RESOLVE INDOUBT to resolve threads left in doubt because WebSphere MQ or a transaction manager could not resolve them automatically.

Note: This command does not apply to units of recovery associated with batch or TSO applications, unless you are using the RRS adapter.

**Synonym:** RES IND

**RESOLVE INDOUBT**

![Diagram of RESOLVE INDOUBT command]

**Parameter descriptions**

- **connection-name**
  - 1 through 8 character connection name.
  - For a CICS connection it is the CICS applid.
  - For an IMS adaptor connection, it is the IMS control region job name.
  - For an IMS bridge connection, it is the WebSphere MQ queue manager name.
  - For an RRS connection, it is RRSBATCH.

- **ACTION**
  - Specifies whether to commit or back out the in-doubt threads:
    - **COMMIT**
      - Commits the threads
    - **BACKOUT**
      - Backs out the threads

- **CMDSCOPE**
  - This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.
  - CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.
  - **' '**
    - The command is executed on the queue manager on which it was entered. This is the default value.

Notes:

1. Valid only when the queue manager is a member of a queue-sharing group.
**RESOLVE INDOUBT**

**qmgr-name**

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

**NID**

Network identifier. Specifies the thread or threads to be resolved.

*(network-id)*

This is as returned by the DISPLAY THREAD command, and is of the form net-node.net-urid, where:

- net-node identifies the originator of the thread, except RRSBATCH where it is omitted.
- net-urid is the hexadecimal number assigned to the unit of recovery by the originating system for the specific thread to be resolved.

When net-node is present there must be a period (.) between it and net-urid.

(*) Resolves all threads associated with the connection.

**QMNAME**

Specifies that if the designated queue manager is INACTIVE, WebSphere MQ should search information held in the coupling facility about units of work, performed by the indicated queue manager, that match the connection name and network identifier.

Matching units of work are either committed or backed out according to the ACTION specified.

Only the shared portion of the unit of work are resolved by this command.

As the queue manager is necessarily inactive, local messages are unaffected and remain locked until the queue manager restarts, or after restarting, connects with the transaction manager.

**Examples:**

```
RESOLVE INDOUBT(CICSA) ACTION(COMMIT) NID(CICSA.ABCDEF0123456789)
RESOLVE INDOUBT(CICSA) ACTION(BACKOUT) NID(*)
```
Use RESUME QMGR to inform other queue managers in a cluster that the local queue manager is available again for processing and can be sent messages. It reverses the action of the SUSPEND QMGR command.

**Notes:**

1. On UNIX systems, the command is valid only on AIX, HP-UX, Linux, and Solaris.
2. On z/OS, if you define CLUSTER or CLUSNL, the command fails if the channel initiator has not been started.
3. On z/OS, if you define CLUSTER or CLUSNL, any errors are reported to the console on the system where the channel initiator is running; they are not reported to the system that issued the command.

**Synonym:** None

**Parameter descriptions**

**CLUSTER(clusternam e)**

The name of the cluster for which availability is to be resumed.

**CLUSNL(nlname)**

The name of the namelist specifying a list of clusters for which availability is to be resumed.

**LOG**

Resumes logging and update activity for the queue manager that was suspended by a previous SUSPEND QMGR command. Valid on z/OS only.

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group. CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

```
' ' The command is executed on the queue manager on which it was entered. This is the default value.
```
RESUME QMGR

$qmgr-name$

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.
Use RVERIFY SECURITY to set a reverification flag for all specified users. The user is reverified the next time that security is checked for that user.

**Synonym:** REV SEC

**Note:** REVERIFY SECURITY is another synonym for RVERIFY SECURITY.

### Parameter descriptions

**(userids...)**

You must specify one or more user IDs. Each user ID specified is signed off and signed back on again the next time that a request is issued on behalf of that user that requires security checking.

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

' ' The command is executed on the queue manager on which it was entered. This is the default value.

**qmgr-name**

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.
Use SET ARCHIVE to dynamically change certain archive parameter values initially set by CSQ6ARVP.

Synonym: SET ARC

Parameter Block:

- CMDSCOPE("")
- CMDSCOPE(qmgr-name)
- CMDSCOPE(*)

Parameter descriptions

CMDSCOPE

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

""

The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.
You can specify a queue manager name, other than the queue manager on which it was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

**DEFAULT**

Resets the parameters to the values specified in the archive system parameters set by CSQ6ARVP.

**Parameter block**

For a full description of these parameters, see the WebSphere MQ for z/OS System Setup Guide.

Parameter block is any one or more of the following parameters that you want to change:

**ALCUNIT**

Specifies the unit in which primary and secondary space allocations are made.

Specify one of:

- **CYL** Cylinders
- **TRK** Tracks
- **BLK** Blocks

**ARCPFX1**

Specifies the prefix for the first archive log data set name.

See the TSTAMP parameter on page 301 for a description of how the data sets are named and for restrictions on the length of ARCPFX1.

**ARCPFX2**

Specifies the prefix for the second archive log data set name.

See the TSTAMP parameter on page 301 for a description of how the data sets are named and for restrictions on the length of ARCPFX2.

**ARCRETN**

Specifies the retention period, in days, to be used when the archive log data set is created.

The parameter must be in the range zero through 9999.

Discarding archive log data sets is discussed in the WebSphere MQ for z/OS System Administration Guide.

**ARCWRTC**

Specifies the list of z/OS routing codes for messages about the archive log data sets to the operator.

Specify up to 14 routing codes, each with a value in the range 1 through 16. You must specify at least one code. Separate codes in the list by commas, not by blanks.

For more information about z/OS routing codes, see the MVS Routing and Descriptor Codes manual.
SET ARCHIVE

ARCWTOR
Specifies whether a message is to be sent to the operator and a reply is received before attempting to mount an archive log data set.

Other WebSphere MQ users might be forced to wait until the data set is mounted, but they are not affected while WebSphere MQ is waiting for the reply to the message.

Specify either:
YES The device needs a long time to mount archive log data sets. For example, a tape drive. (The synonym is Y.)
NO The device does not have long delays. For example, DASD. (The synonym is N.)

BLKSIZE
Specifies the block size of the archive log data set. The block size you specify must be compatible with the device type you specify in the UNIT parameter.

The parameter must be in the range 4 097 through 28 672. The value you specify is rounded up to a multiple of 4 096.

This parameter is ignored for data sets that are managed by the storage management subsystem (SMS).

CATALOG
Specifies whether archive log data sets are cataloged in the primary integrated catalog facility (ICF) catalog.

Specify either:
NO Archive log data sets are not cataloged. (The synonym is N.)
YES Archive log data sets are cataloged. (The synonym is Y.)

COMPACT
Specifies whether data written to archive logs is to be compacted. This option applies only to a 3480 or 3490 device that has the improved data recording capability (IDRC) feature. When this feature is turned on, hardware in the tape control unit writes data at a much higher density than normal, allowing for more data on each volume. Specify NO if you do not use a 3480 device with the IDRC feature or a 3490 base model, with the exception of the 3490E.

Specify YES if you want the data to be compacted.

Specify either:
NO Do not compact the data sets. (The synonym is N.)
YES Compact the data sets. (The synonym is Y.)

PRIQTY
Specifies the primary space allocation for DASD data sets in ALCUNITs.

The value must be greater than zero.

This value must be sufficient for a copy of either the log data set or its corresponding BSDS, whichever is the larger.

PROTECT
Specifies whether archive log data sets are to be protected by discrete ESM (external security manager) profiles when the data sets are created.

Specify either:
NO Profiles are not created. (The synonym is N.)
YES Discrete data set profiles are created when logs are off-loaded. (The synonym is Y.) If you specify YES:
ESM protection must be active for WebSphere MQ.

- The user ID associated with the WebSphere MQ address space must have authority to create these profiles.
- The TAPEVOL class must be active if you are archiving to tape. Otherwise, off-loads will fail.

**QUIESCE**

Specifies the maximum time in seconds allowed for the quiesce when an ARCHIVE LOG command is issued with MODE QUIESCE specified.

The parameter must be in the range 1 through 999.

**SECOQTY**

Specifies the secondary space allocation for DASD data sets in ALCUNITs.

The parameter must be greater than zero.

**TSTAMP**

Specifies whether the archive log data set name has a time stamp in it.

Specify either:

- **NO** Names do not include a time stamp. (The synonym is N.) The archive log data sets are named:
  
  `arcpfxi.Annnnnnn`

  Where `arcpfxi` is the data set name prefix specified by ARCPFX1 or ARCPFX2. `arcpfxi` can have up to 35 characters.

- **YES** Names include a time stamp. (The synonym is Y.) The archive log data sets are named:
  
  `arcpfxi.cyyddd.Thhmsst.Annnnnnn`

  where `c` is 'D' for the years up to and including 1999 or 'E' for the year 2000 and later, and `arcpfxi` is the data set name prefix specified by ARCPFX1 or ARCPFX2. `arcpfxi` can have up to 19 characters.

- **EXT** Names include a time stamp. The archive log data sets are named:
  
  `arcpfxi.Oyyyyddd.Thhmsst.Annnnnnn`

  Where `arcpfxi` is the data set name prefix specified by ARCPFX1 or ARCPFX2. `arcpfxi` can have up to 17 characters.

**UNIT**

Specifies the device type or unit name of the device that is used to store the first copy of the archive log data set.

Specify a device type or unit name of 1 through 8 characters.

If you archive to DASD, you can specify a generic device type with a limited volume range.

**UNIT2**

Specifies the device type or unit name of the device that is used to store the second copy of the archive log data sets.

Specify a device type or unit name of 1 through 8 characters.

If this parameter is blank, the value set for the UNIT parameter is used.

**Usage notes**

1. The new values will be used at the next archive log off-load.
Use SET LOG to dynamically change certain log system parameter values that were initially set by CSQ6LOGP.

**Synonym: SET LOG**

Parameter Block:

- `DEALLCT(integer)`
- `MAXARCH(integer)`
- `MAXRTU(integer)`
- `WRTHRSH(integer)`

**Notes:**
1. Valid only when the queue manager is a member of a queue-sharing group.

### Parameter descriptions

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

- `' '`: The command is executed on the queue manager on which it was entered. This is the default value.
- `qmgr-name`: The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.
  
  You can specify a queue manager name, other than the queue manager on which it was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

- `*`: The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

- `DEFAULT`: Resets the parameters to the values specified in the archive system parameters set by CSQ6LOGP.
DEFAULT

Resets the parameters to the values specified in the archive system parameters set by CSQ6LOGP.

Parameter block

For a full description of these parameters, see the WebSphere MQ for z/OS System Setup Guide.

Parameter block is any one or more of the following parameters that you want to change:

DEALLCT

Specifies the length of time that an allocated archive read tape unit is allowed to remain unused before it is deallocated. You are recommended to specify the maximum possible values, within system constraints, for both options to achieve the optimum performance for reading archive tapes.

This, together with the MAXRTU parameter, allows WebSphere MQ to optimize archive log reading from tape devices.

The possible values are:

integer  Specifies the maximum time in minutes, between zero and 1439. Zero means that a tape unit is deallocated immediately.

NOLIMIT or 1440

Indicates that the tape unit is never deallocated.

MAXARCH

Specifies the maximum number of archive log volumes that can be recorded in the BSDS. When this number is exceeded, recording begins again at the start of the BSDS.

Use a decimal number in the range 10 through 1000.

MAXRTU(integer)

Specifies the maximum number of dedicated tape units that can be allocated to read archive log tape volumes. This overrides the value for MAXRTU set by CSQ6LOGP in the archive system parameters.

This, together with the DEALLCT parameter, allows WebSphere MQ to optimize archive log reading from tape devices.

Notes:

1. The integer value can range from 1 to 99.
2. If the number specified is greater than the current specification, the maximum number of tape units allowable for reading archive logs increases.
3. If the number specified is less than the current specification, tape units that are not being used are immediately deallocated to adjust to the new value. Active, or premounted, tape units remain allocated.
4. A tape unit is a candidate for deallocation because of a lowered value only if there is no activity for the unit.
5. When you are asked to mount an archive tape and you reply “CANCEL”, the MAXRTU value is reset to the current number of tape units.

For example, if the current value is 10, but you reply “CANCEL” to the request for the seventh tape unit, the value is reset to six.
**SET LOG**

**WRTHRSH**

Specifies the number of 4 KB output buffers to be filled before they are written to the active log data sets.

The larger the number of buffers, the less often the write takes place, and this improves the performance of WebSphere MQ. The buffers might be written before this number is reached if significant events, such as a commit point, occur.

Specify the number of buffers in the range 1 through 256.

**Usage notes**

1. Any changes to WRTHRSH take immediate effect.
2. Any change to MAXARCH takes effect for the next scheduled off-load (that is, not for any off-load in progress at the time the command is issued).
Use SET SYSTEM to dynamically change certain system parameter values that were initially set by CSQ6SYSP.

**Synonym:** None

```
SET SYSTEM  DEFAULT
```

**Parameter Block:**

- `CTHREAD(integer)`
- `IDBACK(integer)`
- `IDFORE(integer)`
- `LOGLOAD(integer)`
- `SERVICE(character)`
- `STATIME(integer)`
- `TRACTBL(integer)`

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group.

**Parameter descriptions**

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

- `' '`
  - The command is executed on the queue manager on which it was entered. This is the default value.

- `qmgr-name`
  - The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

  You can specify a queue manager name, other than the queue manager on which it was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

- `*`
  - The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.
**SET SYSTEM**

**DEFAULT**

Resets the parameters to the values specified in the archive system parameters set by CSQ6SYSP.

**Parameter block**

For a full description of these parameters, see the [WebSphere MQ for z/OS System Setup Guide](#).

Parameter block is any one or more of the following parameters that you want to change:

**CTHREAD**

Specifies the maximum number of connections from batch, CICS, IMS, TSO, and the channel initiator to a single instance of WebSphere MQ.

Specify a number in the range 1 through 32 767.

**IDBACK**

Specifies the maximum number of background batch and TSO connections to a single instance of WebSphere MQ.

Specify a number in the range 1 through 32 767.

**IDFORE**

Specifies the maximum number of TSO foreground connections to WebSphere MQ.

Specify a number in the range zero through 32 767.

**LOGLOAD**

Specifies the number of log records that WebSphere MQ writes between the start of one checkpoint and the next. WebSphere MQ starts a new checkpoint after the number of records that you specify has been written.

Specify a value in the range 200 through 16 000 000.

**SERVICE**

This parameter is reserved for use by IBM.

**STATIME**

Specifies the interval, in minutes, between consecutive gatherings of statistics.

Specify a number in the range zero through 1440.

If you specify a value of zero, both statistics data and accounting data is collected at the SMF data collection broadcast.

**TRACTBL**

Specifies the default size, in 4 KB blocks, of trace table where the global trace facility stores WebSphere MQ trace records.

Specify a value in the range 1 through 999.

**Note:** Storage for the trace table is allocated in the ECSA. Therefore, you must select this value with care.

**Usage notes**

1. The new values take immediate effect, with the possible exception of STATIME and TRACTBL.

Changes to STATIME take effect when the current interval expires, unless the new interval is less than the unexpired portion of the current interval, in which case statistics are gathered immediately and the new interval then takes effect.
For TRACTBL, if there is any trace currently in effect, the existing trace table continues to be used, and its size is unchanged. A new global trace table is only obtained for a new START TRACE command. If a new trace table is created with insufficient storage, the old trace table continues to be used, and the message CSQW153E is displayed.

2. If CTHREAD is reduced, any currently active threads can still connect to the WebSphere MQ system. If CTHREAD is increased, waiting threads are connected, up to the point where the number of active threads equals CTHREAD.
Use START CHANNEL to start a channel.

Notes:
1. On z/OS:
   a. This is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, see the WebSphere MQ Intercommunication manual.
   b. The command server and the channel initiator must be running.
2. This command can be issued to a channel of any type except CLNTCONN channels (including those that have been defined automatically). If, however, it is issued to a receiver (RCVR), server-connection (SVRCONN) or cluster-receiver (CLUSRCVR) channel, the only action is to enable the channel, not to start it.
3. Where there is both a locally defined channel and an auto-defined cluster-sender channel of the same name, the command applies to the locally defined channel. If there is no locally defined channel but more than one auto-defined cluster-sender channel, the command applies to the channel that was last added to the local queue manager’s repository.

Synonym: STA CHL

Parameter descriptions

(channel-name)
The name of the channel definition to be started. This is required. The name must be that of an existing channel defined on this queue manager.
This parameter applies to z/OS only and can take the values of:
- PRIVATE
- SHARED
- FIXSHARED

In conjunction with the various values of the CMDSCOPE parameter, this parameter controls two types of channel:

**SHARED**
- A receiving channel is shared if it was started in response to an inbound transmission directed to the queue-sharing group.
- A sending channel is shared if its transmission queue has a disposition of SHARED.

**PRIVATE**
- A receiving channel is private if it was started in response to an inbound transmission directed to the queue manager.
- A sending channel is private if its transmission queue has a disposition other than SHARED.

**Note:** This disposition is **not** related to the disposition set by the disposition of the queue-sharing group of the channel definition.

The combination of the CHLDISP and CMDSCOPE parameters also controls from which queue manager the channel is operated. The possible options are:
- On the local queue manager where the command is issued.
- On another specific named queue manager in the group.
- On every active queue manager in the group.
- On the most suitable queue manager in the group, determined automatically by the queue manager itself.

The various combinations of CHLDISP and CMDSCOPE are summarized in Table 11.

<table>
<thead>
<tr>
<th>CHLDISP</th>
<th>CMDSCOPE( ) or CMDSCOPE (local-qmgr)</th>
<th>CMDSCOPE (qmgr-name)</th>
<th>CMDSCOPE(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIVATE</td>
<td>Start as a private channel on the local queue manager</td>
<td>Start as a private channel on the named queue manager</td>
<td>Start as a private channel on all active queue managers</td>
</tr>
</tbody>
</table>
### Table 11. CHLDISP and CMDSCOPE for START CHANNEL (continued)

<table>
<thead>
<tr>
<th>CHLDISP</th>
<th>CMDSCOPE( ) or CMDSCOPE (local-qmgr)</th>
<th>CMDSCOPE (qmgr-name)</th>
<th>CMDSCOPE(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHARED</td>
<td>For a shared SDR, RQSTR, and SVR channel, start as a shared channel on the most suitable queue manager in the group. For a shared RCVR and SVRCONN channel, start the channel as a shared channel on all active queue managers. For a shared CLUSSDR or CLUSRCVR channel, this option is not permitted. This might automatically generate a command using CMDSCOPE and send it to the appropriate queue managers. If there is no definition for the channel on the queue managers to which the command is sent, or if the definition is unsuitable for the command, the action fails there. The definition of a channel on the queue manager where the command is entered might be used to determine the target queue manager where the command is actually run. Therefore, it is important that channel definitions are consistent. Inconsistent channel definitions might result in unexpected command behavior.</td>
<td>Not permitted</td>
<td>Not permitted</td>
</tr>
<tr>
<td>FIXSHARE</td>
<td>For a shared SDR, RQSTR, and SVR channel, with a nonblank CONNAME, start as a shared channel on the local queue manager. For all other types, this option is not permitted.</td>
<td>For a shared SDR, RQSTR, and SVR with a nonblank CONNAME, start as a shared channel on the named queue manager. For all other types, this option is not permitted.</td>
<td>Not permitted</td>
</tr>
</tbody>
</table>

Channels started with CHLDISP(FIXSHARE) are tied to the specific queue manager; if the channel initiator on that queue manager stops for any reason, the channels are not recovered by another queue manager in the group. See the [WebSphere MQ Intercommunication](#) manual for full details about SHARED and FIXSHARE channels.

### CMDSCOPE

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

If CHLDISP is set to SHARED, CMDSCOPE must be blank or the local queue manager.

The command is executed on the queue manager on which it was entered. This is the default value.
qmgr-name

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

This option is not permitted if CHLDISP is FIXSHARED.
Use START CHINIT to start a channel initiator.

**Note:** On z/OS:
1. This is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, see the WebSphere MQ Intercommunication manual.
2. The command server must be running.

**Synonym:** STA CHI

### WebSphere MQ for z/OS

**START CHINIT**

![Diagram of START CHINIT command for z/OS]

**Notes:**
1. Valid only when the queue manager is a member of a queue-sharing group.

### WebSphere MQ on other platforms

**START CHINIT**

![Diagram of START CHINIT command for other platforms]

### Parameter descriptions

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

- `'` The command is executed on the queue manager on which it was entered. This is the default value.
- `qmgr-name` The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.
You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

**ENVPARM(jcl-substitution)**

The parameters and values to be substituted in the JCL procedure (xxxxCHIN, where xxxx is the queue manager name) that is used to start the channel initiator address space.

- **jcl-substitution**
  - One or more character strings of the form keyword=value enclosed in single quotation marks. If you use more than one character string, separate the strings by commas and enclose the entire list in single quotation marks, for example ENVPARM(’HLQ=CSQ,VER=520’).

This parameter is valid only on z/OS.

**INITQ(string)**

The name of the initiation queue for the channel initiation process. This is the initiation queue that is specified in the definition of the transmission queue.

- This must not be specified on z/OS (the initiation queue on z/OS is always SYSTEM.CHANNEL.INITQ). On AIX, Compaq OpenVMS, HP-UX, Linux, OS/2 Warp, OS/400, Solaris, and Windows, you can specify which initiation queue to use; if you do not specify this, SYSTEM.CHANNEL.INITQ is used. On other platforms it must be specified.

**PARM(member-name)**

The load module that contains the channel initiator initialization parameters. *member-name* is the name of a load module provided by the installation. The default is CSQXPARM, which is provided by WebSphere MQ.

This parameter is valid only on z/OS.
START CMD SERV

START CMD SERV

<table>
<thead>
<tr>
<th>Compaq NSK</th>
<th>Compaq OpenVMS</th>
<th>OS/400</th>
<th>OS/2 Warp</th>
<th>UNIX systems</th>
<th>Windows</th>
<th>z/OS</th>
</tr>
</thead>
</table>

Use START CMD SERV to initialize the command server.

**Synonym:** STA CS

**START CMD SERV**

**Usage notes**

1. START CMD SERV starts the command server and allows it to process commands in the system-command input queue (SYSTEM.COMMAND.INPUT), mover commands, and commands using CMDSCOPE.
2. If this command is issued through the initialization files or through the operator console before work is released to the queue manager (that is, before the command server is started automatically), it overrides any earlier STOP CMD SERV command and allows the queue manager to start the command server automatically by putting it into an ENABLED state.
3. If this command is issued through the operator console while the command server is in a STOPPED or DISABLED state, it starts the command server and allows it to process commands on the system-command input queue, mover commands, and commands using CMDSCOPE immediately.
4. If the command server is in a RUNNING or WAITING state (including the case when the command is issued through the command server itself), or if the command server has been stopped automatically because the queue manager is closing down, no action is taken, the command server remains in its current state, and an error message is returned to the command originator.
5. START CMD SERV can be used to restart the command server after it has been stopped, either because of a serious error in handling command messages, or commands using the CMDSCOPE parameter.
START LISTENER

Use START LISTENER to start a channel listener.

Notes:

1. On UNIX systems, the command is valid only for AIX, HP-UX, Linux, and Solaris.

2. On z/OS:
   a. This is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, see the WebSphere MQ Intercommunication manual.
   b. The command server and the channel initiator must be running.
   c. A total of four listener tasks can be started. These are for TCP/IP and LU62 with dispositions of INDISP(QMGR) or INDISP(GROUP).
   d. If IPADDR is not specified, the listener listens on all available addresses.
   e. For TCP/IP, it is possible to listen on multiple addresses and port combinations.
   f. For each START LISTENER for TCP/IP request, the address and port combination is added to the list of combinations upon which the listener is currently listening.
   g. A START LISTENER for TCP/IP request fails if it specifies the same, or a subset or superset of an existing, combination of addresses and ports upon which a TCP/IP listener is currently listening.
   h. If you are starting a listener on a specific address to provide a secure interface with a security product, for example a firewall, it is important to ensure there is no linkage to the other non-secure interfaces in the system. You should disable IP forwarding and routing from other non-secure interfaces so that packets arriving at the other interface do not get passed to this specific address.
      Consult the appropriate TCP/IP documentation for information on how to do this.

3. On OS/400, OS/2 Warp, UNIX systems, and Windows, this command is valid only for channels for which the transmission protocol (TRPTYPE) is TCP.

Synonym: STA LSTR

START LISTENER

```
START LISTENER
  CMDSCOPE('
') (2)
  INDISP(QMGR) (2)
  INDISP(GROUP) (1)
CMDSCOPE(qmgr-name) (1)
```
**Parameter descriptions**

**CMDSCOPE**

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

- '' The command is executed on the queue manager on which it was entered. This is the default value.
- *qmgr-name* The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

**INDISP**

Specifies the disposition of the inbound transmissions that are to be handled. The possible values are:

- **QMGR**
  
  Listen for transmissions directed to the queue manager. This is the default.

- **GROUP**
  
  Listen for transmissions directed to the queue-sharing group. This is allowed only if there is a shared queue manager environment.

**IPADDR**

IP address for TCP/IP specified in dotted decimal or alphanumeric form. This is valid only if the transmission protocol (TRPTYPE) is TCP/IP.

**LUNAME**

The symbolic destination name for the logical unit as specified in the APPC side information data set. (This LU must be the same LU that is specified in the channel initiator parameters to be used for outbound transmissions.)
START LISTENER

This parameter is valid only for channels with a transmission protocol (TRPTYPE) of LU 6.2. A START LISTENER command that specifies TRPTYPE(LU62) must also specify the LUNAME parameter. This parameter is supported only on z/OS.

PORT(port-number)
Port number for TCP. This is valid only if the transmission protocol (TRPTYPE) is TCP.
This parameter is supported only on z/OS.

TRPTYPE
Transport type to be used. This is optional.
TCP TCP. This is the default if TRPTYPE is not specified.
LU62 SNA LU 6.2.
This parameter is supported only on z/OS.
START QMGR

Use START QMGR to initialize the queue manager. When the operation has been completed, the queue manager is active and available to CICS, IMS, batch, and TSO applications.

Synonym: STA QMGR

Parameter descriptions

These are optional.

**ENVPARM**(jcl-substitution)
The parameters and values to be substituted in the JCL procedure (xxxxMSTR, where xxxx is the queue manager name) that is used to start the queue manager address space.

**jcl-substitution**

One or more character strings of the form:

```
keyword=value
```

enclosed in single quotation marks. If you use more than one character string, separate the strings by commas and enclose the entire list in single quotation marks, for example

```
ENVPARM('HLQ=CSQ,VER=520')
```

MSTR is accepted as a synonym for ENVPARM

**PARM**(member-name)
The load module that contains the queue manager initialization parameters. *member-name* is the name of a load module provided by the installation.

The default is CSQZPARM, which is provided by WebSphere MQ.
Use START TRACE to start traces. When you issue this command, a trace number is returned in message number CSQW130I. You can use this trace number (TNO) in ALTER TRACE, DISPLAY TRACE, and STOP TRACE commands.

**Synonym**: STA TRACE

**START TRACE**

```
START TRACE (GLOBAL)
CMDSCOPE("") (1)
```

**Destination block:**

```
DEST(GTF)
```

**Constraint block:**

```
CLASS(*)
IFCID(*)
RMID(*)
```

Notes:

1. Valid only when the queue manager is a member of a queue-sharing group.

**Parameter descriptions**

If you do not specify a trace type to be started, the default (GLOBAL) trace is started. The types are:

**ACCTG**

Collects accounting data that can be used to charge your customers for their use of your queue manager. The synonym is A.
START TRACE

Note: Accounting data can be lost if the accounting trace is started or stopped while applications are running. For information about the conditions that must be satisfied for successful collection of accounting data, see the WebSphere MQ for z/OS System Setup Guide.

GLOBAL
This includes data from the entire queue manager. The synonym is G.

STAT
Collects statistical data broadcast by various components of WebSphere MQ, at time intervals that can be chosen during installation. The synonym is S.

CMDSCOPE
This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

' '
The command is executed on the queue manager on which it was entered. This is the default value.

$qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

COMMENT(string)
Specifies a comment that is reproduced in the trace output record (except in the resident trace tables). It can be used to record why the command was issued.

$string is any character string. It must be enclosed in single quotation marks if it includes a blank, comma, or special character.

Destination block

DEST
Specifies where the trace output is to be recorded. More than one value can be specified, but do not use the same value twice.

The meaning of each value is as follows:

GTF
The z/OS Generalized Trace Facility (GTF). If used, the GTF must be started and accepting user (USR) records before the START TRACE command is issued.

RES
A wrap-around table residing in the ECSA, or a data space for RMID 231.

SMF
The System Management Facility (SMF). If used, the SMF must be functioning before the START TRACE command is issued. The SMF record numbers reserved for use by WebSphere MQ are 115 and 116.

SRV
A serviceability routine reserved for IBM use only; not for general use.
START TRACE

Note: If your IBM support center need you to use this destination for your trace data they will supply you with module CSQWVSER. If you try to use destination SRV without CSQWVSER an error message is produced at the z/OS console when you issue the START TRACE command.

Allowed values, and the default value, depend on the type of trace started, as shown in the following table:

Table 12. Destinations allowed for each trace type

<table>
<thead>
<tr>
<th>Type</th>
<th>GTF</th>
<th>RES</th>
<th>SMF</th>
<th>SRV</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL</td>
<td>Allowed</td>
<td>Default</td>
<td>No</td>
<td>Allowed</td>
</tr>
<tr>
<td>STAT</td>
<td>No</td>
<td>No</td>
<td>Default</td>
<td>Allowed</td>
</tr>
<tr>
<td>ACCTG</td>
<td>Allowed</td>
<td>No</td>
<td>Default</td>
<td>Allowed</td>
</tr>
</tbody>
</table>

Constraint block

The constraint block places optional constraints on the kinds of data collected by the trace. The allowed constraints depend on the type of trace started, as shown in the following table:

Table 13. Constraints allowed for each trace type

<table>
<thead>
<tr>
<th>Type</th>
<th>CLASS</th>
<th>IFCID</th>
<th>RMID</th>
<th>USERID</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL</td>
<td>Allowed</td>
<td>Allowed</td>
<td>Allowed</td>
<td>Allowed</td>
</tr>
<tr>
<td>STAT</td>
<td>Allowed</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ACCTG</td>
<td>Allowed</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

CLASS

Introduces a list of classes of data gathered. The classes allowed, and their meaning, depend on the type of trace started:

(*) Starts a trace for all classes of data.

(integer)

Any number in the class column of the table that follows. You can use more than one of the classes that are allowed for the type of trace started. A range of classes can be specified as m:n (for example, CLASS(01:03)). If you do not specify a class, the default is to start class 1.

Table 14. IFCID descriptions for IFCID trace events and classes

<table>
<thead>
<tr>
<th>Class</th>
<th>IFCID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>0000</td>
<td>Reserved for IBM service</td>
</tr>
<tr>
<td>02</td>
<td>0018</td>
<td>User parameter error detected in a control block</td>
</tr>
<tr>
<td>03</td>
<td>0016</td>
<td>User parameter error detected on entry to MQI</td>
</tr>
<tr>
<td></td>
<td>0017</td>
<td>User parameter error detected on exit from MQI</td>
</tr>
<tr>
<td></td>
<td>0018</td>
<td>User parameter error detected in a control block</td>
</tr>
<tr>
<td>04</td>
<td>Various</td>
<td>Reserved for IBM service</td>
</tr>
</tbody>
</table>

Statistics trace

01 0001 Subsystem statistics
START TRACE

Table 14. IFCID descriptions for IFCID trace events and classes (continued)

<table>
<thead>
<tr>
<th>Class</th>
<th>IFCID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0002</td>
<td>Queue manager statistics</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Accounting trace</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>0003</td>
<td>The CPU time spent processing MQI calls and a count of MQPUT and MQGET calls</td>
</tr>
<tr>
<td>03</td>
<td>0025</td>
<td>Enhanced accounting and statistical data</td>
</tr>
</tbody>
</table>

IFCID

Reserved for IBM service.

RMID

Introduces a list of specific resource managers for which trace information is gathered. You cannot use this option for STAT or ACCTG traces.

(*) Starts a trace for all resource managers. This is the default.

(integer)

The identifying number of any resource manager in Table 15. You can use up to 8 of the allowed resource manager identifiers; do not use the same one twice.

If the list of RMIDs includes 231, the tracing for this resource manager is not started if one of the following is true:
- TRACE(STAT) or TRACE(ACCTG) is specified
- The list of destinations does not include RES
- This list of classes does not include 01 or 04

Also, comments are truncated to 120 characters.

If tracing for RMID 231 is started, it stops if the channel initiator is stopped.

Table 15. Resource Manager identifiers that are allowed

<table>
<thead>
<tr>
<th>RMID</th>
<th>Resource manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initialization procedures</td>
</tr>
<tr>
<td>2</td>
<td>Agent services management</td>
</tr>
<tr>
<td>3</td>
<td>Recovery management</td>
</tr>
<tr>
<td>4</td>
<td>Recovery log management</td>
</tr>
<tr>
<td>6</td>
<td>Storage management</td>
</tr>
<tr>
<td>7</td>
<td>Subsystem support for allied memories</td>
</tr>
<tr>
<td>8</td>
<td>Subsystem support for subsystem interface (SSI) functions</td>
</tr>
<tr>
<td>12</td>
<td>System parameter management</td>
</tr>
<tr>
<td>16</td>
<td>Instrumentation commands, trace, and dump services</td>
</tr>
<tr>
<td>23</td>
<td>General command processing</td>
</tr>
<tr>
<td>24</td>
<td>Message generator</td>
</tr>
<tr>
<td>26</td>
<td>Instrumentation accounting and statistics</td>
</tr>
<tr>
<td>148</td>
<td>Connection manager</td>
</tr>
<tr>
<td>197</td>
<td>CF manager</td>
</tr>
<tr>
<td>199</td>
<td>Functional recovery</td>
</tr>
<tr>
<td>200</td>
<td>Security management</td>
</tr>
</tbody>
</table>
Table 15. Resource Manager identifiers that are allowed (continued)

<table>
<thead>
<tr>
<th>RMID</th>
<th>Resource manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>Data management</td>
</tr>
<tr>
<td>211</td>
<td>Lock management</td>
</tr>
<tr>
<td>212</td>
<td>Message management</td>
</tr>
<tr>
<td>213</td>
<td>Command server</td>
</tr>
<tr>
<td>215</td>
<td>Buffer management</td>
</tr>
<tr>
<td>231</td>
<td>Channel Initiator</td>
</tr>
<tr>
<td>242</td>
<td>WebSphere MQ-IMS bridge</td>
</tr>
<tr>
<td>245</td>
<td>DB2® manager</td>
</tr>
</tbody>
</table>

TDATA
Reserved for IBM service.

USERID
Introduces a list of specific user IDs for which trace information is gathered. You cannot use this option for STAT or ACCTG traces.

(*) Starts a trace for all user IDs. This is the default.

(userid) Names a user ID. You can use up to 8 user IDs; a separate trace is started for each.
Use STOP CHANNEL to stop a channel.

Notes:
1. On z/OS:
   a. This is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, see the WebSphere MQ Intercommunication manual.
   b. The command server and the channel initiator must be running.
2. Any channels in STOPPED state need to be started manually; they are not started automatically. See the WebSphere MQ Intercommunication manual for information about restarting stopped channels.
3. This command can be issued to a channel of any type except CLNTCONN channels (including those that have been defined automatically).
4. Where there is both a locally defined channel and an auto-defined cluster-sender channel of the same name, the command applies to the locally defined channel. If there is no locally defined channel but more than one auto-defined cluster-sender channel, the command applies to the channel that was last added to the local queue manager’s repository.

Synonym: STOP CHL

STOP CHANNEL

Parameter descriptions

(channel-name)
The name of the channel to be stopped. This is required.
CMDSCOPE

This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

If CHLDISP is set to SHARED, CMDSCOPE must be blank or the local queue manager.

' ' The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name only if you are using a queue-sharing group environment and if the command server is enabled.

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

CHLDISP

This parameter applies to z/OS only and can take the values of:
• PRIVATE
• SHARED

In conjunction with the various values of the CMDSCOPE parameter, this parameter controls two types of channel:

SHARED

A receiving channel is shared if it was started in response to an inbound transmission directed to the queue-sharing group.

A sending channel is shared if its transmission queue has a disposition of SHARED.

PRIVATE

A receiving channel is private if it was started in response to an inbound transmission directed to the queue manager.

A sending channel is private if its transmission queue has a disposition other than SHARED.

Note: This disposition is not related to the disposition set by the disposition of the queue-sharing group of the channel definition.

The combination of the CHLDISP and CMDSCOPE parameters also controls from which queue manager the channel is operated. The possible options are:
• On the local queue manager where the command is issued.
• On another specific named queue manager in the group.
• On every active queue manager in the group.
• On the most suitable queue manager in the group, determined automatically by the queue manager itself.

The various combinations of CHLDISP and CMDSCOPE are summarized in Table 16 on page 326.
**Table 16. CHLDISP and CMDSCOPE for STOP CHANNEL**

<table>
<thead>
<tr>
<th>CHLDISP</th>
<th>CMDSCOPE( ) or CMDSCOPE(qmgr-name)</th>
<th>CMDSCOPE(qmgr-name)</th>
<th>CMDSCOPE(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIVATE</td>
<td>Stop as a private channel on the local queue manager.</td>
<td>Stop as a private channel on the named queue manager</td>
<td>Stop as a private channel on all active queue managers</td>
</tr>
<tr>
<td>SHARED</td>
<td>For RCVR and SVRCONN channels, stop as shared channel on all active queue managers.</td>
<td>Not permitted</td>
<td>Not permitted</td>
</tr>
<tr>
<td></td>
<td>For SDR, RQSTR, and SVR channels, stop as a shared channel on the queue manager where it is running. If the channel is in an inactive state (not running), or if it is in RETRY state because the channel initiator on which it was running has stopped, a STOP request for the channel is issued on the local queue manager. This might automatically generate a command using CMDSCOPE and send it to the appropriate queue manager. If there is no definition for the channel on the queue manager to which the command is sent, or if the definition is unsuitable for the command, the command fails. The definition of a channel on the queue manager where the command is entered might be used to determine the target queue manager where the command is actually run. Therefore, it is important that channel definitions are consistent. Inconsistent channel definitions might result in unexpected command behavior.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CONNAME**: `connection-name`

- Connection name. Only channels matching the specified connection name are stopped.

**MODE**

- Specifies whether the current batch is allowed to finish in a controlled manner. This parameter is optional.

**QUIESCE**

- Allows the current batch to finish processing, except on z/OS where the channel stops after the current message has finished processing. (The batch is then ended and no more messages are sent, even if there are messages waiting on the transmission queue.)

- For a receiving channel, if there is no batch in progress, the channel waits for either:
  - The next batch to start
STOP CHANNEL

- The next heartbeat (if heartbeats are being used) before it stops.

For server-connection channels, allows the current connection to end.

This is the default.

FORCE
Terminates transmission of any current batch. This is likely to result in in-doubt situations.

For server-connection channels, breaks the current connection, returning MQRC_CONNECTION_BROKEN.

TERMINATE
On z/OS this is synonymous with FORCE. On other platforms, this parameter terminates transmission of any current batch. This allows the command to actually terminate the channel thread or process.

For server-connection channels, breaks the current connection, returning MQRC_CONNECTION_BROKEN.

QMNAME(qmname)
Queue manager name. Only channels matching the specified remote queue manager are stopped

STATUS
Specifies the new state of any channels stopped by this command.

STOPPED
The channel is stopped. For a sender or server channel the transmission queue is set to GET(DISABLED) and NOTRIGGER.

This is the default if QMNAME or CONNAME are not specified.

INACTIVE
The channel is inactive. For a sender or server channel the transmission queue is unaffected.

This is the default if QMNAME or CONNAME are specified.

Usage notes
If you specify either QMNAME or CONNAME, STATUS must either be INACTIVE or not specified. Do not specify a QMNAME or CONNAME and STATUS(STOPPED). It is not possible to have a channel stopped for one partner but not for others. This sort of function can be provided by a channel security exit. For more information about channel exits, see the WebSphere MQ Intercommunication manual.
Use STOP CHINIT to stop a channel initiator.

Notes:
1. This is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, see the WebSphere MQ Intercommunication manual.
2. The command server must be running.

**Synonym:** STOP CHI

**STOP CHINIT**

```
STOP CHINIT
```

**Parameter descriptions**

**CMDSCOPE**
This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

```
CMDSCOPE('', (1))
CMDSCOPE(qmgr-name), (1)
CMDSCOPE(*), (1)
```

**Notes:**
1. Valid only when the queue manager is a member of a queue-sharing group.

**SHARED**
Specifies whether the channel initiator should attempt to restart any active shared sending channels that it owns on another queue manager. The possible values are:

* The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

**Notes:**
1. Valid only when the queue manager is a member of a queue-sharing group.
STOP CHINIT

**RESTART**
Shared sending channels are to be restarted. This is the default.

**STOP**
Shared sending channels are not to be restarted.

**Usage notes**

1. When you issue the STOP CHINIT command, WebSphere MQ stops any channels that are running in the following way:
   - Sender and server channels are stopped using STOP CHANNEL MODE(QUIESCE)
   - All other channels are stopped using STOP CHANNEL MODE(FORCE)

   See [“STOP CHANNEL” on page 324](#) for information about what this involves.

2. You might receive communications-error messages as a result of issuing the STOP CHINIT command.
STOP CMDSEVR

STOP CMDSEVR

Use STOP CMDSEVR to stop the command server.

Synonym: STOP CS

STOP CMDSEVR

Usage notes

1. STOP CMDSEVR stops the command server from processing commands in the system-command input queue (SYSTEM.COMMAND.INPUT), mover commands, and commands using CMDScope.

2. If this command is issued through the initialization files or through the operator console before work is released to the queue manager (that is, before the command server is started automatically), it prevents the command server from starting automatically and puts it into a DISABLED state. It overrides an earlier START CMDSEVR command.

3. If this command is issued through the operator console or the command server while the command server is in a RUNNING state, it stops the command server when it has finished processing its current command. When this happens, the command server enters the STOPPED state.

4. If this command is issued through the operator console while the command server is in a WAITING state, it stops the command server immediately. When this happens, the command server enters the STOPPED state.

5. If this command is issued while the command server is in a DISABLED or STOPPED state, no action is taken, the command server remains in its current state, and an error message is returned to the command originator.
STOP LISTENER

Use STOP LISTENER to stop a channel listener.

Notes:
1. This is valid only for channels used for distributed queuing without CICS. If you are using CICS for distributed queuing, see the WebSphere MQ Intercommunication manual.
2. The command server and the channel initiator must be running.
3. If a listener is listening on multiple addresses or ports, only the address and port combinations with the address, or port, specified are stopped.
4. If a listener is listening on all addresses for a particular port, a stop request for a specific IPADDR with the same port fails.
5. If neither an address nor a port is specified, all addresses and ports are stopped and the listener task ends.

Synonym: STOP LSTR

Parameter descriptions

CMDSCOPE
This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

''
The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.
STOP LISTENER

INDISP
Specifies the disposition of the inbound transmissions that the listener handles. The possible values are:
- **QMGR**
  Handling for transmissions directed to the queue manager. This is the default.
- **GROUP**
  Handling for transmissions directed to the queue-sharing group. This is allowed only if there is a shared queue manager environment.

**IPADDR**
IP address for TCP/IP specified in dotted decimal or alphanumeric form. This is valid only if the transmission protocol (TRPTYPE) is TCP/IP.

**PORT**
The port number for TCP/IP. This is the port number on which the listener is to stop listening. This is valid only if the transmission protocol is TCP/IP.

**TRPTYPE**
Transmission protocol used. This is optional.
- **TCP**
  TCP. This is the default if TRPTYPE is not specified.
- **LU62**
  SNA LU 6.2.

The listener stops in quiesce mode (it disregards any further requests).
STOP QMGR

Use STOP QMGR to stop the queue manager.

**Synonym:** There is no synonym for this command.

**STOP QMGR**

<table>
<thead>
<tr>
<th>CMDSCOPE (&quot;&quot;&quot;)</th>
<th>CMDSCOPE (qmgr-name)</th>
<th>CMDSCOPE (*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODE (QUIESCE)</td>
<td>MODE (FORCE)</td>
<td>MODE (RESTART)</td>
</tr>
</tbody>
</table>

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group.

**Parameter descriptions**

The parameters are optional.

**CMDSCOPE**

This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

""

The command is executed on the queue manager on which it was entered. This is the default value.

\textit{qmgr-name}

The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

\textit{*}

The command is executed on the local queue manager and is also passed to every active queue manager in the queue-sharing group. The effect of this is the same as entering the command on every queue manager in the queue-sharing group.

**MODE**

Specifies whether programs currently being executed are allowed to finish.

**QUIESCE**

Allows programs currently being executed to finish processing. No new program is allowed to start. This is the default.

This option means that all connections to other address spaces must terminate before the queue manager stops. The system operator can determine whether any connections remain by using...
STOP QMGR

the DISPLAY THREAD command, and can cancel remaining connections using z/OS commands.

This option deregisters WebSphere MQ from the MVS automatic restart manager (ARM).

FORCE
Terminates programs currently being executed, including utilities. No new program is allowed to start. This option might cause in-doubt situations.

This option might not work if all the active logs are full, and log archiving has not occurred. In this situation you must issue the z/OS command CANCEL to terminate.

This option deregisters WebSphere MQ from the MVS automatic restart manager (ARM).

RESTART
Terminates programs currently being executed, including utilities. No new program is allowed to start. This option might cause in-doubt situations.

This option might not work if all the active logs are full, and log archiving has not occurred. In this situation you must issue the MVS command CANCEL to terminate.

This option does not deregister WebSphere MQ from ARM, so the queue manager is eligible for immediate automatic restart.
Use STOP TRACE to stop tracing.

**Synonym:** There is no synonym for this command.

**STOP TRACE**

<table>
<thead>
<tr>
<th>Compaq NSK</th>
<th>Compaq OpenVMS</th>
<th>OS/400</th>
<th>OS/2 Warp</th>
<th>UNIX systems</th>
<th>Windows</th>
<th>z/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

**Destination block:**

```
DEST(GTF)
```

**Constraint block:**

```
CLASS(*)
RMID(*)
TNO(*)
USERID(*)
```

**Notes:**

1. Valid only when the queue manager is a member of a queue-sharing group.

**Parameter descriptions**

Each option that you use limits the effect of the command to active traces that were started using the same option, either explicitly or by default, with exactly the same parameter values.

You must specify a trace type or an asterisk. STOP TRACE(*) stops all active traces.
The trace types are:

**ACCTG**
Accounting data (the synonym is A)

*Note:* Accounting data can be lost if the accounting trace is started or stopped while applications are running. For information about the conditions that must be satisfied for successful collection of accounting data, see the [WebSphere MQ for z/OS System Setup Guide](#).

**GLOBAL**
Service data from the entire queue manager (the synonym is G)

**STAT**
Statistical data (the synonym is S)

* All active traces

**CMDSCOPE**
This parameter specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

'' The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

For further descriptions of each type, see “START TRACE” on page 319.

**COMMENT(string)**
Specifies a comment that is reproduced in the trace output record (except in the resident trace tables), and can be used to record why the command was issued.

*string* is any character string. It must be enclosed in single quotation marks if it includes a blank, comma, or special character.

**Destination block**

**DEST**
Limits the action of the STOP TRACE to traces started for particular destinations. More than one value can be specified, but do not use the same value twice. If no value is specified, the list is not limited.

Possible values and their meanings are:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTF</td>
<td>The Generalized Trace Facility</td>
</tr>
<tr>
<td>RES</td>
<td>A wrap-around table residing in the ECSA</td>
</tr>
<tr>
<td>SMF</td>
<td>The System Management Facility</td>
</tr>
<tr>
<td>SRV</td>
<td>A serviceability routine designed for problem diagnosis</td>
</tr>
</tbody>
</table>

See “START TRACE” on page 319 for a list of allowed destinations for each trace type.
Constraint block

CLASS(integer)
Limits the action of the STOP TRACE to traces started for particular classes. See the START TRACE command for a list of allowed classes. A range of classes can be specified as m:n (for example, CLASS(01:03)). You cannot specify a class if you did not specify a trace type.

The default is CLASS(*), which does not limit the command.

RMID(integer)
Limits the action of the STOP TRACE to traces started for particular resource managers. See the START TRACE command for a list of allowed resource manager identifiers.

Do not use this option with the STAT or ACCTG trace type.

If the list of RMIDs includes 231, the tracing for this resource manager is left unchanged if one of the following is true:
- TRACE(GLOBAL) or TRACE(*) is not specified
- The list of destinations does not include RES
- This list of classes does not include 01 or 04

Also, comments are truncated to 120 characters.

The default is RMID(*), which does not limit the command.

TNO(integer)
Limits the action of the STOP TRACE to particular traces, identified by their trace numbers (1 to 32). Up to 8 trace numbers can be used. If more than one number is used, only one value for USERID can be used.

The default is TNO(*), which does not limit the command.

USERID(string)
Limits the action of the STOP TRACE to traces started for particular user ID. Up to 8 user IDs can be used. If more than one user ID is used, only one value can be used for TNO. Do not use this option with STAT.

The default is USERID(*), which does not limit the command.
Use SUSPEND QMGR to inform other queue managers in a cluster that the local,
queue manager is not available for processing and cannot be sent messages, or to
suspend logging and update activity for the queue manager until a subsequent
RESUME QMGR command is issued. Its action can be reversed by the RESUME
QMGR command.

Notes:
1. On UNIX systems, the command is valid only on AIX, HP-UX, Linux, and
Solaris.
2. On z/OS, if you define CLUSTER or CLUSNL, the command fails if the
channel initiator has not been started.
3. On z/OS, if you define CLUSTER or CLUSNL, any errors are reported to the
console on the system where the channel initiator is running; they are not
reported to the system that issued the command.

Synonym: None

Parameter descriptions

CLUSTER(clustername)
The name of the cluster to suspend availability for.

CLUSNL(nlname)
The name of the namelist specifying a list of clusters to suspend
availability for.

LOG Suspends logging and update activity for the queue manager until a
subsequent RESUME request is issued. Any unwritten log buffers are
externalized, a system checkpoint is taken (non-data sharing environment
only), and the BSDS is updated with the high-written RBA before the
update activity is suspended. A highlighted message (CSQJ372I) is issued and remains on the console until update activity has been resumed. Valid on z/OS only.

This option is not allowed when a system quiesce is active by either the ARCHIVE LOG or STOP QMGR command.

Update activity remains suspended until a RESUME QMGR LOG or STOP QMGR command is issued.

This command should not be used during periods of high activity, or for long periods of time. Suspending update activity can cause timing related events such as lock time outs or WebSphere MQ diagnostics dumps when delays are detected.

CMDSCOPE
This parameter applies to z/OS only and specifies how the command is executed when the queue manager is a member of a queue-sharing group.

CMDSCOPE cannot be used for commands issued from the first initialization input data set CSQINP1.

' ' The command is executed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is executed on the queue manager you specify, providing the queue manager is active within the queue-sharing group.

You can specify a queue manager name, other than the queue manager on which the command was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

MODE
Specifies how the suspension of availability is to take effect:

QUIESCE
Other queue managers in the cluster are advised that the local queue manager should not be sent further messages.

FORCE
All inbound channels to other queue managers in the cluster are stopped forcibly. This occurs only if the queue manager has also been forcibly suspended from all other clusters to which the channel belongs.

The MODE keyword is allowed only with CLUSTER or CLUSNL. It is not allowed with LOG.
Appendix. Notices

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

IBM may have patents or pending patent applications covering subject matter described in this information. The furnishing of this information does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

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

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

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England
SO21 2JN.

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

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

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

COPYRIGHT LICENSE:

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

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

- AIX®
- AS/400®
- CICS
- DB2
- IBM
- iSeries
- MQSeries®
- OS/2®
- OS/390®
- OS/400
- RACF
- VTAM
- WebSphere MQ
- z/OS

Linux is a trademark of Linus Torvalds in the United States, or other countries, or both.

Lotus Notes is a trademark of Lotus Development Corporation in the United States, or other countries, or both.

Microsoft®, Windows®, Windows 2000, Windows NT®, and the Windows logo are trademarks of Microsoft Corporation in the United States or other countries, or both.

UNIX is a registered trademark in the United States, or other countries, or both, licensed exclusively through X/Open Company Limited.

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

A
ACCTG parameter
START TRACE 319
STOP TRACE 336
ACTION parameter
RESET CLUSTER 284
RESET TPIPE 289
RESOLVE CHANNEL 290
RESOLVE INDOUBT 293
active thread, display 256
active trace, display list of 258
administrator commands 9
ALCUNIT parameter, SET
ARCHIVE 299
ALTER AUTHINFO command 10
ALTER CFSTRUCT command 11
ALTER CHANNEL command 12
ALTER NAMELIST command 30
ALTER PROCESS command 31
ALTER QALLAS command 43
ALTER QLOCAL command 45
ALTER QMGR parameter 32
ALTER QMODEL command 47
ALTER QREMOTE command 49
ALTER SECURITY command 50
ALTER STGCLASS command 52
ALTER TRACE command 53
ALTTIME parameter
DISPLAY AUTHINFO 175
DISPLAY CFSTRUCT 181
DISPLAY CHANNEL 187
DISPLAY CLUSQmgr 206
DISPLAY NAMELIST 217
ALTTIME parameter (continued)
DISPLAY PROCESS 221
DISPLAY Qmgr 225
DISPLAY QUEUE 244
DISPLAY STGCLASS 253
APPLICID parameter
DEFINE PROCESS 119
DISPLAY PROCESS 221
APPLTAG parameter, DISPLAY
QSTATUS 233
APPLTYPE parameter
DEFINE PROCESS 119
DISPLAY PROCESS 221
DISPLAY QSTATUS 233
archive
display 172
set 298
ARCHIVE LOG command 55
ARCFIX1 parameter, SET
ARCHIVE 299
ARCFIX2 parameter, SET
ARCHIVE 299
ARCRETN parameter, SET
ARCHIVE 299
ARCWRTC parameter, SET
ARCHIVE 299
ARCWTOR parameter, SET
ARCHIVE 300
ASID parameter, DISPLAY
QSTATUS 234
authentication information
alter 10
delete 62
delete definition 155
display attributes 236
ALL parameter
DISPLAY AUTHINFO 175
DISPLAY CFSTRUCT 181
DISPLAY CHANNEL 187
DISPLAY CLUSQmgr 204
DISPLAY NAMELIST 216
DISPLAY PROCESS 220
DISPLAY Qmgr 224
DISPLAY QSTATUS 231
DISPLAY QUEUE 239
DISPLAY SECURITY 250
DISPLAY STGCLASS 253
ALTDATE parameter
DISPLAY AUTHINFO 175
DISPLAY CFSTRUCT 181
DISPLAY CHANNEL 187
DISPLAY CLUSQmgr 206
DISPLAY NAMELIST 217
DISPLAY PROCESS 221
DISPLAY Qmgr 225
DISPLAY QUEUE 244
DISPLAY STGCLASS 253
ALTER AUTHINFO command 10
ALTER CFSTRUCT command 11
ALTER CHANNEL command 12
ALTER NAMELIST command 30
ALTER PROCESS command 31
ALTER QALLAS command 43
ALTER QLOCAL command 45
ALTER QMGR command 32
ALTER QMODEL command 47
ALTER QREMOTE command 49
ALTER SECURITY command 50
ALTER STGCLASS command 52
ALTER TRACE command 53
ALTTIME parameter
DISPLAY AUTHINFO 176
DISPLAY CFSTRUCT 181
DISPLAY CHANNEL 187
DISPLAY CLUSQmgr 206
DISPLAY NAMELIST 217
B
backing up the log 55
BACKUP CFSTRUCT command 58
BATCHES parameter, DISPLAY
CHSTATUS 198
BATCHHB parameter
DEFINE CHANNEL 90
DISPLAY CHANNEL 187
DISPLAY CLUSQmgr 206
BATCHINT parameter
DEFINE CHANNEL 91
DISPLAY CHANNEL 187
DISPLAY CLUSQmgr 206
BLKSIZE parameter, SET ARCHIVE 300
bootstrap data set, recover 270
BOQNAME parameter
DEFINE queues 132
DISPLAY QUEUE 244
BOTHRESH parameter
DEFINE queues 132
DISPLAY QUEUE 245
BROWSE parameter, DISPLAY
QSTATUS 234
BUFF POOL recover 270
buffer pool, defining 66
BUFFERS parameter, DEFINE
BUFF POOL 66
BUFF POOL parameter, DEFINE
BUFF POOL 66
BUFF POOL parameter, DEFINE
PSID 124
BUF S RCV command, DISPLAY
CHSTATUS 198
BUFFSSENT parameter, DISPLAY
CHSTATUS 198
building command scripts 2
building commands
characters with special meanings 2
rules for 1
BYTESRCVD command, DISPLAY
CHSTATUS 198
BYTESSENT parameter, DISPLAY
CHSTATUS 198
C
CANCEL OFFLOAD parameter,
ARCHIVE LOG 55
CATALOG parameter, SET
ARCHIVE 300
CCSID parameter
ALTER Qmgr 34
DISPLAY Qmgr 225
AUTHINFO parameter
DEFINE AUTHINFO 62
DISPLAY AUTHINFO 174
AUTHOREV parameter
ALTER Qmgr 34
DISPLAY Qmgr 225
AUTHTYPE parameter
DEFINE AUTHINFO 63
DISPLAY AUTHINFO 176
auto-definition exit program 34, 225
auto-definition of channels 34, 225
AUTO START parameter
DEFINE CHANNEL 90
DISPLAY CHANNEL 187
CFLEVEL parameter
DEFINE CFSTRUCT 67
DISPLAY CFSTRUCT 182
RECOVER CFSTRUCT 271
cfstruct
alter 11
backup 58
define 67
delete 157
display 181
display status 177
recover 271
© Copyright IBM Corp. 1993, 2002
CFSTRUCT parameter
- BACKUP CFSTRUCT 58
- DEFINE CFSTRUCT 67
- DEFINE queues 133
- DELETE CFSTRUCT 157
- DISPLAY CFSTRUCT 177, 181
- DISPLAY QUEUE 239

CHADEV parameter
- ALTER QMGR 34
- DISPLAY QMGR 225

CHADEXIT parameter
- ALTER QMGR 35
- DISPLAY QMGR 225

channel
-alter parameters 12
-auto-definition 34, 225
-define parameters 70
-delete definition 158
-display 183
- ping 266
- reset 281
-resolve 290
-start 308
-start initiator 312
-start listener 315
-stop 324

channel initiator
- start 312
- stop 328

CHANNEL parameter
- DEFINE CHANNEL 90
- DELETE CHANNEL 158
- DISPLAY CHANNEL 185
- DISPLAY CLUSQMGR 204
- DISPLAY QMGR 224
- DISPLAY QUEUE 240
- REFRESH CLUSTER 284
- RESOLVE CHANNEL 292
- RESOLVE INDOUBT 293
- RESUME QMGR 295
- SUSPEND QMGR 339

CHLDISP parameter
- DISPLAY CHSTATUS 194
- PING CHANNEL 267
- RESET CHANNEL 282
- RESOLVE CHANNEL 291
- START CHANNEL 309
- STOP CHANNEL 324

channel status, displaying 190

channels, rules for names of 6

CHLTYPE parameter
- DEFINE CHANNEL 91
- DISPLAY CHANNEL 187

CHLTYPE parameter, DISPLAY
- QMGR 225

CLUSDATE parameter
- DISPLAY CLUSQMGR 204
- DISPLAY QUEUE 245

CLUSINFO parameter, DISPLAY
- QUEUE 239

CLUSNL parameter
- DEFINE CHANNEL 92
- DEFINE queues 133
- DISPLAY CHANNEL 188
- DISPLAY QUEUE 239, 245
- RESUME QMGR 295
- SUSPEND QMGR 338

CLUSQMGR parameter
- DISPLAY CLUSQMGR 204
- DISPLAY QUEUE 245

CLUSQTM parameter, DISPLAY
- QUEUE 245

CLUSQT parameter, DISPLAY
- REFRESH CLUSTER 274

cluster
- cluster queue manager, display 202

CLUSTER parameter
- DEFINE CHANNEL 92
- DEFINE queues 134
- DISPLAY CHANNEL 188
- DISPLAY DEFINE QUEUE 204, 245
- REFRESH CLUSTER 274
- RESUME QMGR 295
- SUSPEND QMGR 338

CMDSLEVEL parameter, DISPLAY
- QMGR 225

CMDSLEVEL parameter
- ALTER queue manager 36
- ALTER SECURITY 50
- ALTER TRACE 53
- ARCHIVE LOG 56
- BACKUP CFSTRUCT 58
- CLEAR QLOCAL 60
- DEFINE AUTHINFO 63
- DEFINE CHANNEL 92
- DEFINE NAMELIST 115
- DEFINE PROCESS 120
- DEFINE queues 134
- DEFINE STGCLASS 151
- DEFETE AUTHINFO 155
- DEFETE CHANNEL 159
- DEFETE NAMELIST 161
- DEFETE PROCESS 163
- DEFETE queues 168
- DEFETE STGCLASS 170

CMDSLEVEL parameter (continued)
- DISPLAY ARCHIVE 172
- DISPLAY AUTHINFO 175
- DISPLAY CHANNEL 185
- DISPLAY CHSTATUS 194
- DISPLAY CLUSQMGR 204
- DISPLAY DQM 209
- DISPLAY LOG 212
- DISPLAY MAXSMSQS 214
- DISPLAY NAMELIST 216
- DISPLAY PROCESS 220
- DISPLAY QMGR 224
- DISPLAY QSTATUS 231
- DISPLAY QUEUE 240
- DISPLAY SECURITY 249
- DISPLAY STGCLASS 251
- DISPLAY SYSTEM 254
- DISPLAY THREAD 256
- DISPLAY TRACE 259
- DISPLAY USAGE 261
- MOVE QLOCAL 263
- PING CHANNEL 267
- RECOVER BSDS 270
- RECOVER CFSTRUCT 271
- REFRESH CLUSTER 274
- REFRESH QMGR 276
- REFRESH SECURITY 279
- RESUM EXPRESS 283
- RESUM EXPRESS 284
- RESUM EXPRESS 287
- RESUM EXPRESS 287
- RESUM EXPRESS 288
- RESUM EXPRESS 292
- RESUM EXPRESS 293
- RESUM EXPRESS 295
- RESUM EXPRESS 297
- SET ARCHIVE 298
- SET LOG 302
- SET SYSTEM 305
- START CHANNEL 310
- START CHINIT 312
- START TRACE 320
- STOP CHANNEL 324
- STOP CHINIT 328
- STOP LISTENER 331
- STOP QMGR 333
- STOP TRACE 336
- SUSPEND QMGR 339
- SUSPEND QMGR 339

command character set identifier 34

command scripts, building 2

command server
- display status 208
- start 314
- stop 330

command string
- entering quotes 2
- preserving case 2

COMMANDQ parameter, DISPLAY
- QMGR 225

commands
- rules for building 1
- rules for naming objects in 4
- rules for using 1
- synonym 2

COMMAND parameter
- ALTER TRACE 54
- DISPLAY TRACE 259
- START TRACE 320
HBINT parameter
DEFINE CHANNEL  96
DISPLAY CHANNEL 188
DISPLAY CHSTATUS 198
DISPLAY CLUSQMGR 206

IDBACK parameter, SET SYSTEM  306
IDFORE parameter, SET SYSTEM  306
IFCID parameter
ALTER TRACE 54
START TRACE 322
IGQ parameter
ALTER QMGR 37
DISPLAY QMGR 226
IGQAUT parameter
ALTER QMGR 37, 38
DISPLAY QMGR 226
IGUSER parameter, DISPLAY QMGR 226
IMS Tpipe, reset sequence numbers manually 288
in-doubt thread
display 256
resolve manually 293
INCLINT parameter, REFRESH QMGR 277
INDISP parameter
START LISTENER 316
STOP LISTENER 332
INDOUBT parameter
RESOLVE INDOUBT 293
INDOUBT parameter, DISPLAY CHSTATUS 196
INDTYPE parameter
DEFINE queues 137
DISPLAY QUEUE 246
INHIBTEV parameter
ALTER QMGR 39
DISPLAY QMGR 226
INITQ parameter
DEFINE queues 139
DISPLAY QUEUE 246
START CHINIT 313
INPUT parameter, DISPLAY QSTATUS 234
INQUIRE parameter, DISPLAY QSTATUS 234
INTERVAL parameter
ALTER SECURITY 51
DISPLAY SECURITY 250
IPADDR parameter
START LISTENER 316
STOP LISTENER 332
IPROCS parameter
DISPLAY QSTATUS 234
DISPLAY QUEUE 246

JOBNAME parameter, DISPLAY CHSTATUS 198

K
Kaint parameter
DEFINE CHANNEL  96
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
L
LDAPPWD parameter
DEFINE AUTHINFO 64
DISPLAY AUTHINFO 176
LDAPUSER parameter
DEFINE AUTHINFO 64
DISPLAY AUTHINFO 176
LIKE option
DEFINE AUTHINFO 64
DEFINE CHANNEL 97
DEFINE NAMELIST 115
DEFINE PROCESS 121
DEFINE queues 139
DEFINE STGCLASS 152
list of queue names
alter 30
define 114
delete 161
display 215
listener
start 315
stop 331
local queue
alter parameters 45
clear 60
define 127
delete definition 166
display attributes 236
move 263
LOCALLEV parameter
ALTER QMGR 39
DISPLAY QMGR 226
LOCALADDR parameter
DEFINE CHANNEL 99
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
DISPLAY QMGR 227
DISPLAY QUEUE 246
log
archive 55
display 212
set 302
LOG parameter
RESUME QMGR 295
SUSPEND QMGR 338
LOGLOAD parameter, SET SYSTEM  306
LONGRT5 parameter, DISPLAY CHSTATUS 199
LONGRTY parameter
DEFINE CHANNEL 98
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
LONGTMR parameter
DEFINE CHANNEL 98
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
LSTLUWID parameter, DISPLAY CHSTATUS 196
LSTMSGDA parameter, DISPLAY CHSTATUS 199
LSTMSGIT parameter, DISPLAY CHSTATUS 199
LSTSEQNO parameter, DISPLAY CHSTATUS 197
M
MAXARCH parameter, SET LOG  303
MAXDEPTH parameter
DEFINE QLOCAL 140
DISPLAY QUEUE 246
MAXHANDS parameter
ALTER QMGR 39
DISPLAY QMGR 227
maximum number of messages,
define 112
MAXMGS parameter
ALTER QMGR 39
DEFINE CHANNEL 98
DEFINE QLOCAL 140
DISPLAY CHANNEL 188
DISPLAY CHSTATUS 199
DISPLAY CLUSQMGR 206
DISPLAY QMGR 227
DISPLAY QUEUE 246
MAXPRTY parameter, DISPLAY QMGR 227
MAXRTU parameter, SET LOG  303
maxmsgs
define 112
display 214
MAXSMSCS parameter, DEFINE MAXMGS 112
MAXUMSGS parameter, ALTER QMGR 39
MCNAME parameter
DEFINE CHANNEL 99
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
MCASTAT parameter, DISPLAY CHSTATUS 199
MCATYP parameter
DEFINE CHANNEL 99
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
MCAUSER parameter
DEFINE CHANNEL 99
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
MODE parameter
ARCHIVE LOG 56
STOP CHANNEL 326
STOP QMGR 333
SUSPEND QMGR 339
model queue
alter parameters 47
define 129
delete definition 167
display attributes 236
MODENAME parameter
DEFINE CHANNEL 100
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
MOVE QLOCAL command 263
MQSC commands  9
MRDATA parameter
DEFINE CHANNEL 100

348 WebSphere MQ Script (MQSC) Command Reference
MRDATA parameter (continued)
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
MEXIT parameter
DEFINE CHANNEL 100
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
MRITV parameter
DEFINE CHANNEL 100
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
MRRTY parameter
DEFINE CHANNEL 100
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
MSGDATA parameter
DEFINE CHANNEL 100
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
MSGEXIT parameter
DEFINE CHANNEL 100
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
MRTMR parameter
DEFINE CHANNEL 101
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
MSGDLVSQ parameter
DEFINE CHANNEL 101
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
MSGDATA parameter
DEFINE CHANNEL 101
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
MSGDLVSQ parameter
DEFINE CHANNEL 101
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
MREXIT parameter
DEFINE CHANNEL 100
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
MRTMR parameter
DEFINE CHANNEL 101
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
MRITV parameter
DEFINE CHANNEL 100
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
MRRTY parameter
DEFINE CHANNEL 100
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
MREXIT parameter
DEFINE CHANNEL 100
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 206
N
NAMCOUNT parameter, DISPLAY
NAMELIST 218
NAME parameter, REFRESH
QMG 277
name spaces 4
namelist
alter 30
define 114
delete 161
display contents 215
rules for names of 6
NAMELIST parameter
DELETE NAMELIST 161
DISPLAY NAMELIST 215
NAMELIST parameter, DEFINE
NAMELIST 115
NAMES parameter
DEFINE NAMELIST 116
DISPLAY NAMELIST 218
NETPRTY parameter
DEFINE CHANNEL 102
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
NID parameter, RESOLVE
INDOUBT 294
NLTYPE parameter
DEFINE NAMELIST 116
DISPLAY NAMELIST 217
NOHARDENBO parameter, DEFINE
queues 137
NOPURGE parameter, DELETE
QLOCAL 168
NOREPLACE option
DEFINE AUTHINFO 65
DEFINE CHANNEL 105
DEFINE NAMELIST 117
NOREPLACE option (continued)
DEFINE PROCESS 122
DEFINE queues 145
DEFINE STGCCLASS 153
NOSHAPE parameter, DEFINE
queues 147
NOTRIGGER parameter, DEFINE
queues 148
NPMSPEED parameter
DEFINE CHANNEL 102
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
NPMSPEED parameter, DISPLAY
CHSTATUS 199
O
OBJECT parameter, REFRESH
QMG 277
objects, reserved names 6
OPENTYPE parameter, DISPLAY
QSTATUS 231
operator commands 9
OPPROCS parameter
DISPLAY QSTATUS 232
DISPLAY QUEUE 246
OUTPUT parameter, DISPLAY
QSTATUS 234
P
page set
define 124, 151
display usage 261
PARM parameter
START CHINIT 313
START QMGR 318
PASSWORD parameter
DEFINE CHANNEL 103
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
PERFMEV parameter
ALTER QMG 39
DISPLAY QMG 227
PID parameter, DISPLAY QSTATUS 234
PING CHANNEL command 266
PING QMG command 269
PLATFORM parameter, DISPLAY
QMG 227
PORT parameter, STOP LISTENER 332
priority queue, DEFINE queues 141
PRIQTY parameter, SET ARCHIVE 300
process definition
alter 31
define 118
delete 163
display 219
PROCESS parameter
DEFINE PROCESS 119
DEFINE queues 141
DELETE PROCESS 163
DISPLAY PROCESS 220
DISPLAY QUEUE 247
processes, rules for names of 6
PROTECT parameter, SET
ARCHIVE 300
PSBNAME parameter, DISPLAY
QSTATUS 234
PSID parameter
DEFINE PSID 124
DEFINE STGCCLASS 153
DISPLAY QUEUE 240
DISPLAY STGCCLASS 252
DISPLAY USAGE 261
PSTID parameter, DISPLAY
QSTATUS 234
PURGE parameter, DELETE
QLOCAL 168
PUT parameter
DEFINE queues 141
DISPLAY QUEUE 247
PUTAUT parameter
DEFINE CHANNEL 103
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
Q
QALIAS parameter
DELETE queues 167
QDEPTHHI parameter
DEFINE QLOCAL 142
DISPLAY QUEUE 247
QDEPTHLO parameter
DEFINE QLOCAL 142
DISPLAY QUEUE 247
QPHIEV parameter
DEFINE queues 142
DISPLAY QUEUE 247
QDPLOEV parameter
DEFINE queues 142
DISPLAY QUEUE 247
QDPMAXEV parameter
DEFINE queues 143
DISPLAY QUEUE 247
QLOCAL parameter
DELETE queues 167
MOVE QLOCAL 263
QMIN parameter
DISPLAY CLUSQMGR 205
DISPLAY QMG 227
DISPLAY QUEUE 247
RESET CLUSTER 285
QNAME parameter
DEFINE CHANNEL 103
DISPLAY CHANNEL 188
DISPLAY QMG 227
DISPLAY THREAD 257
RESET CLUSTER 285
RESOLVE INDOUBT 294
STOP CHANNEL 327
QMODEL parameter
DELETE queues 167
QTYPE parameter, DISPLAY
QLOCAL 205
QREMOTE parameter
DELETE queues 167
QSGDISP parameter
CLEAR QLOCAL 61
DEFINE CHANNEL 103
DEFINE NAMELIST 116
DEFINE PROCESS 121
DEFINE queues 143
QSGDISP parameter (continued)
DEFINE STGCLASS 153
DELETE AUTHINFO 155
DELETE CHANNEL 159
DELETE NAMELIST 162
DELETE PROCESS 164
DELETE QLOCAL 168
DELETE STGCLASS 171
DISPLAY AUTHINFO 175
DISPLAY CHANNEL 185
DISPLAY NAMELIST 216
DISPLAY PROCESS 220
DISPLAY QSTATUS 232, 235
DISPLAY QUEUE 241
DISPLAY STGCLASS 252
MOVE QLOCAL 264
QSGNAME parameter, DISPLAY
QMGR 227
QSTATS parameter, RESET QSTATS 286
QSTATUS parameter, DISPLAY
QSTATUS 231
QSVCIEV parameter
DEFINE queues 144
DISPLAY QUEUE 247
QSVCINT parameter
DEFINE queues 145
DISPLAY QUEUE 247
QTYPE parameter, DISPLAY
QUEUE 247
queue attributes, display 236
queue manager
alter parameters 32
display parameters 222
ping 269
refresh parameters 276
resume 295
start 318
stop 333
suspend 338
queue manager alias, defining 131
queue names 4
QUEUE parameter
CLEAR QLOCAL 60
DEFINE queues 132
DISPLAY QUEUE 239
queue status
displaying 229
reset 286
queues
reserved names 5
rules for names of 4
QUEUES parameter, RESET CLUSTER 285
QUIESCE parameter, SET ARCHIVE 301
R
RCVDATA parameter
DEFINE CHANNEL 104
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
RCVEXIT parameter
DEFINE CHANNEL 105
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
RCVSEQ parameter, RESET TPIPE 289
RECOVER BSDS command 270
RECOVER CFSTRUCT command 271
RECOVER parameter
DEFINE CFSTRUCT 68
DISPLAY CFSTRUCT 182
REFRESH CLUSTER command 273
REFRESH QMGR command 276
REFRESH SECURITY command 279
remote queue
alter parameters 49
define 131
delete local definition 167
display attributes 236
REMOTEEV parameter
ALTER QMGR 40
DISPLAY QMGR 227
REPLACE option
DEFINE AUTHINFO 65
DEFINE CHANNEL 105
DEFINE NAMELIST 117
DEFINE PROCESS 122
DEFINE queues 145
DEFINE STGCLASS 153
reply-to queue alias, defining 131
REPOS parameter
ALTER QMGR 40
DISPLAY QMGR 227
REFRESH CLUSTER 274
REPOSNL parameter
ALTER QMGR 40
DISPLAY QMGR 227
reserved names
objects 6
queues 5
RESET CHANNEL command 281
RESET CLUSTER command 284
RESET QSTATS command 286
RESET TPIPE command 288
RESOLVE CHANNEL command 290
RESOLVE INDOUBT command 293
RESUME QMGR command 295
RETINTVL parameter
DEFINE queues 146
DISPLAY QUEUE 247
RMD parameter
DISPLAY TRACE 260
START TRACE 322
STOP TRACE 337
RNAME parameter
DEFINE QREMOTE 146
DISPLAY QUEUE 247
RQDATA parameter
DEFINE QREMOTE 146
DISPLAY QUEUE 247
RQNAME parameter
DEFINE QREMOTE 146
DISPLAY QUEUE 247
rules for using commands 1
RVERIFY SECURITY command 297
SCYDATA parameter (continued)
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
SCYEXIT parameter
DEFINE CHANNEL 105
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
SECTYPE parameter, SET ARCHIVE 301
security
alter parameters 50
display parameters 249
rebuild 279
refresh 279
reverify 297
SECURITY parameter
RVERIFY SECURITY 297
SENDDATA parameter
DEFINE CHANNEL 106
DISPLAY CHANNEL 188
DISPLAY CLUSQMGR 207
SENDEXIT parameter
DEFINE CHANNEL 106
DISPLAY CHANNEL 189
DISPLAY CLUSQMGR 207
SENDSEQ parameter, RESET TPIPE 289
SEQNUM parameter, RESET CHANNEL 283
sequence numbers, resetting on an IMS Tpipe 288
SEQWRAP parameter
DEFINE CHANNEL 106
DISPLAY CHANNEL 189
DISPLAY CLUSQMGR 207
SET ARCHIVE command 298
SET LOG command 302
SET parameter, DISPLAY QSTATS 235
SET SYSTEM command 305
SHARE parameter
DEFINE queues 147
DISPLAY QUEUE 247
SHARED parameter, STOP CHINIT 328
SHORT parameter, DISPLAY
CHSTATUS 195
SHORTRTS parameter, DISPLAY
CHSTATUS 199
SHORTTRY parameter
DEFINE CHANNEL 107
DISPLAY CHANNEL 189
DISPLAY CLUSQMGR 207
SHORTTMR parameter
DEFINE CHANNEL 107
DISPLAY CHANNEL 189
DISPLAY CLUSQMGR 207
SSLCAUTH parameter
DEFINE CHANNEL 107
DISPLAY CHANNEL 189
DISPLAY CLUSQMGR 207
SSLCIPH parameter
ALTER QMGR 40
DISPLAY QMGR 227
SSLCRNL parameter
ALTER QMGR 40
DISPLAY QMGR 227
SSLCRYP parameter
ALTER QMGR 40
DISPLAY QMGR 227

350 WebSphere MQ Script (MQSC) Command Reference
Sending your comments to IBM

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

To make comments about the functions of IBM products or systems, talk to your IBM representative or to your IBM authorized remarketer.

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

You can send your comments to IBM in any of the following ways:

- By mail, to this address:
  User Technologies Department (MP095)
  IBM United Kingdom Laboratories
  Hursley Park
  WINCHESTER,
  Hampshire
  SO21 2JN
  United Kingdom
- By fax:
  – From outside the U.K., after your international access code use 44–1962–816151
  – From within the U.K., use 01962–816151
- Electronically, use the appropriate network ID:
  – IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
  – IBMLink™: HURSLEY(IDRCF)
  – Internet: idrcf@hursley.ibm.com

Whichever method you use, ensure that you include:

- The publication title and order number
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.