WebSphere MQ

Script (MQSC) Command Reference

Version 7.0

SC34-6941-00

WebSphere MQ

Script (MQSC) Command Reference

Version 7.0

SC34-6941-00

Note
Before using this information and the product it supports, be sure to read the general information under notices at the back of this book.

First edition (April 2008)

This edition of the book applies to the following products:

- IBM WebSphere MQ, Version 7.0
- IBM WebSphere MQ for z/OS, Version 7.0

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

Unless otherwise stated, the information also applies to the following products:

- MQSeries for Compaq NonStop Kernel, V5.1
- WebSphere MQ for HP OpenVMS, V5.3

© Copyright International Business Machines Corporation 1999, 2008. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Tables vii

Chapter 1. Using MQSC commands . . . 1

Rules for using MQSC commands 1
Characters with special meanings 2
Generic values 3
Building command scripts 3
Using commands in z/OS 4
Rules for naming WebSphere MQ objects 5
Queue names 6
Other object names 7
Syntax diagrams 9
How to read railroad diagrams 9

Chapter 2. The MQSC commands . . . 11

ALTER AUTHINFO 12
Parameter descriptions for ALTER AUTHINFO 12
ALTER BUFFPOOL 13
Parameter descriptions for ALTER BUFFPOOL 13
Usage notes for ALTER BUFFPOOL 13
ALTER CFSTRUCT 14
Parameter descriptions for ALTER CFSTRUCT . . 14
ALTER CHANNEL 15
Sender channel 16
Server channel 19
Receiver channel 22
Requester channel 25
Client-connection channel 28
Server-connection channel 30
Cluster-sender channel 32
Cluster-receiver channel 35
Parameter descriptions for ALTER CHANNEL . 37
ALTER LISTENER 38
Parameter descriptions for ALTER LISTENER . . 39
ALTER NAMELIST 40
Parameter descriptions for ALTER NAMELIST . 40
ALTER PROCESS 41
Parameter descriptions for ALTER PROCESS . . 42
ALTER PSID 43
Parameter descriptions for ALTER PSID 43
ALTER QMGR 44
Parameter descriptions for ALTER QMGR . . . 49
ALTER queues 78
ALTER QALIAS 79
ALTER QLOCAL 81
ALTER QMODEL 85
ALTER QREMOTE 88
Parameter descriptions for DEFINE QUEUE and ALTER QUEUE 89
ALTER SECURITY 112
Parameter descriptions for ALTER SECURITY 112
ALTER SERVICE 114
Parameter descriptions for ALTER SERVICE . . 114
ALTER STGCLASS 115
Parameter descriptions for ALTER STGCLASS 115
ALTER SUB 116
Parameter descriptions for ALTER SUB 117
Usage notes for ALTER SUB 120
ALTER TOPIC 121
Parameter descriptions for ALTER TOPIC . . . 122
ALTER TRACE 123
Parameter descriptions for ALTER TRACE . . 123
ARCHIVE LOG 125
Parameter descriptions for ARCHIVE LOG . . 125
Usage notes for ARCHIVE LOG 127
BACKUP CFSTRUCT 128
Keyword and parameter descriptions for BACKUP CFSTRUCT 128
Usage notes for BACKUP CFSTRUCT 129
CLEAR QLOCAL 130
Parameter descriptions for CLEAR QLOCAL 130
Usage notes for CLEAR QLOCAL 131
CLEAR TOPICSTR 131
Parameter descriptions for CLEAR TOPICSTR 132
Usage notes for CLEAR TOPICSTR 133
DEFINE AUTHINFO 134
Parameter descriptions for DEFINE AUTHINFO 134
Usage Notes for DEFINE AUTHINFO 138
DEFINE BUFFPOOL 139
Parameter descriptions for DEFINE BUFFPOOL 139
DEFINE CFSTRUCT 141
Keyword and parameter descriptions for DEFINE CFSTRUCT 141
Usage notes for DEFINE CFSTRUCT 143
DEFINE CHANNEL 144
Sender channel 145
Server channel 149
Receiver channel 153
Requester channel 156
Client-connection channel 160
Server-connection channel 163
Cluster-sender channel 166
Cluster-receiver channel 170
Parameter descriptions for DEFINE CHANNEL 173
DEFINE LISTENER 208
Parameter descriptions for DEFINE LISTENER 209
DEFINE LOG 212
Parameter descriptions for DEFINE LOG . . . 212
Usage note for DEFINE LOG 213
DEFINE MAXSMSGS 214
Parameter descriptions for DEFINE MAXSMSGS 214
DEFINE NAMELIST 216
Parameter descriptions for DEFINE NAMELIST 217
DEFINE PROCESS 221
Parameter descriptions for DEFINE PROCESS 222
DEFINE PSID 227
Parameter descriptions for DEFINE PSID . . . 227
Usage notes for DEFINE PSID 228
DEFINE QUEUE 229
DEFINE QALIAS 230
DEFINE QLOCAL 232

© Copyright IBM Corp. 1999, 2008

DEFINE QMODEL 236
DEFINE QREMOTE 240
Parameter descriptions for DEFINE QUEUE and ALTER QUEUE. 241
Usage notes for DEFINE queues 263
DEFINE SERVICE 265
Parameter descriptions for DEFINE SERVICE 266
DEFINE STGCLASS 268
Parameter descriptions for DEFINE STGCLASS 268
Usage notes for DEFINE STGCLASS 271
DEFINE SUB 273
Parameter descriptions for DEFINE SUB . . . 274
Usage notes for DEFINE SUB 277
DEFINE TOPIC. 278
Parameter descriptions for DEFINE TOPIC . . 279
Usage notes for DEFINE TOPIC 287
DELETE AUTHINFO. 288
Parameter descriptions for DELETE AUTHINFO 288
DELETE BUFFPOOL 290
Parameter descriptions for DELETE BUFFPOOL 290
Usage note for DELETE BUFFPOOL. 290
DELETE CFSTRUCT 291
Keyword and parameter descriptions for DELETE CFSTRUCT 291
Usage notes for DELETE CFSTRUCT 291
DELETE CHANNEL 292
Parameter descriptions for DELETE CHANNEL 292
DELETE LISTENER 295
Keyword and parameter descriptions for DELETE LISTENER 295
Usage notes for DELETE LISTENER. 295
DELETE NAMELIST 296
Parameter descriptions for DELETE NAMELIST 296
DELETE PROCESS 298
Parameter descriptions for DELETE PROCESS 298
DELETE PSID 300
Parameter descriptions for DELETE PSID . . . 300
Usage notes for DELETE PSID 300
DELETE queues 301
DELETE QALIAS 301
DELETE QLOCAL 301
DELETE QMODEL 302
DELETE QREMOTE 302
Parameter descriptions for DELETE queues . . 303
DELETE SERVICE. 306
Keyword and parameter descriptions for DELETE SERVICE. 306
Usage notes for DELETE SERVICE 306
DELETE SUB 307
Parameter descriptions for DELETE SUB . . . 307
DELETE STGCLASS 309
Parameter descriptions for DELETE STGCLASS 309
DELETE TOPIC 311
Parameter descriptions for DELETE TOPIC . . 311
Usage note 312
DISPLAY ARCHIVE 313
Parameter descriptions for DISPLAY ARCHIVE 313
Usage notes for DISPLAY ARCHIVE 314
DISPLAY AUTHINFO 315
Parameter descriptions for DISPLAY AUTHINFO 315
DISPLAY CFSTATUS 319
Keyword and parameter descriptions for DISPLAY CFSTATUS 319
DISPLAY CFSTRUCT. 325
Keyword and parameter descriptions for DISPLAY CFSTRUCT. 325
Requested parameters 326
Usage notes for DISPLAY CFSTRUCT 327
DISPLAY CHANNEL. 328
Parameter descriptions for DISPLAY CHANNEL 330
DISPLAY CHINIT 341
Parameter descriptions for DISPLAY CHINIT 341
Usage notes for DISPLAY CHINIT 341
DISPLAY CHSTATUS. 343
Parameter descriptions for DISPLAY CHSTATUS 346
Usage notes for DISPLAY CHSTATUS 362
DISPLAY CLUSQMGR 364
Parameter descriptions for DISPLAY CLUSQMGR 367
DISPLAY CMDSERV 374
Usage notes for DISPLAY CMDSERV 374
DISPLAY CONN 375
Parameter descriptions for DISPLAY CONN . . 378
DISPLAY GROUP 389
Parameter descriptions for DISPLAY GROUP 389
Usage notes for DISPLAY GROUP 389
DISPLAY LISTENER 390
Keyword and parameter descriptions for DISPLAY LISTENER 391
DISPLAY LOG 394
Parameter descriptions for DISPLAY LOG . . . 394
Usage notes for DISPLAY LOG 394
DISPLAY LSSTATUS 396
Keyword and parameter descriptions for DISPLAY LSSTATUS 397
DISPLAY MAXSMSGS 400
Parameter descriptions for DISPLAY MAXSMSGS. 400
DISPLAY NAMELIST. 401
Parameter descriptions for DISPLAY NAMELIST 402
DISPLAY PROCESS 406
Parameter descriptions for DISPLAY PROCESS 407
DISPLAY PUBSUB. 410
Parameter descriptions for DISPLAY PUBSUB 410
DISPLAY QMGR 414
Parameter descriptions for DISPLAY QMGR . . 420
DISPLAY QMSTATUS 431
Parameter descriptions for DISPLAY QMSTATUS 431
DISPLAY QSTATUS 433
Parameter descriptions for DISPLAY QSTATUS 435
DISPLAY QUEUE 446
Parameter descriptions for DISPLAY QUEUE 449
DISPLAY SBSTATUS 462
Parameter descriptions for DISPLAY SBSTATUS 463
DISPLAY SECURITY 466
Parameter descriptions for DISPLAY SECURITY 466
DISPLAY SERVICE 468
Keyword and parameter descriptions for DISPLAY SERVICE 468
DISPLAY STGCLASS 471

Parameter descriptions for DISPLAY STGCLASS 472
DISPLAY SUB 476
Parameter descriptions for DISPLAY SUB . . . 477
Usage notes for DISPLAY SUB 482
DISPLAY SVSTATUS 483
Keyword and parameter descriptions for DISPLAY SVSTATUS 483
DISPLAY SYSTEM. 486
Parameter descriptions for DISPLAY SYSTEM 486
Usage notes for DISPLAY SYSTEM 486
DISPLAY THREAD 488
Parameter descriptions for DISPLAY THREAD 488
DISPLAY TOPIC 491
Parameter descriptions for DISPLAY TOPIC . . 492
DISPLAY TPSTATUS 497
Parameter descriptions for DISPLAY TPSTATUS 498
Usage notes for DISPLAY TPSTATUS 503
DISPLAY TRACE 504
Parameter descriptions for DISPLAY TRACE 505
DISPLAY USAGE 507
Parameter descriptions for DISPLAY USAGE 507
MOVE QLOCAL 509
Parameter descriptions for MOVE QLOCAL . . 509
Usage notes for MOVE QLOCAL. 510
PING CHANNEL 512
Parameter descriptions for PING CHANNEL 512
PING QMGR 515
RECOVER BSDS 516
Parameter descriptions for RECOVER BSDS . . 516
RECOVER CFSTRUCT 517
Keyword and parameter descriptions for RECOVER CFSTRUCT 517
Usage notes for RECOVER CFSTRUCT. . . . 518
REFRESH CLUSTER 519
Parameter descriptions for REFRESH CLUSTER 520
Usage Notes for REFRESH CLUSTER 521
REFRESH QMGR 522
Parameter descriptions for REFRESH QMGR 522
REFRESH SECURITY. 525
Parameter descriptions for REFRESH SECURITY 526
Usage notes for REFRESH SECURITY 528
RESET CHANNEL 530
Parameter descriptions for RESET CHANNEL 531
RESET CLUSTER 533
Parameter descriptions for RESET CLUSTER 533
Usage notes for RESET CLUSTER 534
RESET QMGR 535
Parameter descriptions for RESET QMGR . . . 535
Usage notes for RESET QMGR 536
RESET QSTATS. 537
Parameter descriptions for RESET QSTATS . . 537
Usage notes for RESET QSTATS 538
RESET TPIPE 539
Parameter descriptions for RESET TPIPE . . . 539
RESOLVE CHANNEL 541
Parameter descriptions for RESOLVE CHANNEL 541
Usage notes for RESOLVE CHANNEL 543
RESOLVE INDOUBT 544
Parameter descriptions for RESOLVE INDOUBT 544
RESUME QMGR 546
Parameter descriptions for RESUME QMGR . . 547
RVERIFY SECURITY 548
Parameter descriptions for RVERIFY SECURITY 548
SET ARCHIVE 550
Parameter descriptions for SET ARCHIVE. . . 551
SET LOG 555
Parameter descriptions for SET LOG 555
SET SYSTEM 558
Parameter descriptions for SET SYSTEM . . . 558
START CHANNEL 561
Parameter descriptions for START CHANNEL 561
START CHINIT. 565
WebSphere MQ for z/OS 565
WebSphere MQ on other platforms 565
Parameter descriptions for START CHINIT . . 565
START CMDSERV. 567
Usage notes for START CMDSERV 567
START LISTENER. 568
WebSphere MQ for z/OS 568
WebSphere MQ on other platforms 569
Parameter descriptions for START LISTENER 569
START QMGR 571
Parameter descriptions for START QMGR . . . 571
START SERVICE 572
Parameter descriptions for START SERVICE . . 572
START TRACE 573
Parameter descriptions for START TRACE. . . 574
STOP CHANNEL 578
Parameter descriptions for STOP CHANNEL 579
Usage notes for STOP CHANNEL 582
STOP CHINIT 583
Parameter descriptions for STOP CHINIT . . . 583
Usage notes for STOP CHINIT 584
STOP CMDSERV 585
Usage notes for STOP CMDSERV 585
STOP CONN 586
Parameter descriptions for STOP CONN . . . 586
STOP LISTENER 587
Parameter descriptions for STOP LISTENER . . 587
STOP QMGR 589
Parameter descriptions for STOP QMGR . . . 589
STOP SERVICE. 591
Parameter descriptions for STOP SERVICE . . 591
STOP TRACE 592
Parameter descriptions for STOP TRACE . . . 593
SUSPEND QMGR 596
Parameter descriptions for SUSPEND QMGR 596

Notices 599

Index 603

Sending your comments to IBM . . . 615

Tables

1. How to read railroad diagrams 9
2. DEFINE and ALTER QUEUE parameters 89
3. 132
4. DEFINE and ALTER CHANNEL parameters 173
5. Examples of how the LOCLADDR parameter can be used 187
6. How the IP stack to be used for communication is determined 188
7. CipherSpecs that can be used with WebSphere MQ SSL support 202
8. DEFINE and ALTER QUEUE parameters 241
9. Parameters that can be returned by the DISPLAY CHANNEL command 334
10. CHLDISP and CMDSCOPE for DISPLAY CHSTATUS CURRENT 362
11. CHLDISP and CMDSCOPE for DISPLAY CHSTATUS SHORT 362
12. CHLDISP and CMDSCOPE for DISPLAY CHSTATUS SAVED 362
13. Parameters that can be returned by the DISPLAY QUEUE command 455
14. CHLDISP and CMDSCOPE for PING CHANNEL 514
15. CHLDISP and CMDSCOPE for RESET CHANNEL 531
16. CHLDISP and CMDSCOPE for RESOLVE CHANNEL 542
17. CHLDISP and CMDSCOPE for START CHANNEL 562
18. Destinations allowed for each trace type 575
19. Constraints allowed for each trace type 576
20. Descriptions of trace events and classes 576
21. Resource Manager identifiers that are allowed 577
22. CHLDISP and CMDSCOPE for STOP CHANNEL 580

© Copyright IBM Corp. 1999, 2008

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, see the WebSphere MQ Programmable Command Formats and Administration Interface manual.

This section describes:

- "Rules for using MQSC commands"
- "Rules for naming WebSphere MQ objects" on page 5
- How to read dotted decimal diagrams

The general format of the commands is shown in Chapter 2, "The MQSC commands," on page 11.

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

© Copyright IBM Corp. 1999, 2008

- Generic values ending with an asterisk (on i5/OS® these must be enclosed in single quotation marks)
- A single asterisk (for example, TRACE(*)) (on i5/OS 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.

- On platforms other than z/OS, 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 `(' ')`. The exception to this is if the attribute being used is one of the following:
  - `TOPICSTR`
  - `SUB`
  - `USERDATA`
  - `SELECTOR`

  then two single quotation marks with no space are interpreted as a zero-length string.

  On z/OS, if you want a quoted blank space, you must enter it as such (' '). A string containing no characters ('') is the same as entering (), the rules for which are explained below.
- 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. Anything that is not contained within quotation marks is 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:

	Blanks are used as separators. Multiple blanks are equivalent to a single blank, except in strings that have quotation marks (') round them.
,	Commas are used as separators. Multiple commas are equivalent to a single comma, except in strings that have quotation marks (') round them.
'	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.
''	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.

=	On z/OS, an equals sign indicates the start of a parameter value which is ended by a comma or blank.
(	An open parenthesis indicates the beginning of a parameter value or list of values.
)	A close parenthesis indicates the end of a parameter value or list of values.
:	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.
*	An asterisk means “all”. For example, DISPLAY TRACE (*) means display all traces, and DISPLAY QUEUE (PAY*) means display all queues whose names begin with PAY.

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.

Generic values

Wherever a parameter can have a generic value, it is entered ending with an asterisk (*), for example ABC*. A generic value means 'all values beginning with'; so ABC* means 'all values beginning with ABC'.

If characters that require quotes are used in the value, the asterisk must be placed inside the quotes, thus 'abc*'. The asterisk must be the last or only character in the value.

The question mark (?) and colon (:) are not allowed in generic values.

Building command scripts

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

- The CSQINP1, CSQINP2, and CSQINPX initialization data sets or the CSQUTIL batch utility on z/OS
- The STRMQM command on i5/OS
- The **runmqsc** command on HP OpenVMS, Compaq NSK, Linux®, 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, i5/OS, Solaris, and Windows, each line can be of any length up to a maximum of 2048 characters.
  - On other UNIX systems, and HP 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, i5/OS, Solaris, and Windows you can use a semicolon character (;) to terminate a command, even if you have entered a plus sign (+) 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.
- When running MQSC commands interactively, you end the interactive session by typing the END command. This applies to:
  - Windows and UNIX systems, where you start the interactive session by entering **runmqsc**
  - i5/OS systems, where you start the interactive session from the WRKMWM panel
- On Windows, if certain special characters such as the pound sign (£) and the logical NOT (¬) are used in a command script (for example, as part of an object description), they will be displayed differently in the output from a command such as DISPLAY QLOCAL.

Using commands in z/OS

As described in the WebSphere MQ for z/OS Concepts and Planning Guide, commands can be issued from:

- The z/OS console or equivalent
- The initialization input data sets CSQINP1, CSQINP2, and CSQINPX

- The CSQUTIL batch utility
- Suitably authorized applications, sending commands as messages to the SYSTEM.COMMAND.INPUT queue

However, not all commands can be issued from all these sources. Commands can be classified according to whether they can be issued from:

1 CSQINP1
2 CSQINP2
C The z/OS console
R The command server and command queue, by means of CSQUTIL, CSQINPX, or applications

Within the command descriptions that follow, these sources are identified by the use of the characters 1, 2, C, and R respectively in the z/OS column of the table at the top of each command description.

Rules for naming WebSphere MQ objects

WebSphere MQ authentication information, channel, client channel, listener, namelist, process, queue, service, storage class and topic 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.

Note:

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 i5/OS 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:

SYSTEM.ADMIN.ACTIVITY.QUEUE	Queue for activity reports
SYSTEM.ADMIN.CHANNEL.EVENT	Queue for channel events
SYSTEM.ADMIN.COMMAND.EVENT	Queue for command events
SYSTEM.ADMIN.COMMAND.QUEUE	Queue to which PCF command messages are sent
SYSTEM.ADMIN.CONFIG.EVENT	Queue for configuration events
SYSTEM.ADMIN.PERFM.EVENT	Queue for performance events
SYSTEM.ADMIN.QMGR.EVENT	Queue for queue manager events
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	Queue for trace-route reply messages
SYSTEM.AUTH.DATA.QUEUE	The queue that holds access control lists for the queue manager. (Not for z/OS)
SYSTEM.CHANNEL.INITQ	Queue used for distributed queuing on z/OS
SYSTEM.CHANNEL.SYNCQ	Queue used for distributed queuing on z/OS
SYSTEM.CICS.INITIATION.QUEUE	Queue used for triggering (**not** for z/OS)
SYSTEM.CLUSTER.COMMAND.QUEUE	Queue used to communicate repository changes between queue managers (AIX, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS only)
SYSTEM.CLUSTER.REPOSITORY.QUEUE	Queue used to hold information about the repository (AIX, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS only)
SYSTEM.CLUSTER.TRANSMIT.QUEUE	Transmission queue for all destinations managed by cluster support (AIX, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS only)
SYSTEM.COMMAND.INPUT	Queue to which command messages are sent on z/OS
SYSTEM.COMMAND.REPLY.MODEL	Model queue definition for command replies (for z/OS)

SYSTEM.DEAD.LETTER.QUEUE	Dead-letter queue (**not** for z/OS)
SYSTEM.DEFAULT.ALIAS.QUEUE	Default alias queue definition
SYSTEM.DEFAULT.INITIATION.QUEUE	Queue used to trigger a specified process (**not** for z/OS)
SYSTEM.DEFAULT.LOCAL.QUEUE	Default local queue definition
SYSTEM.DEFAULT.MODEL.QUEUE	Default model queue definition
SYSTEM.DEFAULT.REMOTE.QUEUE	Default remote queue definition
SYSTEM.DURABLE.SHARED.SUBSCRIBER.QUEUE	A local queue used to hold a persistent copy of the durable subscriptions in the queue manager
SYSTEM.DURABLE.SUBSCRIBER.QUEUE	A local queue used to hold a persistent copy of the durable subscriptions in the queue manager
SYSTEM.HIERARCHY.STATE	Queue used to hold information about the state of inter-queue manager relationships in a publish/subscribe hierarchy
SYSTEM.INTER.QMGR.CONTROL	Queue used in a pub/sub hierarchy to receive requests from a remote queue manager to create a proxy subscription
SYSTEM.INTER.QMGR.PUBS	Queue used in a pub/sub hierarchy to receive publications from a remote queue manager
SYSTEM.INTER.QMGR.FANREQ	Queue used in a pub/sub hierarchy to process requests to create a proxy subscription on a remote queue manager
SYSTEM.MQEXPLORER.REPLY.MODEL	Model queue definition for replies for MQ Explorer
SYSTEM.MQSC.REPLY.QUEUE	Model queue definition for MQSC command replies (**not** for z/OS)
SYSTEM.QSG.CHANNEL.SYNCQ	Shared local queue used for storing messages that contain the synchronization information for shared channels (z/OS **only**)
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**)
SYSTEM.RETAINED.PUB.QUEUE	A local queue used to hold a copy of each retained publication in the queue manager.

Other object names

Processes, namelists, clusters, topics, 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. CF structures can have names up to 12 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
SYSTEM.AUTO.RECEIVER	Default receiver channel for auto definition (AIX, HP-UX, Linux, Solaris, and Windows only)
SYSTEM.AUTO.SVRCONN	Default server-connection channel for auto definition (AIX, HP-UX, Linux, z/OS, i5/OS, Solaris, and Windows only)
SYSTEM.BASE.TOPIC	Base topic for ASPARENT resolution. If a particular administrative topic object has no parent administrative topic objects, any ASPARENT attributes are inherited from this object
SYSTEM.DEF.CLNTCONN	Default client-connection channel definition
SYSTEM.DEF.CLUSRCVR	Default cluster-receiver channel definition (AIX, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS only)
SYSTEM.DEF.CLUSSDR	Default cluster-sender channel definition (AIX, HP-UX, Linux, i5/OS, 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.LISTENER.LU62	Default SNA listener (Windows only)
SYSTEM.DEFAULT.LISTENER.NETBIOS	Default NetBIOS listener (Windows only)
SYSTEM.DEFAULT.LISTENER.SPX	Default SPX listener (Windows only)
SYSTEM.DEFAULT.LISTENER.TCP	Default TCP/IP listener (AIX, HP-UX, Linux, i5/OS, Solaris, and Windows only)
SYSTEM.DEFAULT.NAMELIST	Default namelist definition (AIX, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS only)
SYSTEM.DEFAULT.PROCESS	Default process definition
SYSTEM.DEFAULT.SEVICE	Default service (AIX, HP-UX, Linux, i5/OS, Solaris, and Windows only)
SYSTEM.DEFAULT.TOPIC	Default topic definition
SYSTEM.QPUBSUB.QUEUE.NAMELIST	A list of queues for the Queued Publish/Subscribe interface to monitor
SYSTEMST	Default storage class definition (z/OS only)

Syntax diagrams

The syntax for commands is presented in the form of a diagram. There are two types of syntax diagram: railroad diagrams and dotted decimal diagrams.

Either type of syntax diagram tells you what you can do with the particular command, and indicates relationships between different options and, sometimes, different values of an option. Railroad diagrams are a visual format suitable for sighted users. Dotted decimal diagrams are text-based diagrams that are more helpful for blind or partially-sighted users.

Only railroad diagrams are available in PDFs: dotted decimal diagrams are only avaliable in the Information Center.

How to read railroad diagrams

Each railroad 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 railroad diagram from left to right and from top to bottom, following the direction of the arrows.

Other conventions used in railroad diagrams are:

Table 1. How to read railroad diagrams

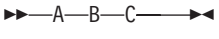
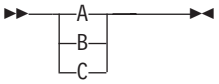
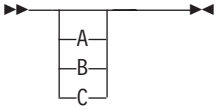
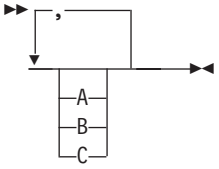
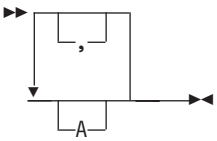
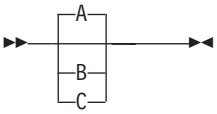
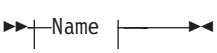
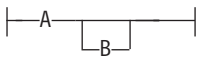
Convention	Meaning
	You must specify values `A`, `B`, and `C`. Required values are shown on the main line of a railroad diagram.
	You may specify value `A`. Optional values are shown below the main line of a railroad diagram.
	Values `A`, `B`, and `C` are alternatives, one of which you must specify.
	Values `A`, `B`, and `C` are alternatives, one of which you might specify.
	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.

Table 1. How to read railroad diagrams (continued)

Convention	Meaning
	You might specify value `A` multiple times. The separator in this example is optional.
	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.
 **Name:** 	The railroad fragment `Name` is shown separately from the main railroad diagram.
Punctuation and uppercase values	Specify exactly as shown.

Chapter 2. The MQSC commands

This chapter describes, in alphabetic order, all the MQSC commands that can be issued by operators and administrators.

© Copyright IBM Corp. 1999, 2008

ALTER AUTHINFO

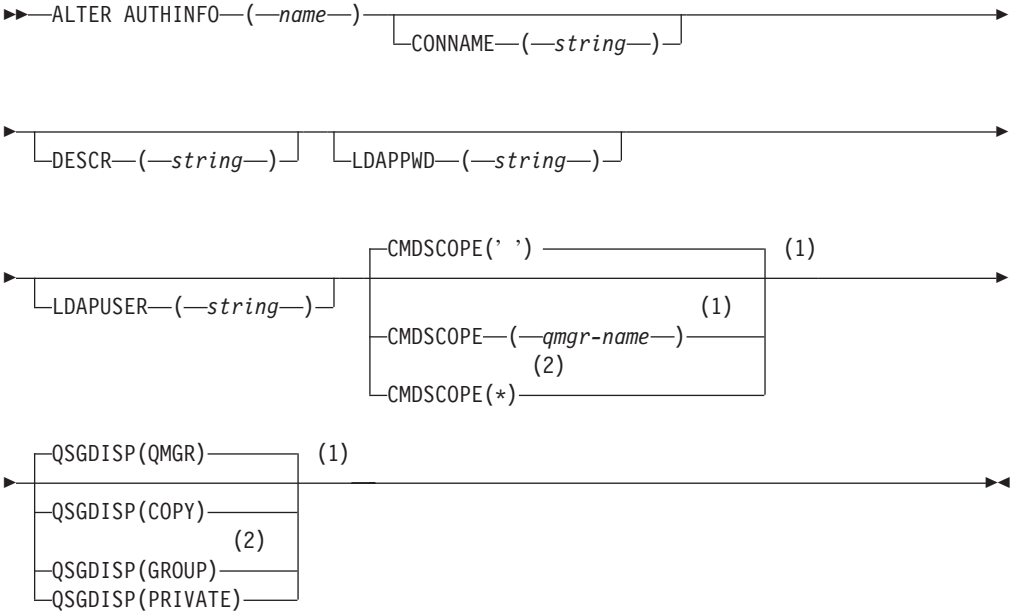
Use the MQSC command ALTER AUTHINFO to alter an authentication information object.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

Authentication information objects contain the definitions required to perform Certificate Revocation List (CRL) checking using LDAP servers, except on i5/OS where these are defined by the Digital Certificate Manager for each Certification Authority.

Synonym: ALT AUTHINFO

ALTER AUTHINFO



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 ALTER AUTHINFO

For a description of the parameters see "DEFINE AUTHINFO" on page 134.

ALTER BUFFPOOL

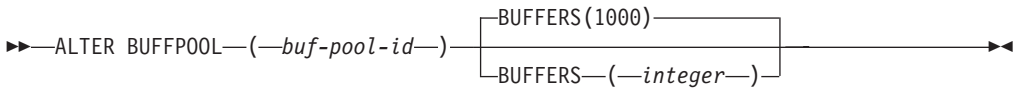
Use the MQSC command ALTER BUFFPOOL to dynamically add buffers to a predefined buffer pool, or remove buffers from a predefined buffer pool.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: ALT BP

ALTER BUFFPOOL



Parameter descriptions for ALTER BUFFPOOL

Parameters for the ALTER BUFFPOOL command.

(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 value 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.

Usage notes for ALTER BUFFPOOL

1. Buffers are added or removed according to whether the value is more than or less than the current allocation (which can be shown by the DISPLAY USAGE command).
2. If there is insufficient storage to add the requested number, as many as possible are added.
3. The command is processed asynchronously. Message CSQP023I is sent to the console when the command is complete.

ALTER CFSTRUCT

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

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

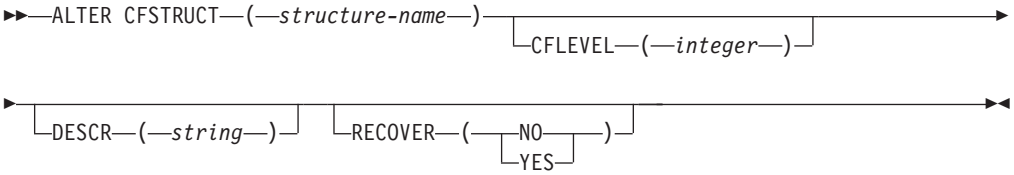
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

1. This command cannot specify the CF administration structure (CSQ_ADMIN).
2. This command is valid only when the queue manager is a member of a queue-sharing group.

Synonym: ALT CFSTRUCT

ALTER CFSTRUCT



Parameter descriptions for ALTER CFSTRUCT

For a description of the parameters see "DEFINE CFSTRUCT" on page 141.

ALTER CHANNEL

Use the MQSC command ALTER CHANNEL to alter the parameters of a channel.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

1. For cluster-sender channels, you can only alter channels that have been created manually.
2. 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 530 for information about the SEQNUM parameter.)

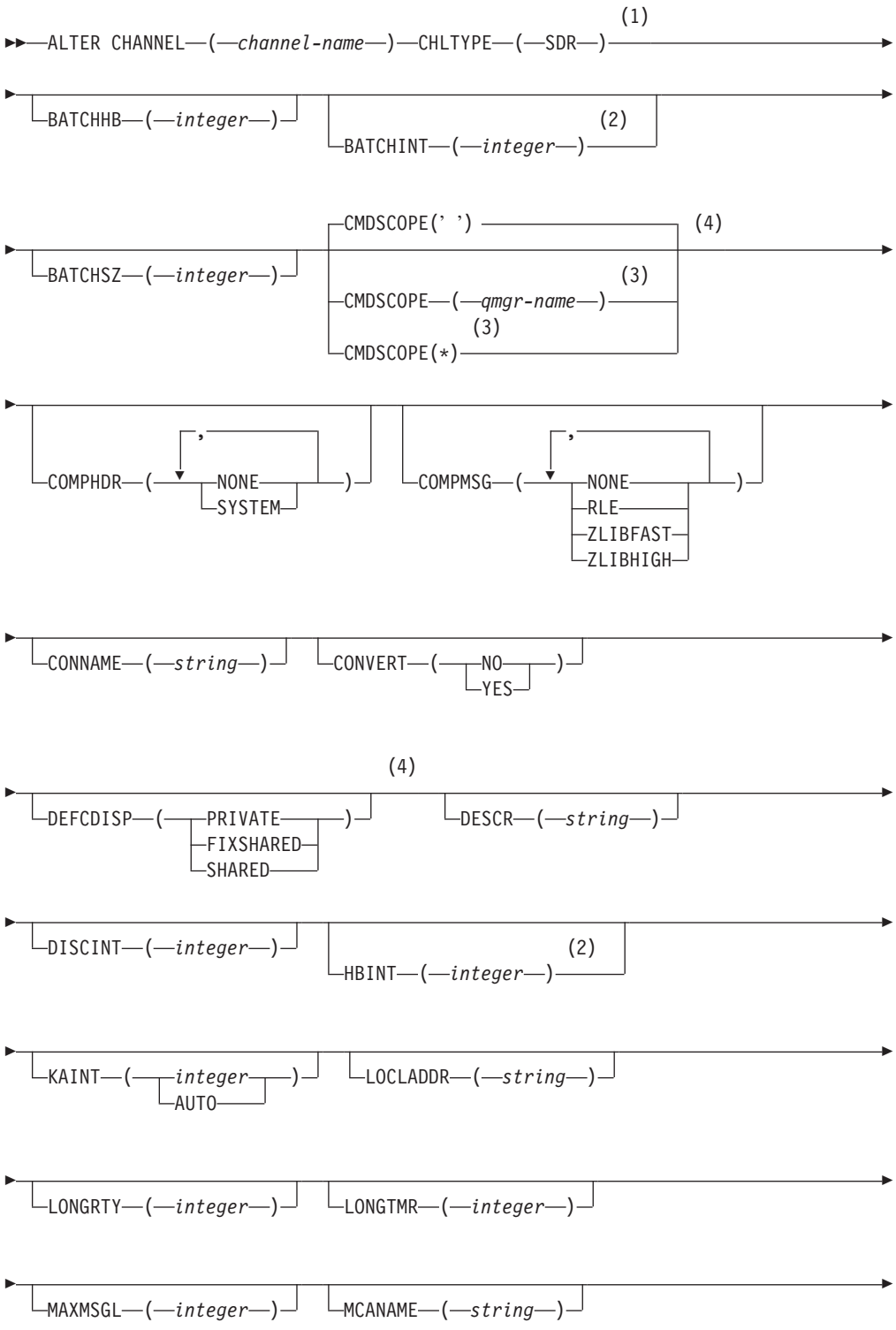
Synonym: ALT CHL

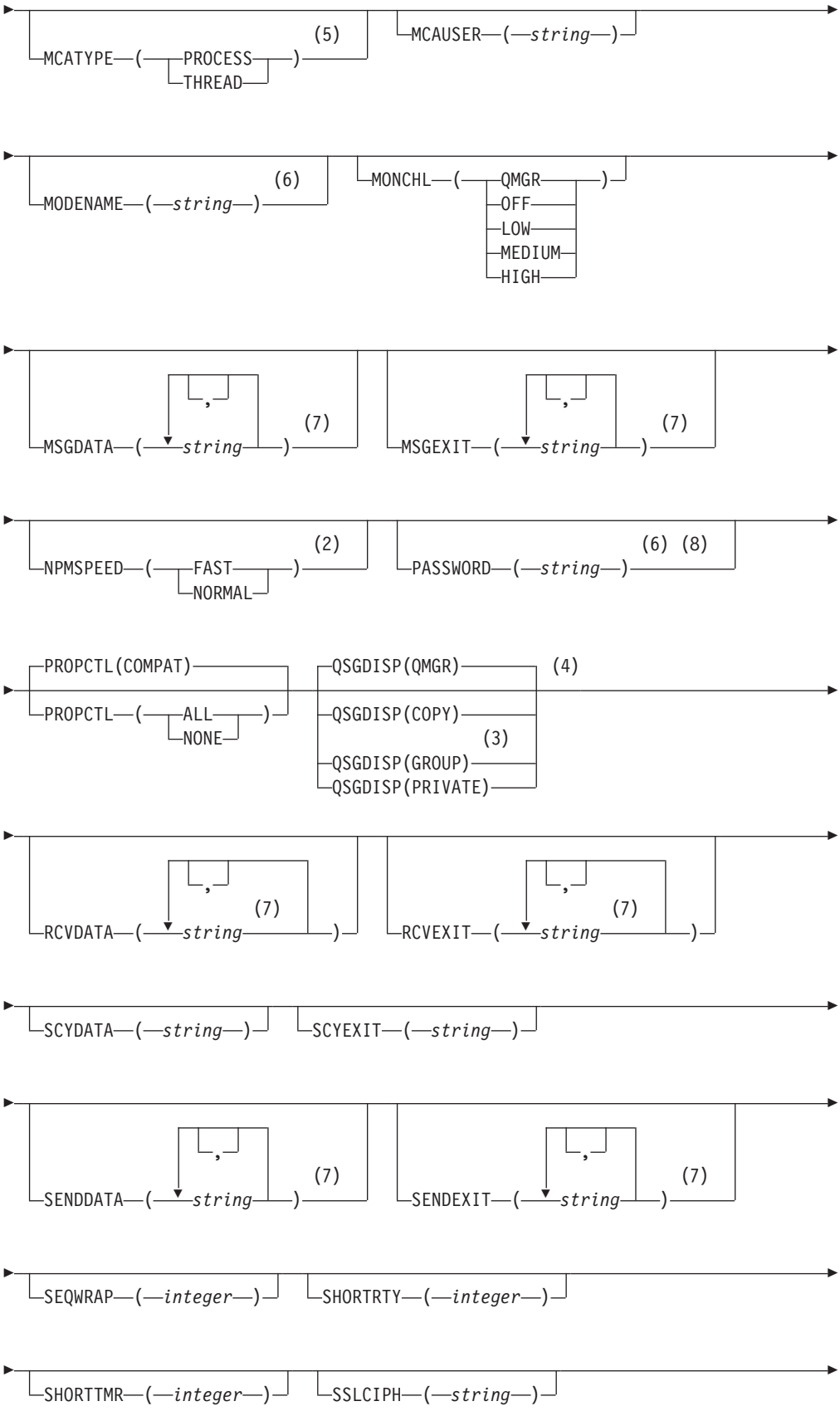
There is a separate syntax diagram for each type of channel:

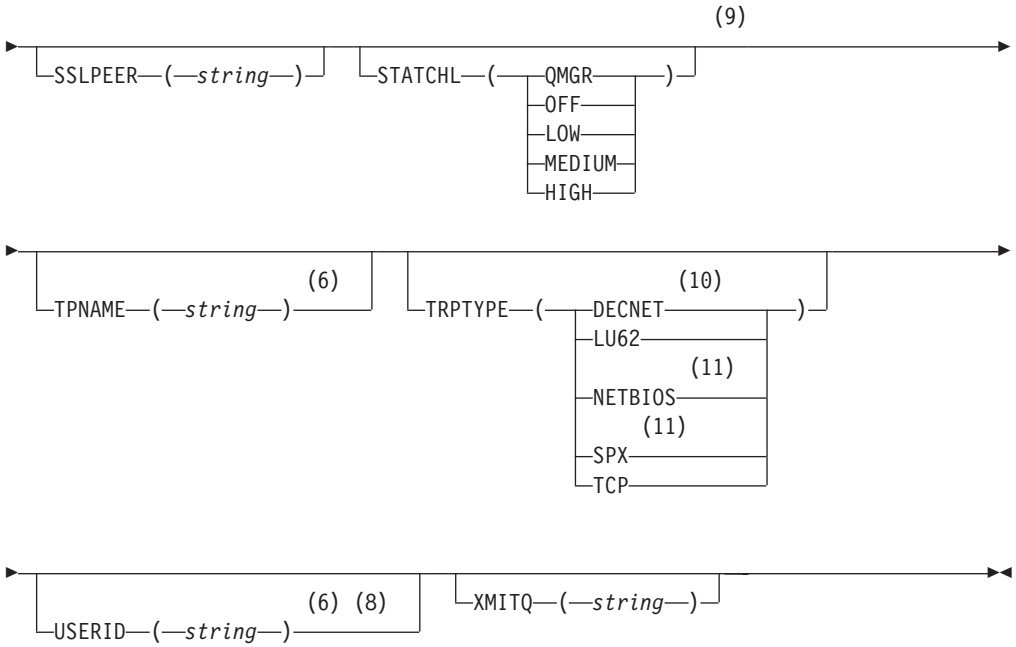
- "Sender channel" on page 16
- "Server channel" on page 19
- "Receiver channel" on page 22
- "Requester channel" on page 25
- "Client-connection channel" on page 28
- "Server-connection channel" on page 30
- "Cluster-sender channel" on page 32
- "Cluster-receiver channel" on page 35

Sender channel

ALTER CHANNEL




MCATYPE(PROCESS | THREAD) (5)
MCAUSER(string)
MODENAME(string) (6)
MONCHL(QMGR | OFF | LOW | MEDIUM | HIGH)
MSGDATA(string) (7)
MSGEXIT(string) (7)
NPMSPEED(FAST | NORMAL) (2)
PASSWORD(string) (6) (8)
PROPCTL(COMPAT)
PROPCTL(ALL | NONE)
QSGDISP(QMGR) (4)
QSGDISP(COPY)
QSGDISP(GROUP) (3)
QSGDISP(PRIVATE)
RCVDATA(string) (7)
RCVEXIT(string) (7)
SCYDATA(string)
SCYEXIT(string)
SENDDATA(string) (7)
SENDEXIT(string) (7)
SEQWRAP(integer)
SHORTRTY(integer)
SHORTTMR(integer)
SSLCIPH(string)



Notes:

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

2 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows and z/OS.

3 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

4 Valid only on z/OS.

5 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

6 Valid only if TRPTYPE is LU62.

7 You can specify more than one value on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS only.

8 Not valid on z/OS.

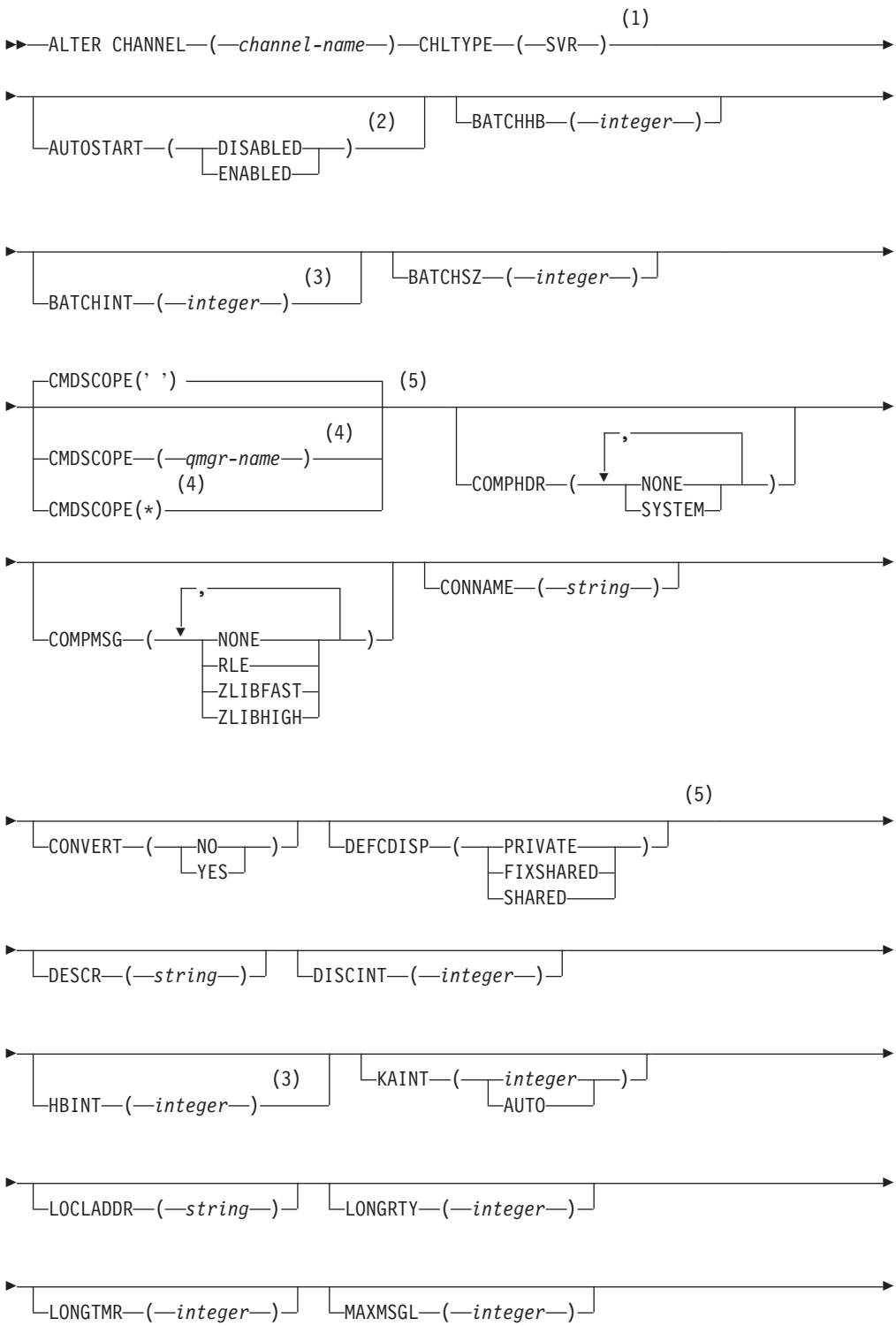
9 This parameter is valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

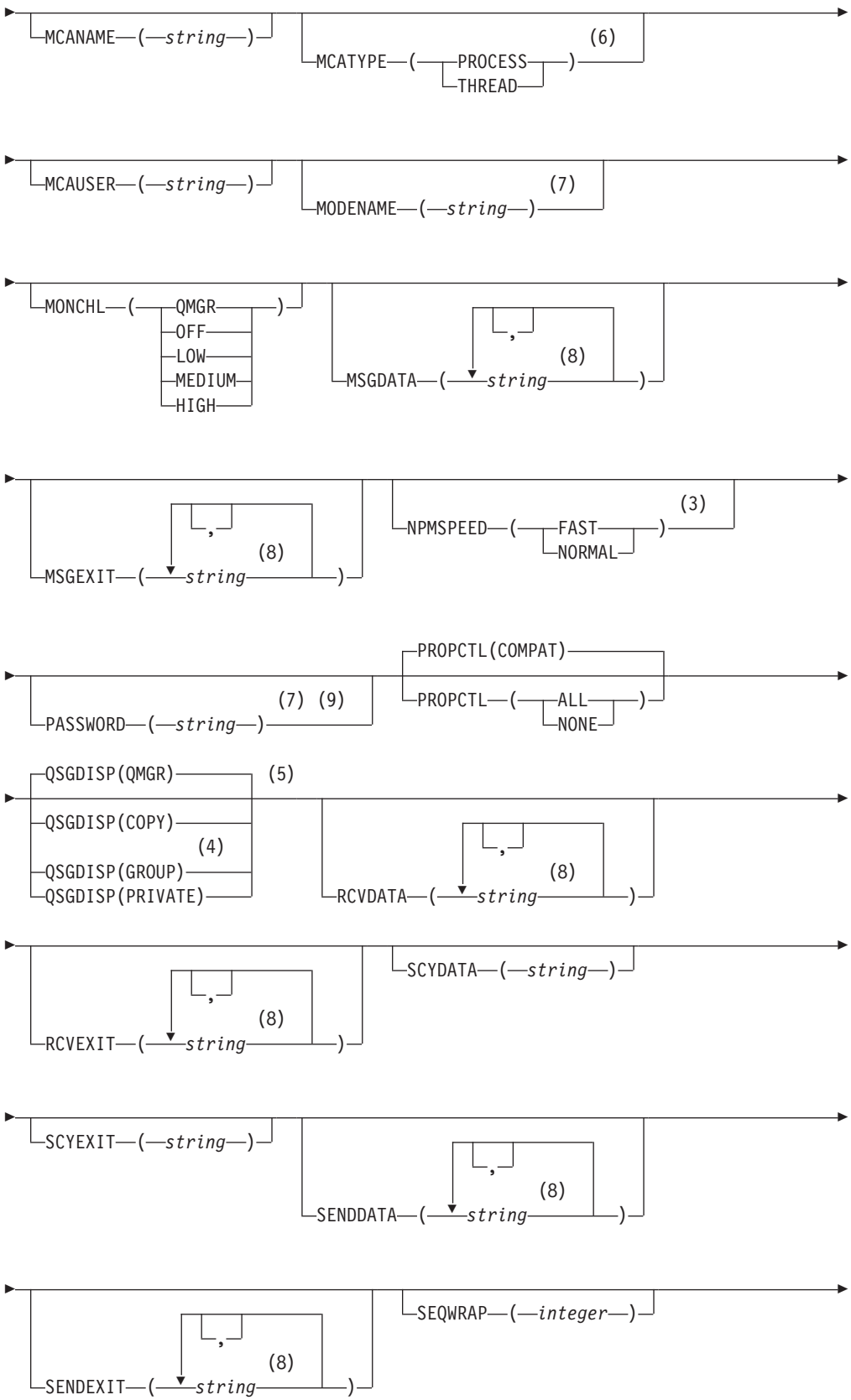
10 Valid only on HP OpenVMS.

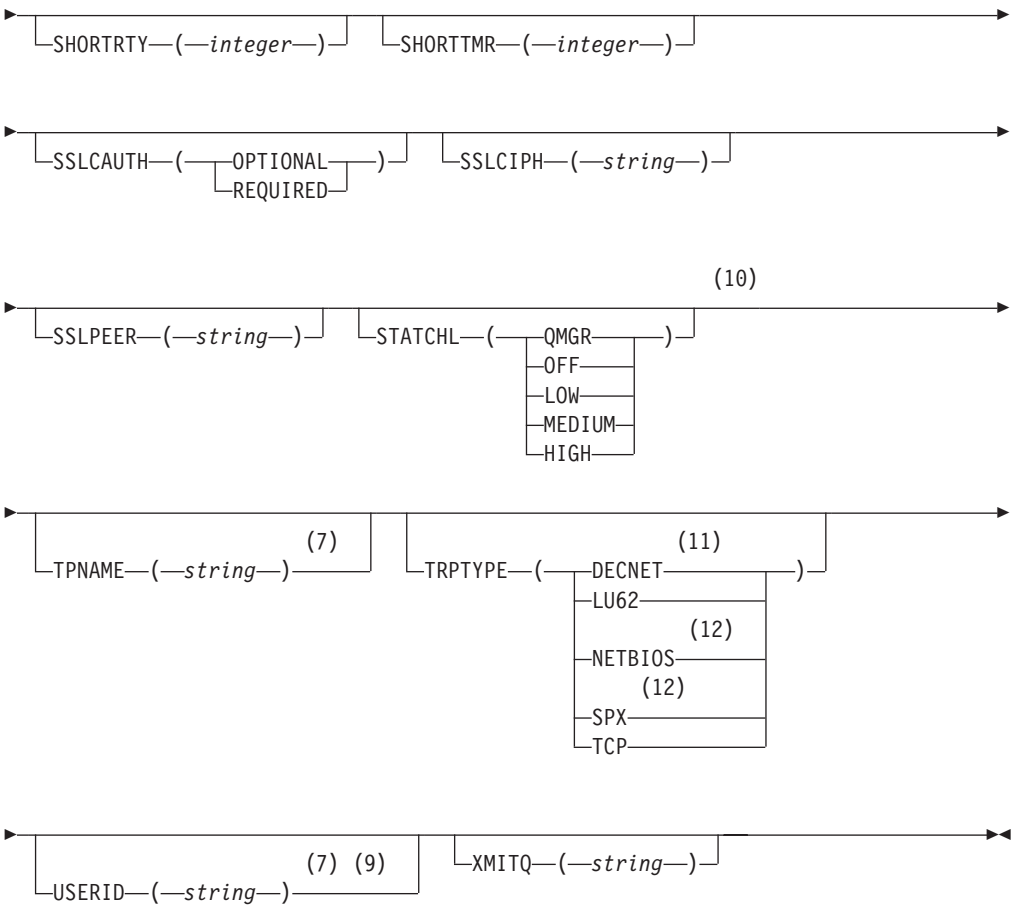
11 Valid only Windows.

Server channel

ALTER CHANNEL




MCANAME(string)
MCATYPE(PROCESS | THREAD) (6)
MCAUSER(string)
MODENAME(string) (7)
MONCHL(QMGR | OFF | LOW | MEDIUM | HIGH)
MSGDATA(string) (8)
MSGEXIT(string) (8)
NPMSPEED(FAST | NORMAL) (3)
PASSWORD(string) (7) (9)
PROPCTL(COMPAT)
PROPCTL(ALL | NONE)
QSGDISP(QMGR) (5)
QSGDISP(COPY)
QSGDISP(GROUP) (4)
QSGDISP(PRIVATE)
RCVDATA(string) (8)
RCVEXIT(string) (8)
SCYDATA(string)
SCYEXIT(string)
SENDDATA(string) (8)
SENDEXIT(string) (8)
SEQWRAP(integer)



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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows and z/OS.

4 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

5 Valid only on z/OS.

6 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

7 Valid only if TRPTYPE is LU62.

8 You can specify more than one value on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS only.

9 Not valid on z/OS.

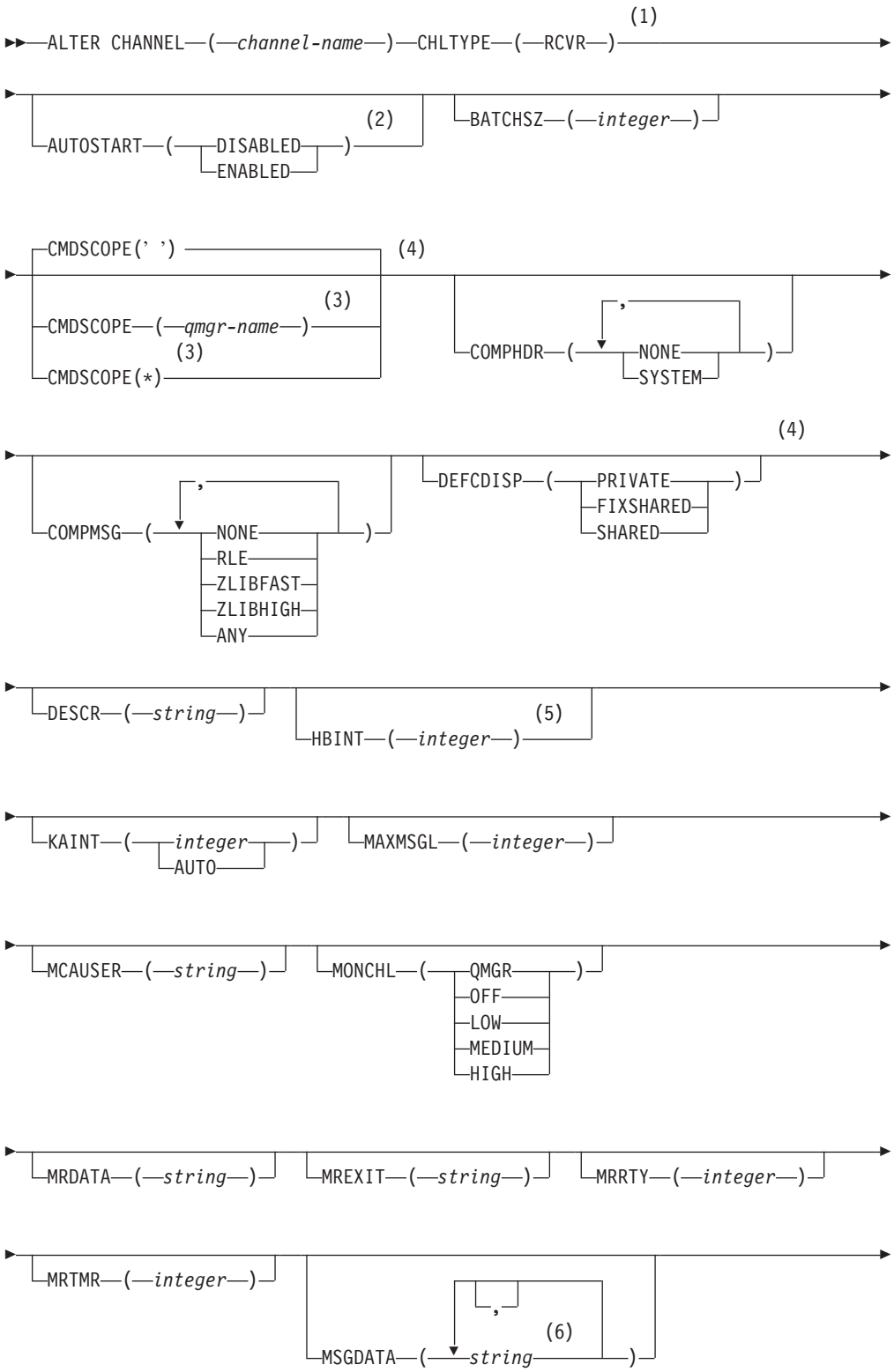
10 This parameter is valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

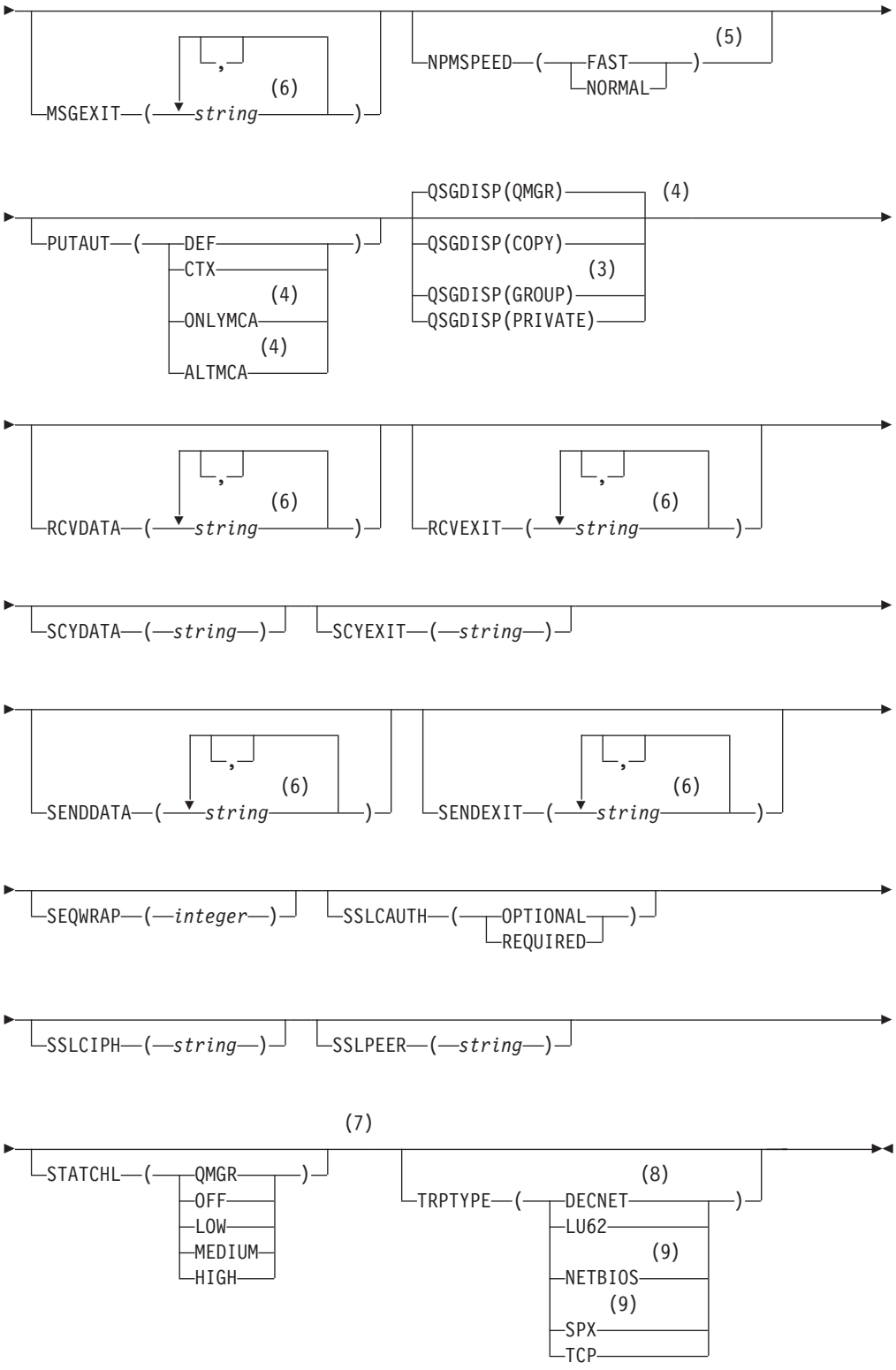
11 Valid only on HP OpenVMS.

12 Valid only on Windows.

Receiver channel

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 on z/OS when the queue manager is a member of a queue-sharing group.

4 Valid only on z/OS.

5 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

6 You can specify more than one value on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS only.

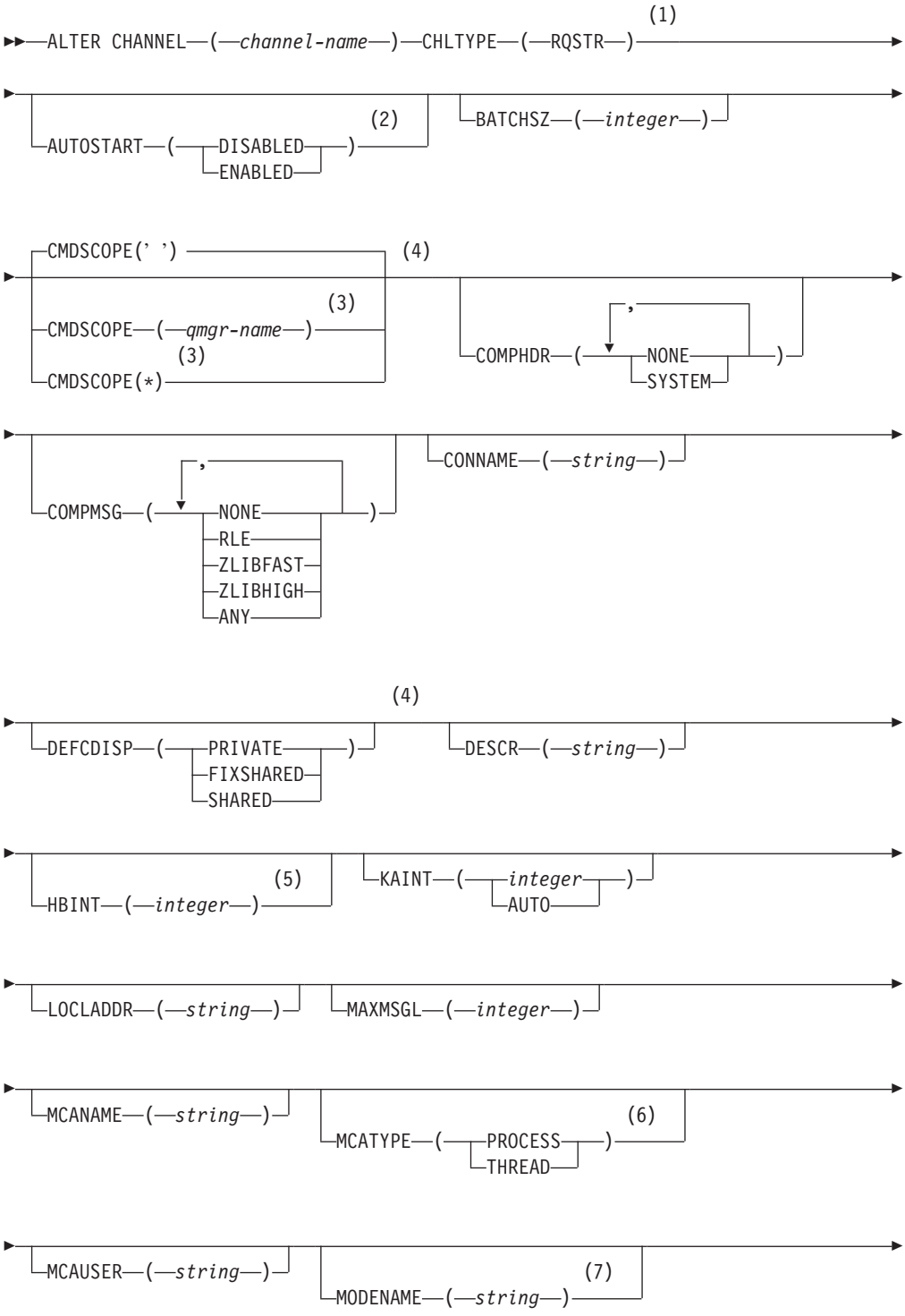
7 This parameter is valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

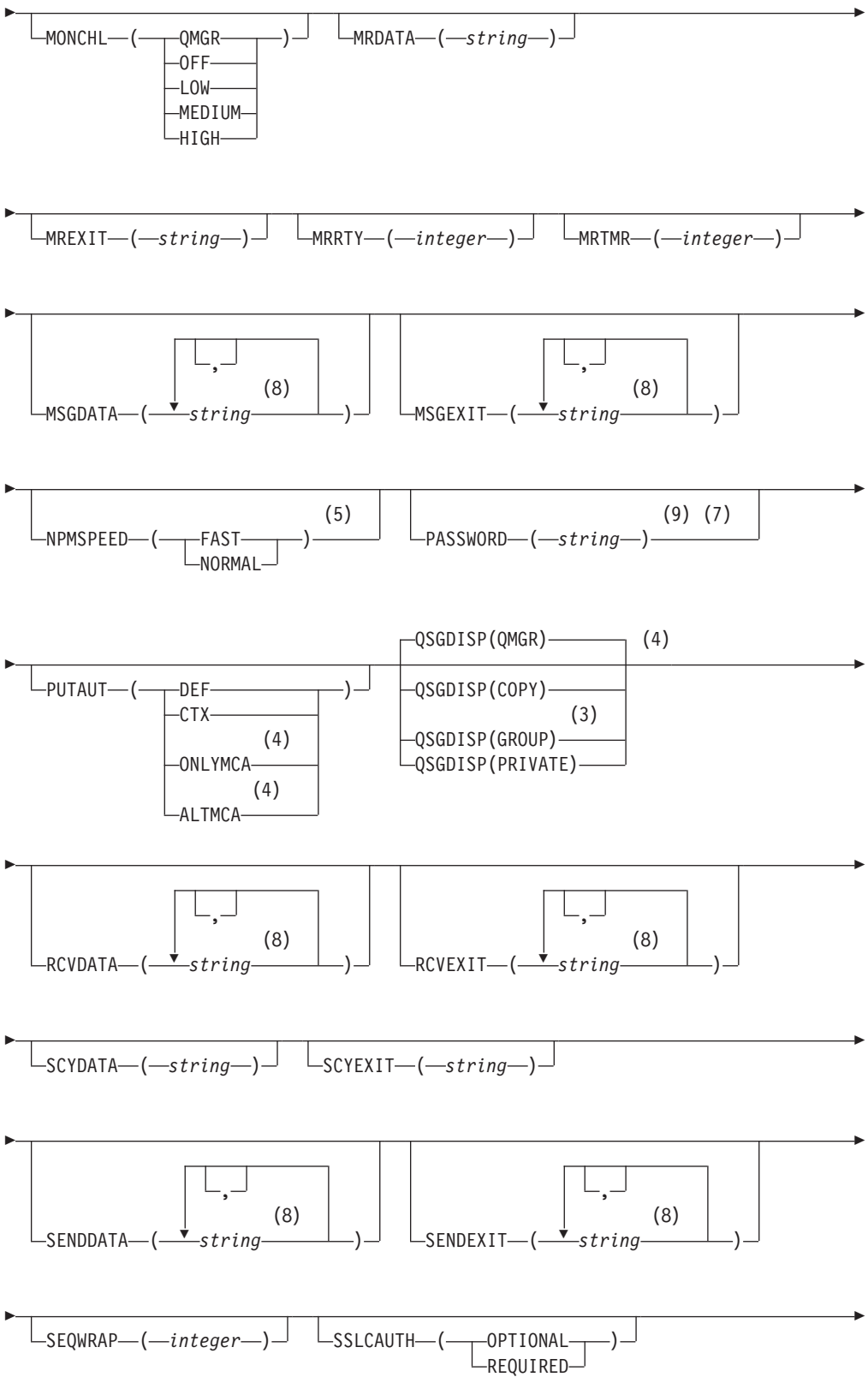
8 Valid only on HP OpenVMS.

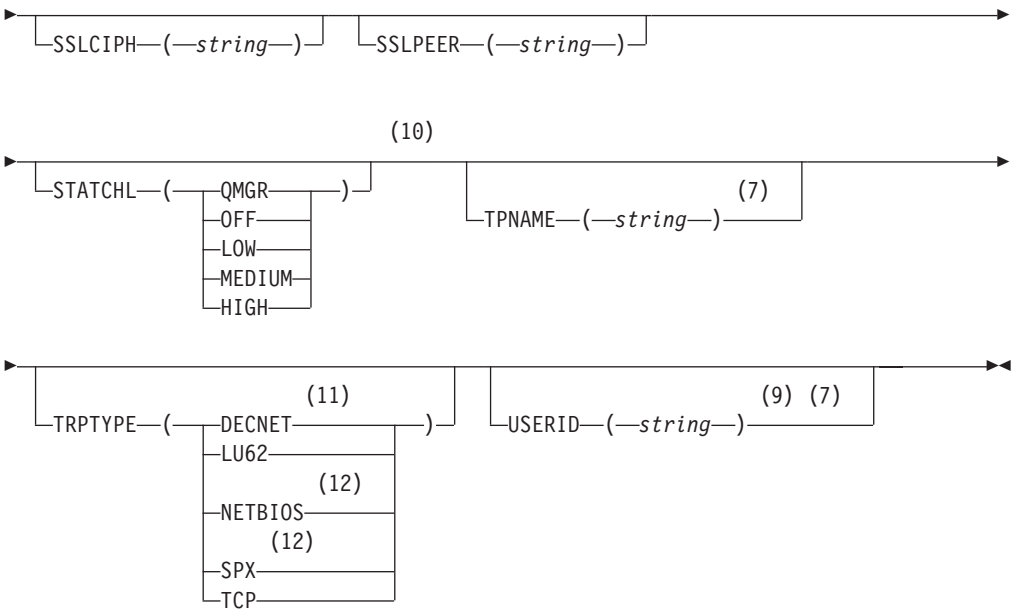
9 Valid only on Windows.

Requester channel

ALTER CHANNEL




MONCHL
QMGR
OFF
LOW
MEDIUM
HIGH
MRDATA
string
MREXIT
string
MRRTY
integer
MRTMR
integer
MSGDATA
string
(8)
MSGEXIT
string
(8)
NPMSPEED
FAST
NORMAL
(5)
PASSWORD
string
(9) (7)
PUTAUT
DEF
CTX
ONLYMCA
(4)
ALTMCA
(4)
QSGDISP(QMGR)
QSGDISP(COPY)
QSGDISP(GROUP)
(3)
QSGDISP(PRIVATE)
(4)
RCVDATA
string
(8)
RCVEXIT
string
(8)
SCYDATA
string
SCYEXIT
string
SENDDATA
string
(8)
SENDEXIT
string
(8)
SEQWRAP
integer
SSLCAUTH
OPTIONAL
REQUIRED



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 z/OS when the queue manager is a member of a queue-sharing group.

4 Valid only on z/OS.

5 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

6 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

7 Valid only if TRPTYPE is LU62.

8 You can specify more than one value on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS only.

9 Not valid on z/OS.

10 This parameter is valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

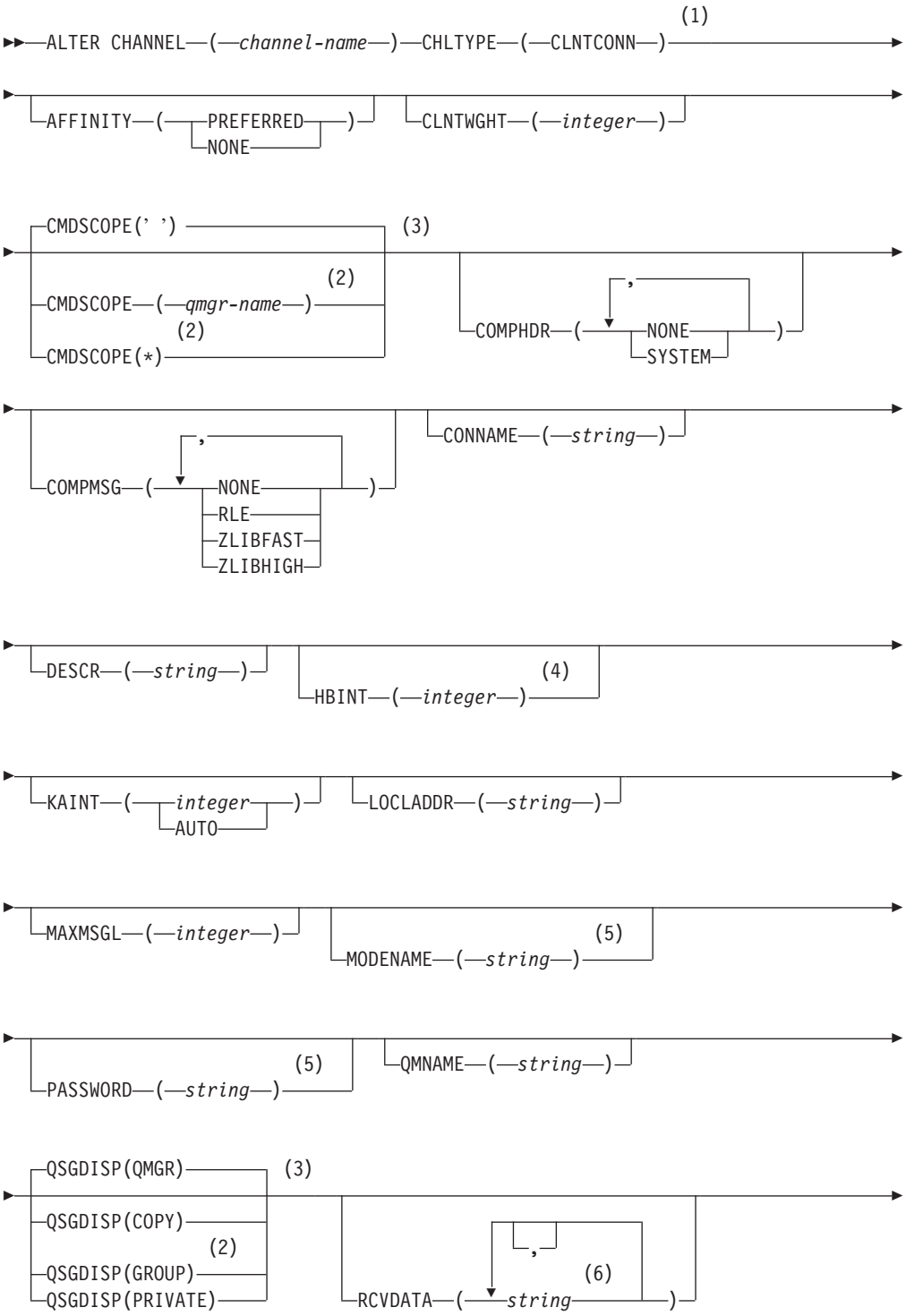
11 Valid only on HP OpenVMS.

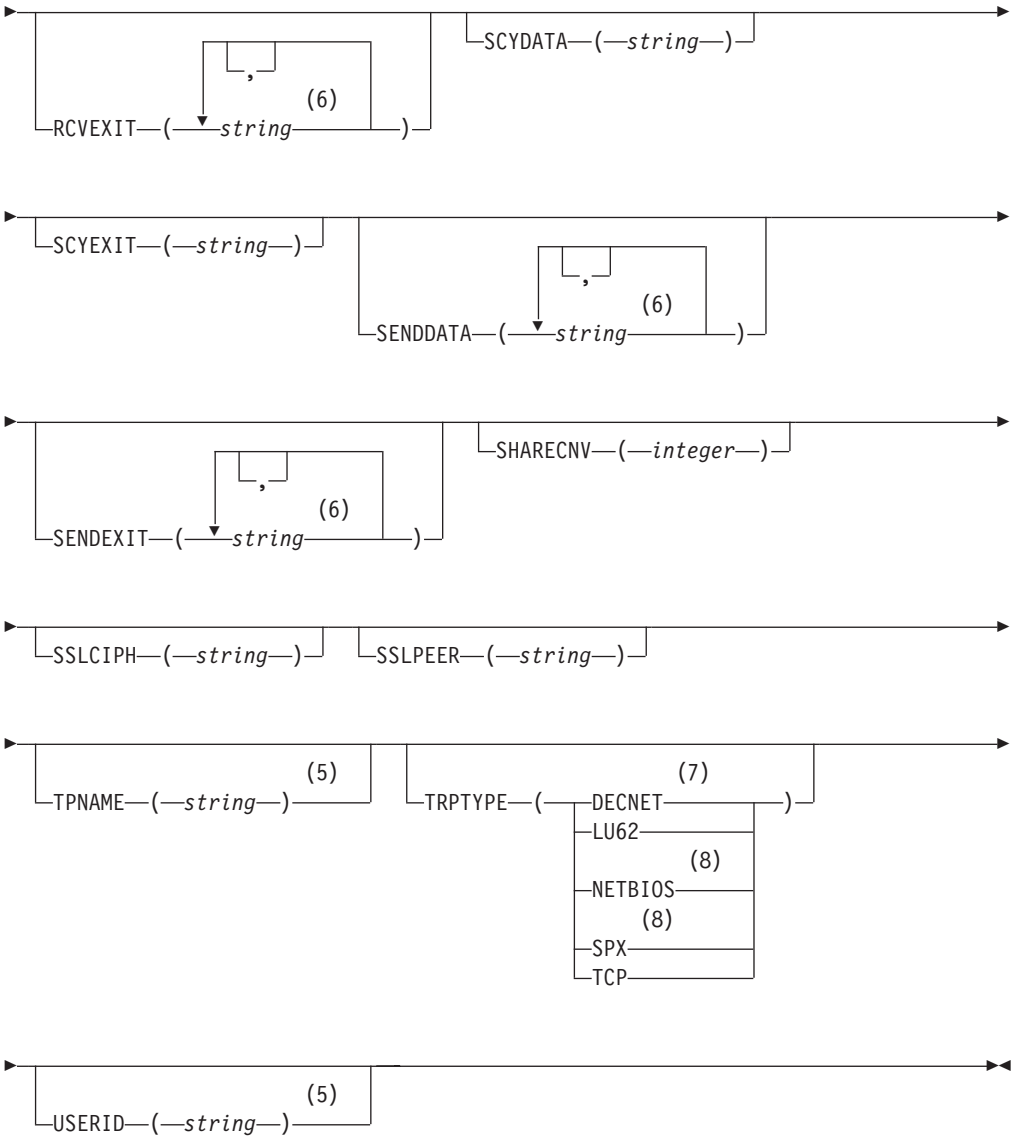
12 Valid only on Windows.

Client-connection channel

Syntax diagram for a client-connection channel when using the ALTER CHANNEL command.

ALTER CHANNEL





Notes:

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

2 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

3 Valid only on z/OS.

4 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

5 Valid only if TRPTYPE is LU62.

6 You can specify more than one value on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS only.

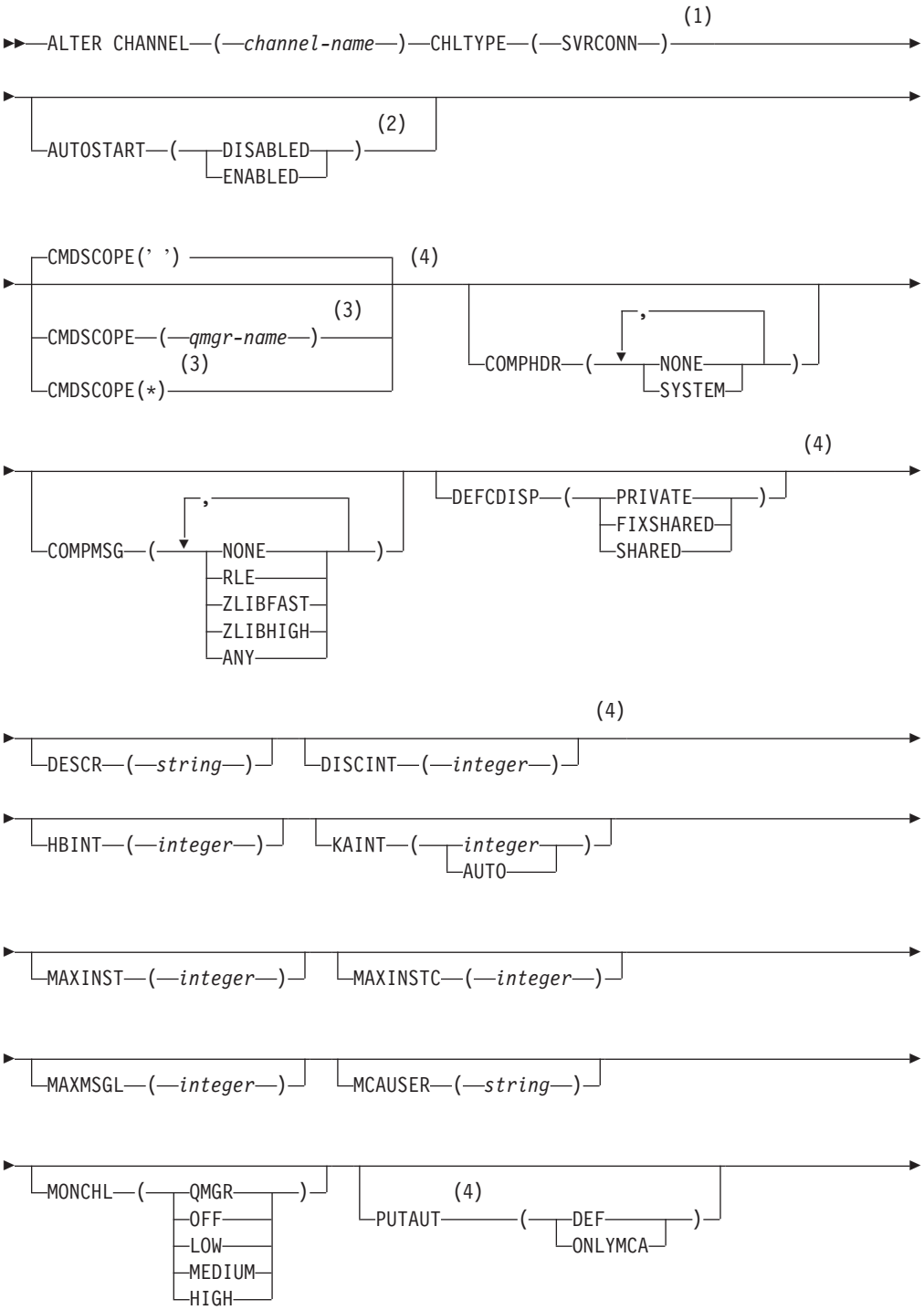
7 Valid only on HP OpenVMS.

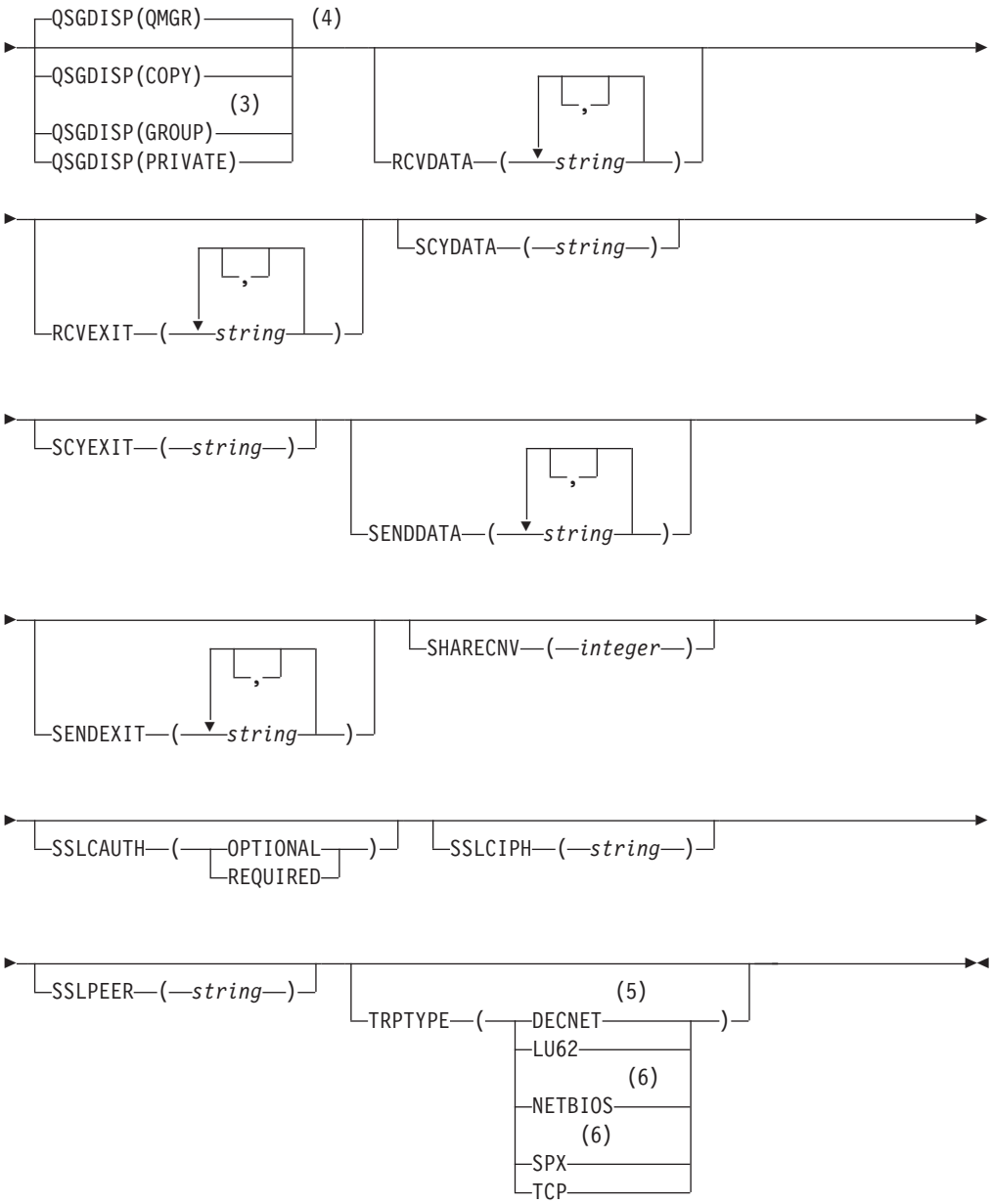
8 Valid only for clients to be run on DOS and Windows.

Server-connection channel

Syntax diagram for a server-connection channel when using the ALTER CHANNEL command.

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 on z/OS when the queue manager is a member of a queue-sharing group.

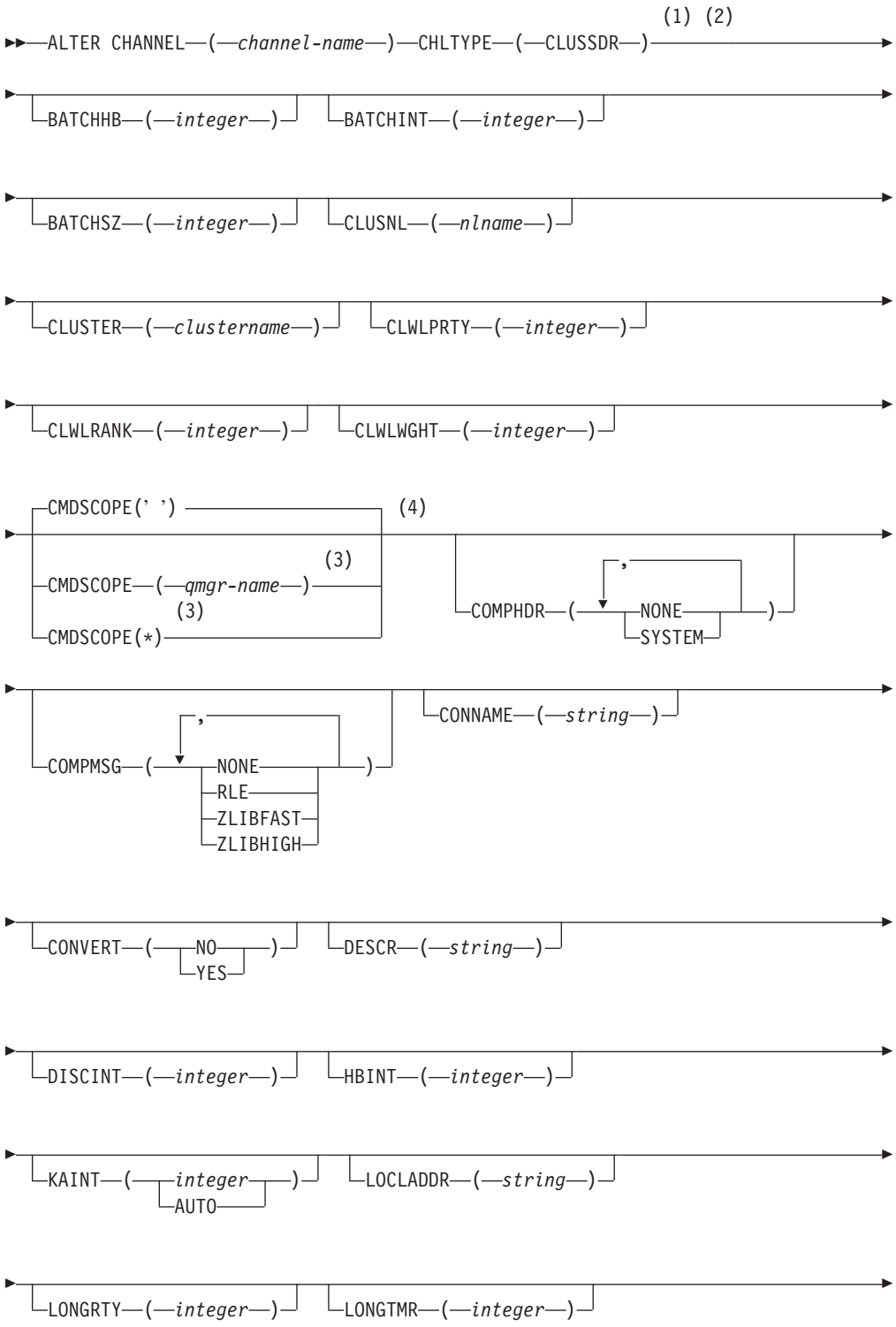
4 Valid only on z/OS.

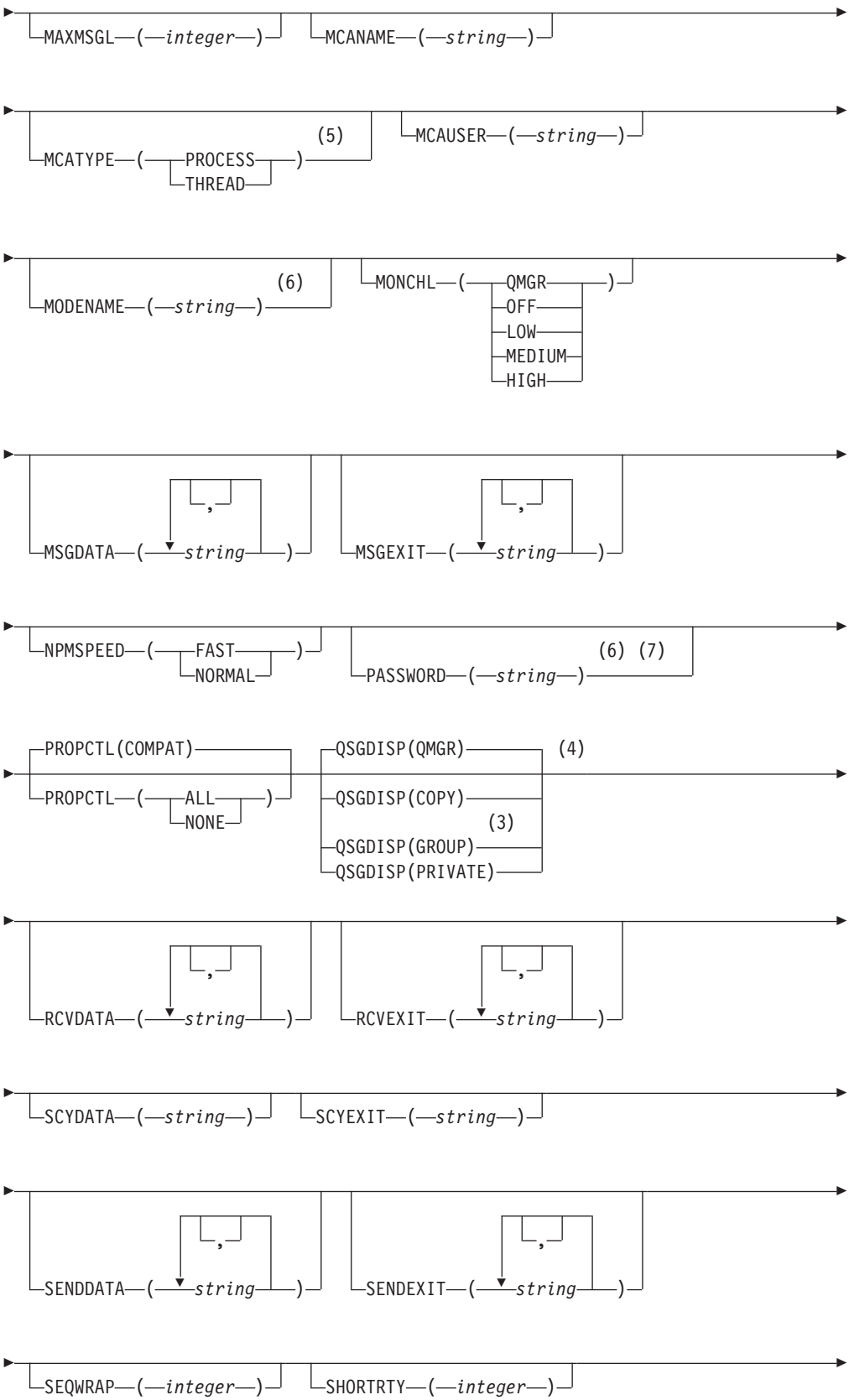
5 Valid only on HP OpenVMS.

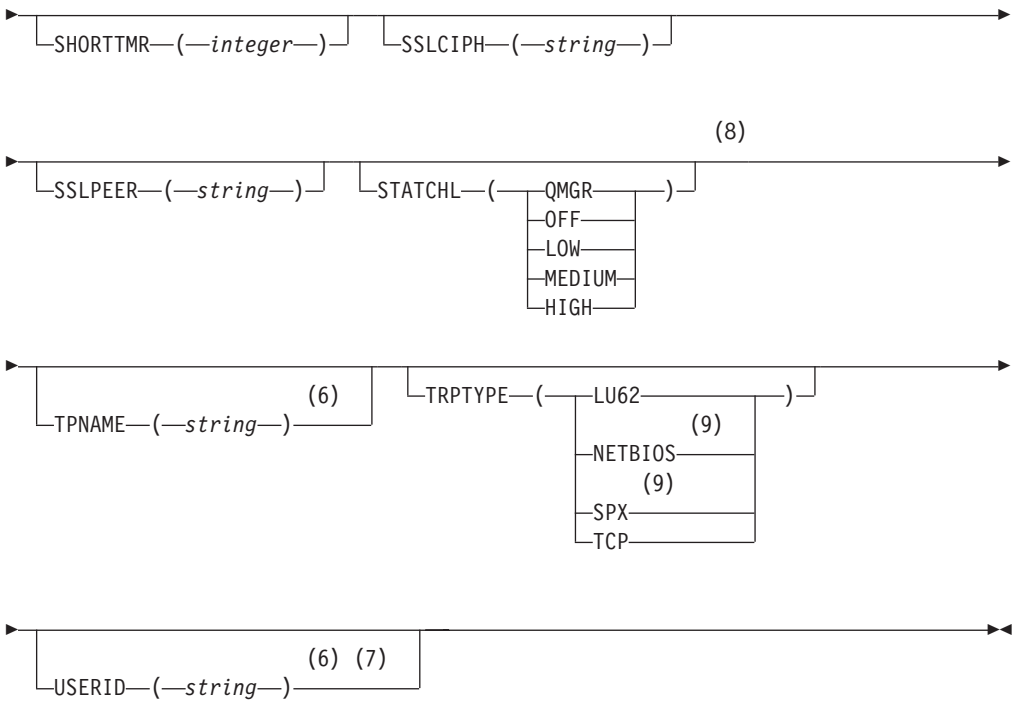
6 Valid only for clients to be run on Windows.

Cluster-sender channel

ALTER CHANNEL




MAXMSGL(integer)
MCANAME(string)
MCATYPE(PROCESS | THREAD) (5)
MCAUSER(string)
MODENAME(string) (6)
MONCHL(QMGR | OFF | LOW | MEDIUM | HIGH)
MSGDATA(string,...)
MSGEXIT(string,...)
NPMSPEED(FAST | NORMAL)
PASSWORD(string) (6) (7)
PROPCTL(COMPAT)
PROPCTL(ALL | NONE)
QSGDISP(QMGR) (4)
QSGDISP(COPY)
QSGDISP(GROUP) (3)
QSGDISP(PRIVATE)
RCVDATA(string,...)
RCVEXIT(string,...)
SCYDATA(string)
SCYEXIT(string)
SENDDATA(string,...)
SENDEXIT(string,...)
SEQWRAP(integer)
SHORTRTY(integer)



Notes:

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

2 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

3 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

4 Valid only on z/OS.

5 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

6 Valid only if TRPTYPE is LU62.

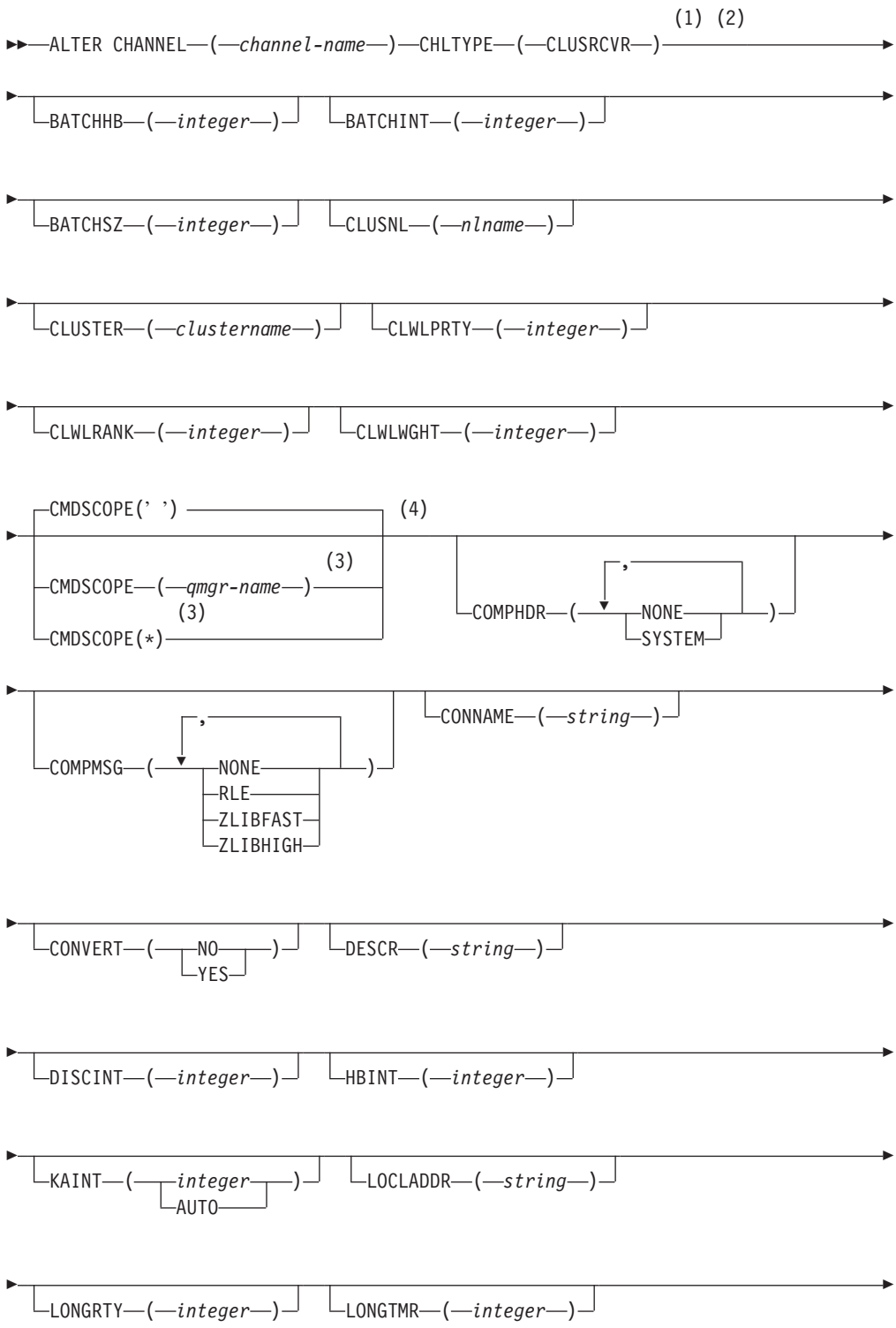
7 Not valid on z/OS.

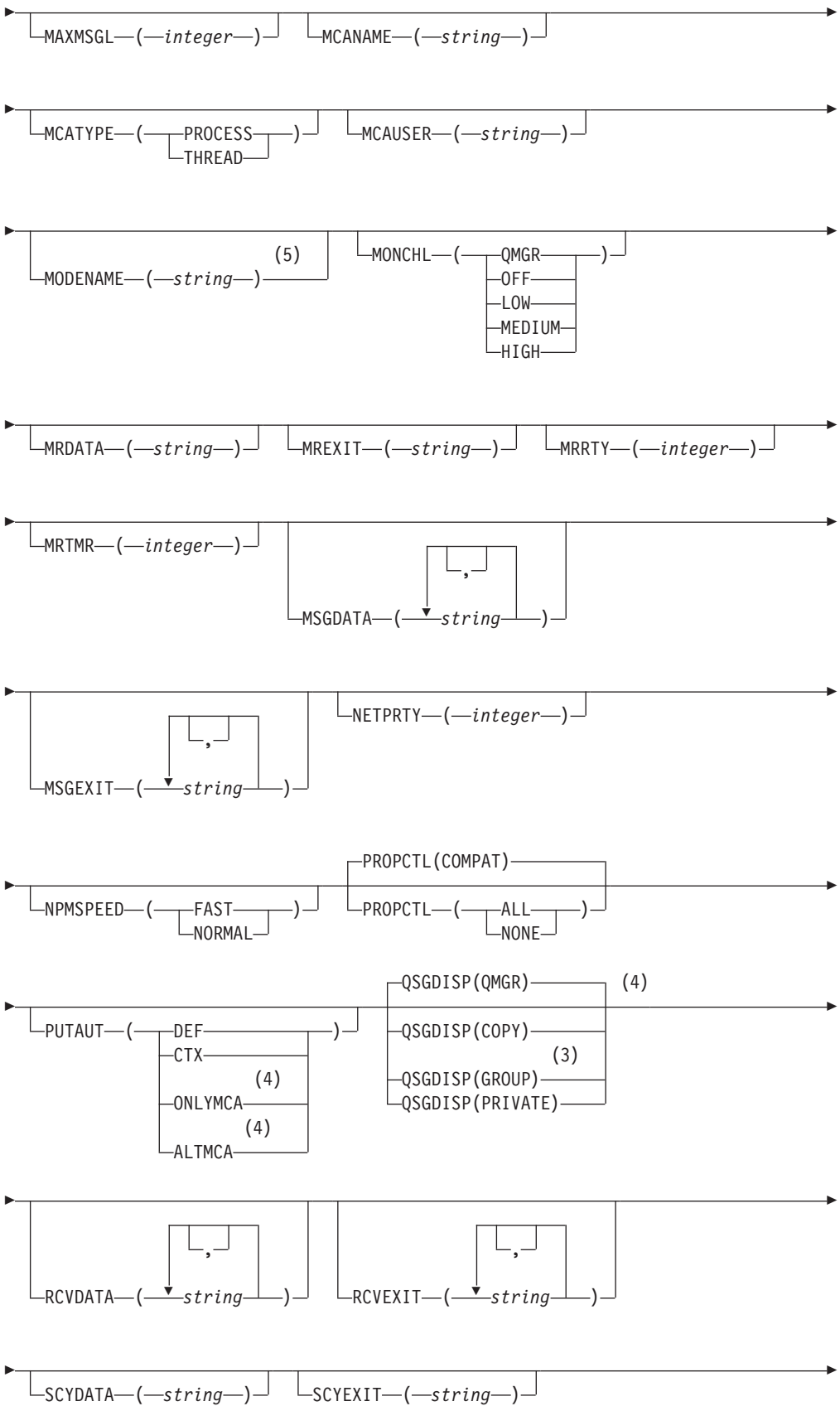
8 This parameter is valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

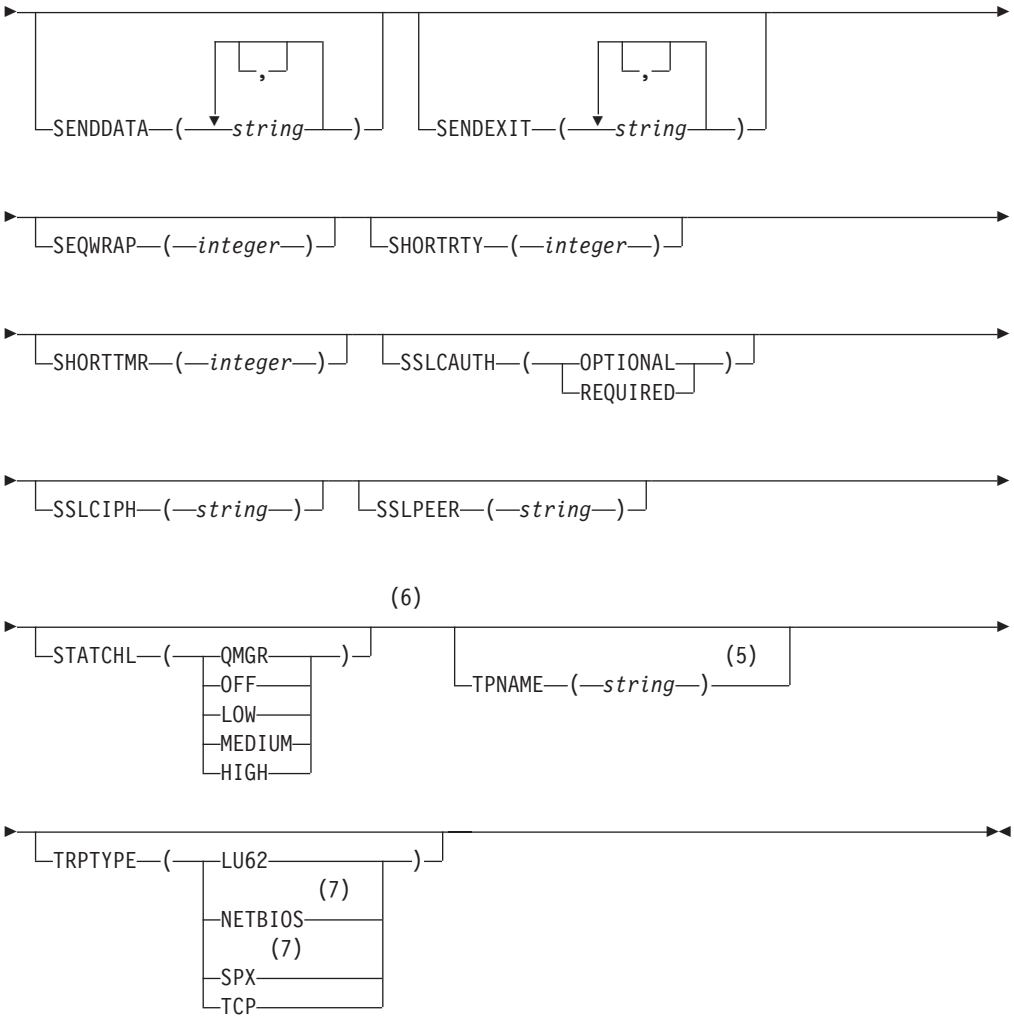
9 Valid only Windows.

Cluster-receiver channel

ALTER CHANNEL




MAXMSGL—(—integer—)
MCANAME—(—string—)
MCATYPE—(—PROCESS—)
THREAD
MCAUSER—(—string—)
MODENAME—(—string—)
(5)
MONCHL—(—QMGR—)
OFF
LOW
MEDIUM
HIGH
MRDATA—(—string—)
MREXIT—(—string—)
MRRTY—(—integer—)
MRTMR—(—integer—)
,
MSGDATA—(—string—)
,
MSGEXIT—(—string—)
NETPRTY—(—integer—)
PROPCTL(COMPAT)
NPMSPEED—(—FAST—)
NORMAL
PROPCTL—(—ALL—)
NONE
QSGDISP(QMGR)
(4)
PUTAUT—(—DEF—)
CTX
QSGDISP(COPY)
(4)
ONLYMCA
(3)
QSGDISP(GROUP)
(4)
ALTMCA
QSGDISP(PRIVATE)
,
RCVDATA—(—string—)
,
RCVEXIT—(—string—)
SCYDATA—(—string—)
SCYEXIT—(—string—)



Notes:

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

2 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

3 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

4 Valid only on z/OS.

5 Valid only if TRPTYPE is LU62.

6 This parameter is valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

7 Valid only on Windows.

Parameter descriptions for ALTER CHANNEL

For a description of the parameters see “DEFINE CHANNEL” on page 144.

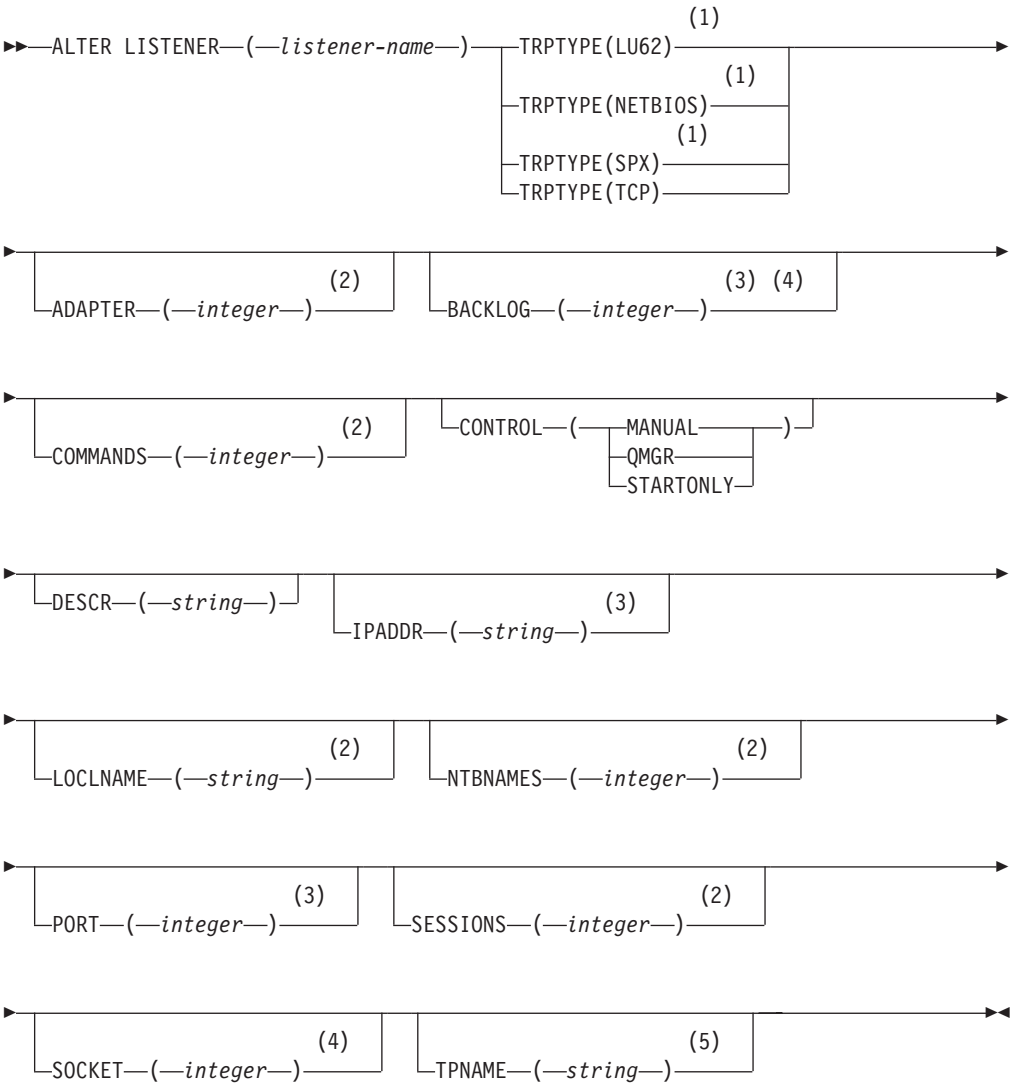
ALTER LISTENER

Use MQSC command ALTER LISTENER to alter the parameters of an existing WebSphere MQ listener definition. If the listener is already running, any changes you make to its definition are effective only after the next time that the listener is started.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	

Synonym: ALT LSTR

ALTER LISTENER



Notes:

1 Valid only on Windows.

2 Valid only on Windows when TRPTYPE is NETBIOS.

3 Valid when TRPTYPE is TCP.

4 Valid on Windows when TRPTYPE is SPX.

5 Valid only on Windows when TRPTYPE is LU62.

Parameter descriptions for ALTER LISTENER

For a description of the parameters see “DEFINE LISTENER” on page 208.

ALTER NAMELIST

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

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

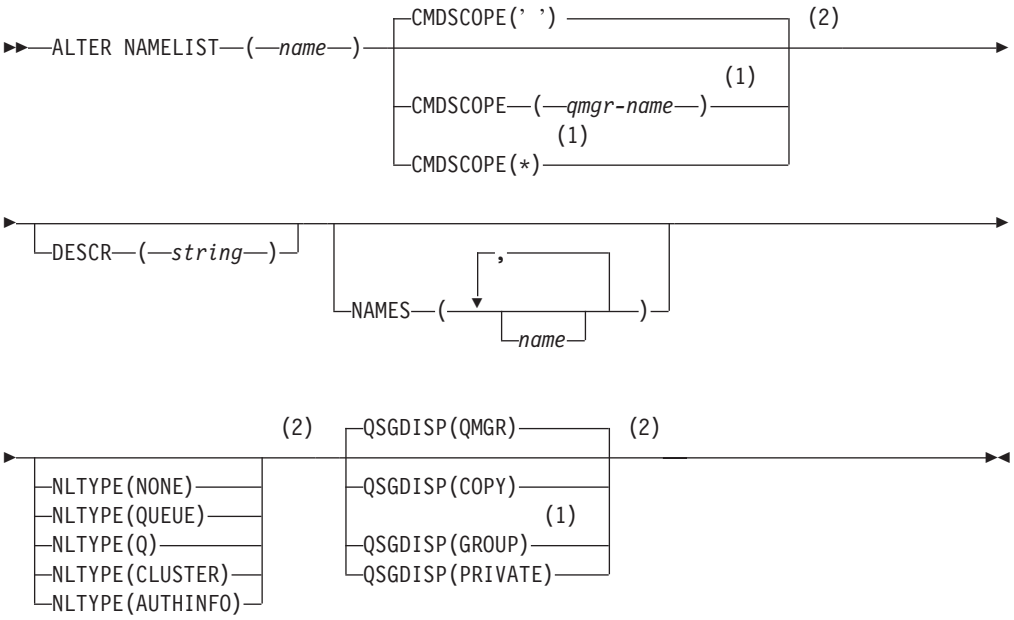
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

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

Synonym: ALT NL

ALTER NAMELIST



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for ALTER NAMELIST

For a description of the parameters see "DEFINE NAMELIST" on page 216

ALTER PROCESS

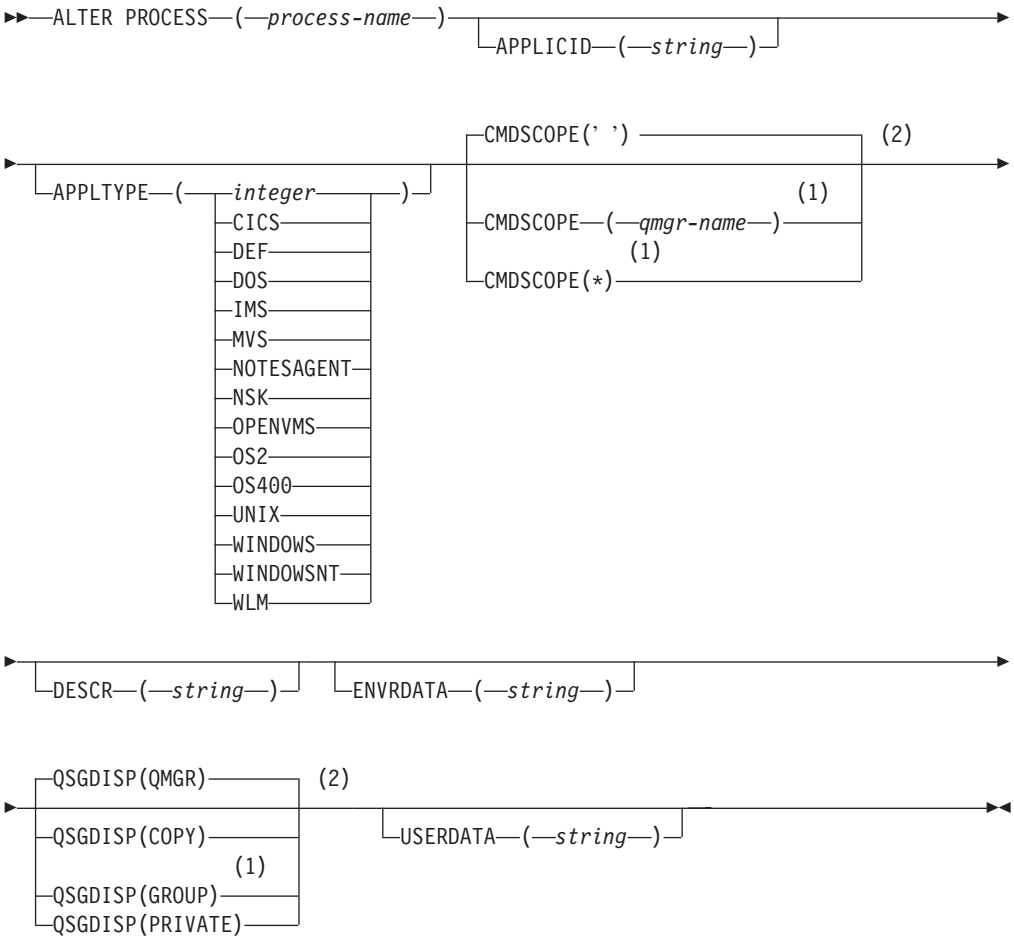
Use the MQSC command ALTER PROCESS to alter the parameters of an existing WebSphere MQ process definition.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: ALT PRO

ALTER PROCESS



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for ALTER PROCESS

For a description of the parameters see “DEFINE PROCESS” on page 221.

ALTER PSID

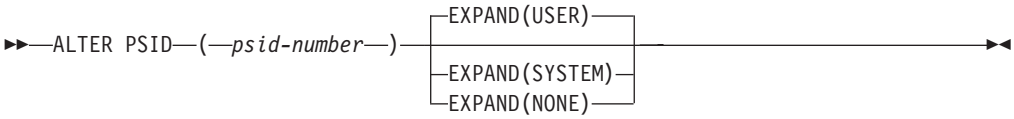
Use the MQSC command ALTER PSID to change the expansion method for a page set.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: ALT PSID

ALTER PSID



Parameter descriptions for ALTER PSID

(psid-number)
Identifier of the page set. This is required.

EXPAND
Controls how the queue manager should expand a page set when it becomes nearly full, and further pages are required in it.

USER The secondary extent size that was specified when the page set was defined is used. If no secondary extent size was specified, or if it was specified as zero, then no dynamic page set expansion can take place.

At restart, if a previously used page set has been replaced with a data set that is smaller, it is expanded until it reaches the size of the previously used data set. Only one extent is required to reach this size.

SYSTEM

A secondary extent size that is approximately 10 per cent of the current size of the page set is used. It may be rounded up depending on the characteristics of the DASD.

The secondary extent size that was specified when the page set was defined is ignored; dynamic expansion can occur if it was zero or not specified.

NONE
No further page set expansion is to take place.

ALTER QMGR

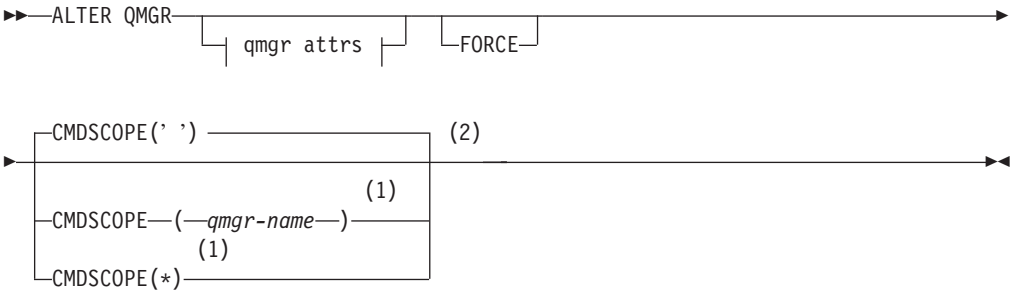
Use MQSC command ALTER QMGR to alter the queue manager parameters for the local queue manager.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

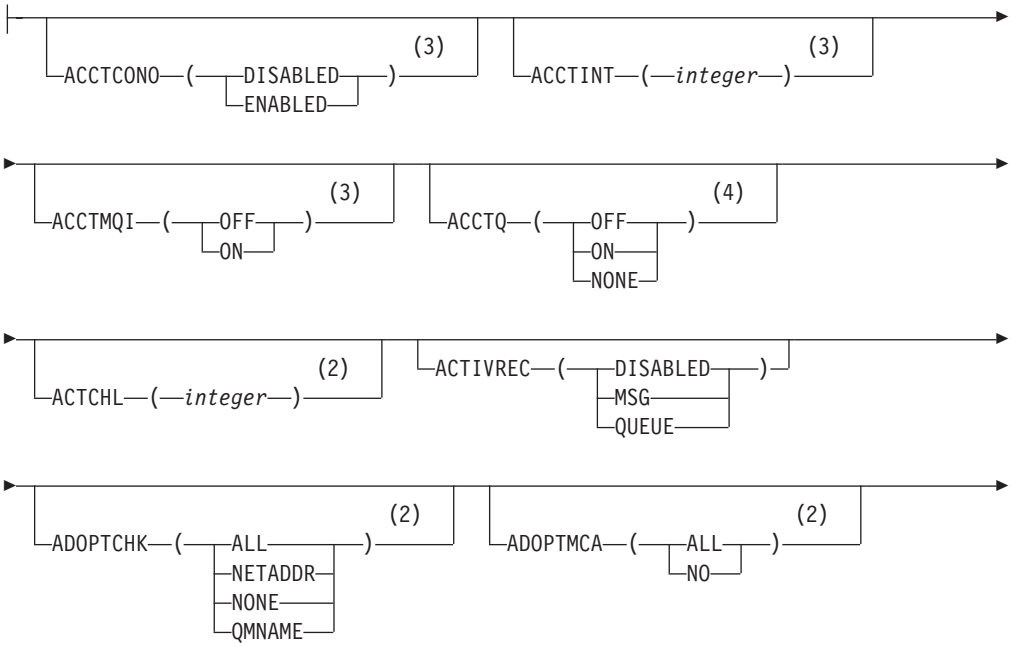
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

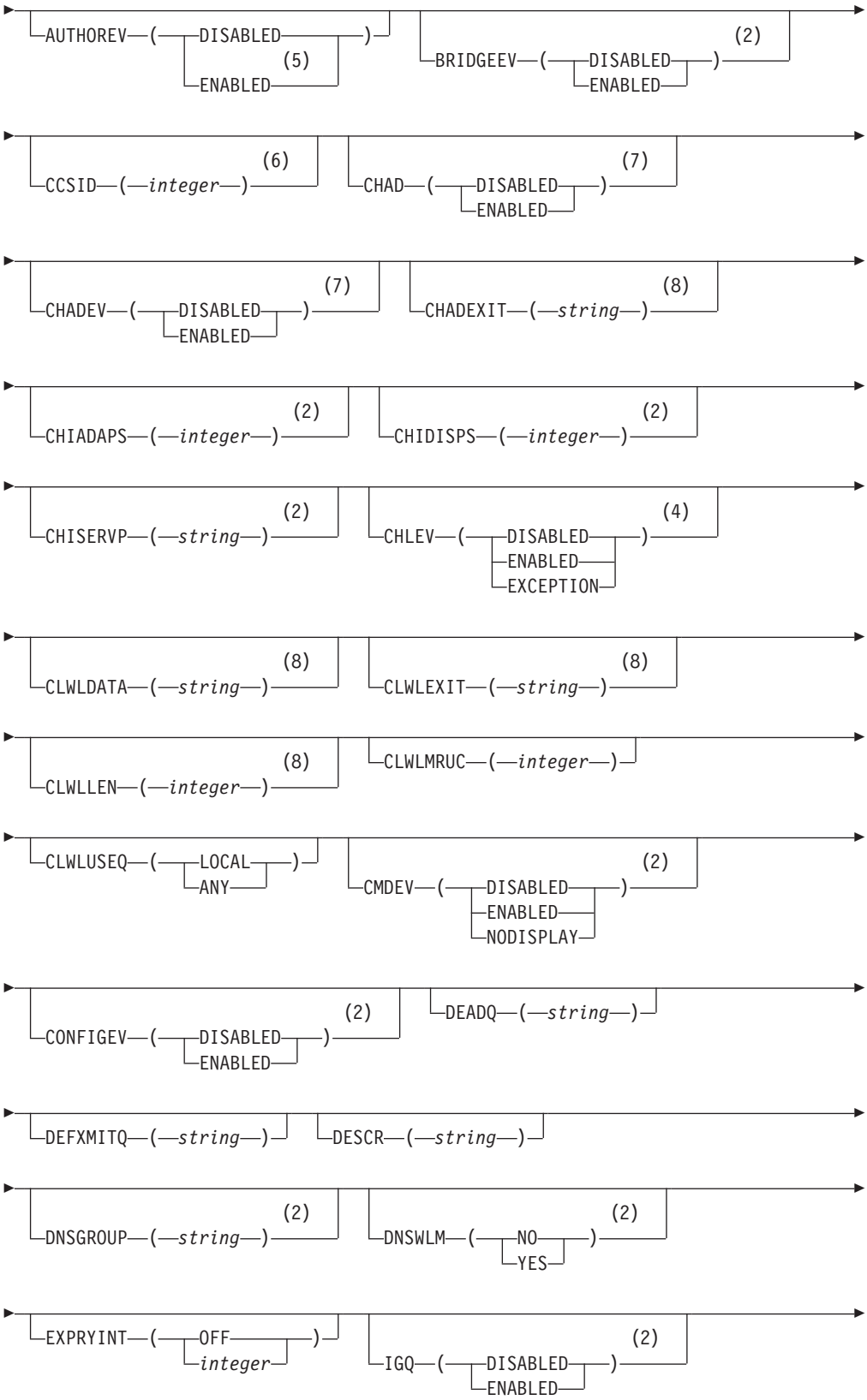
Synonym: ALT QMGR

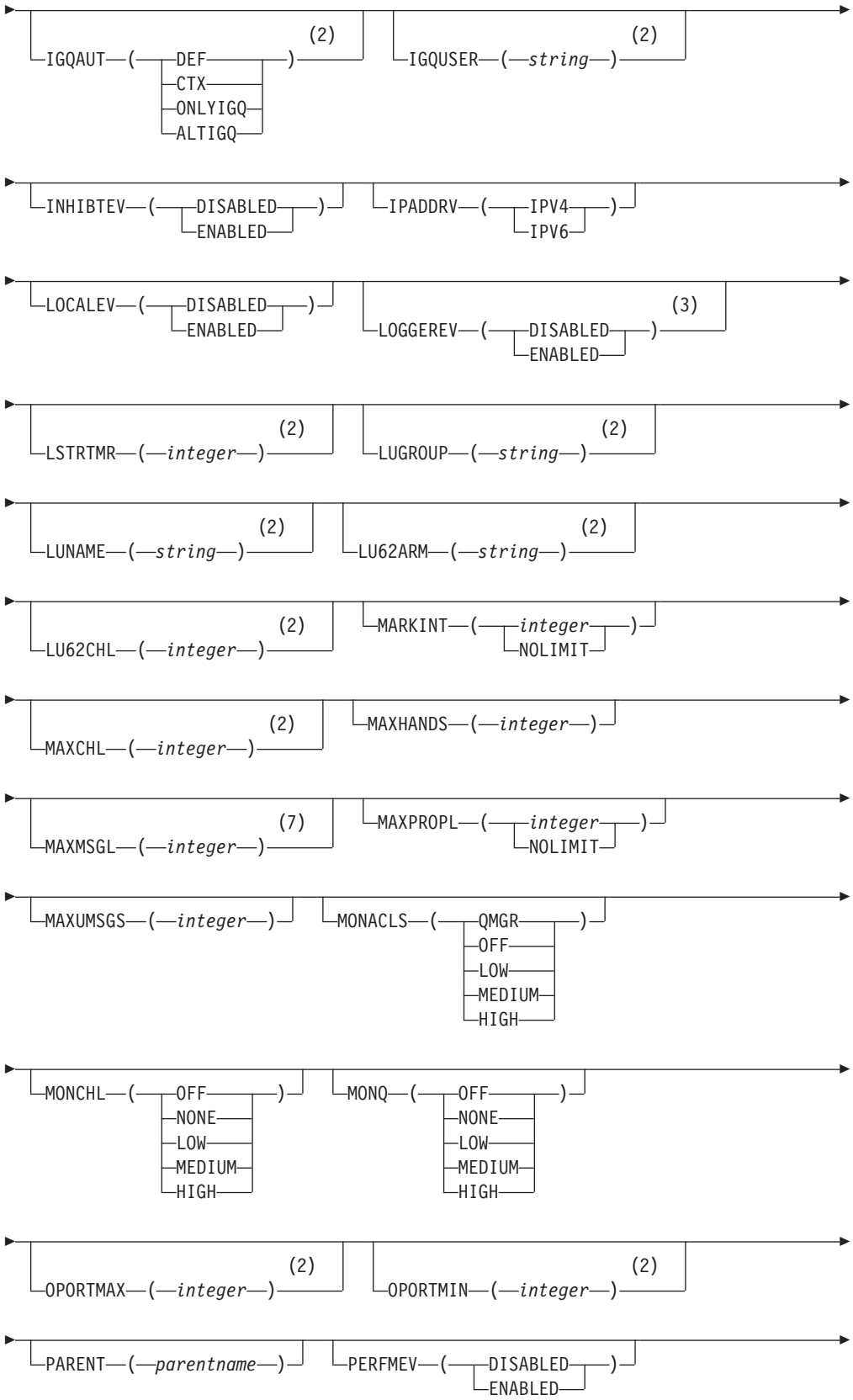
ALTER QMGR

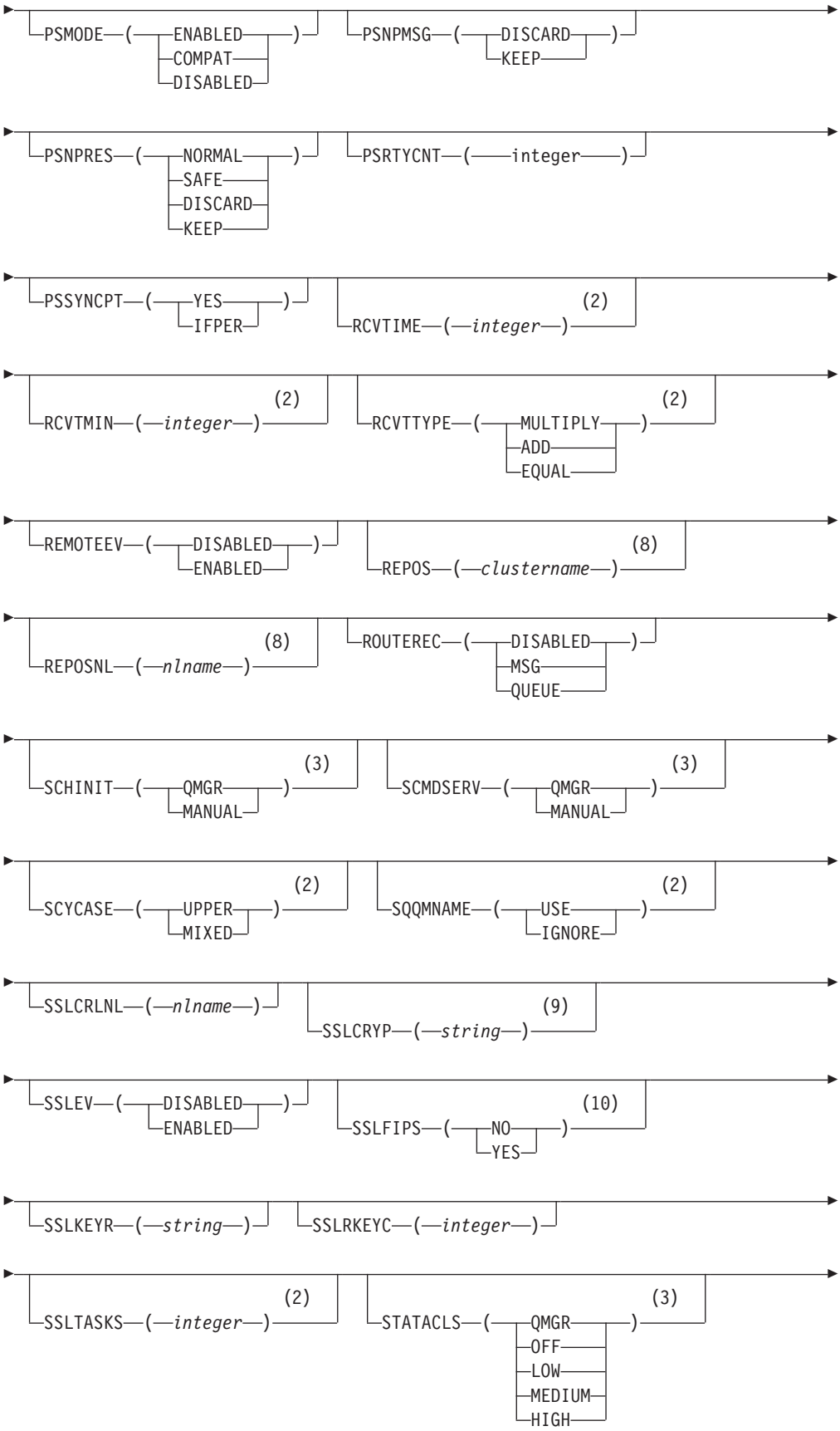


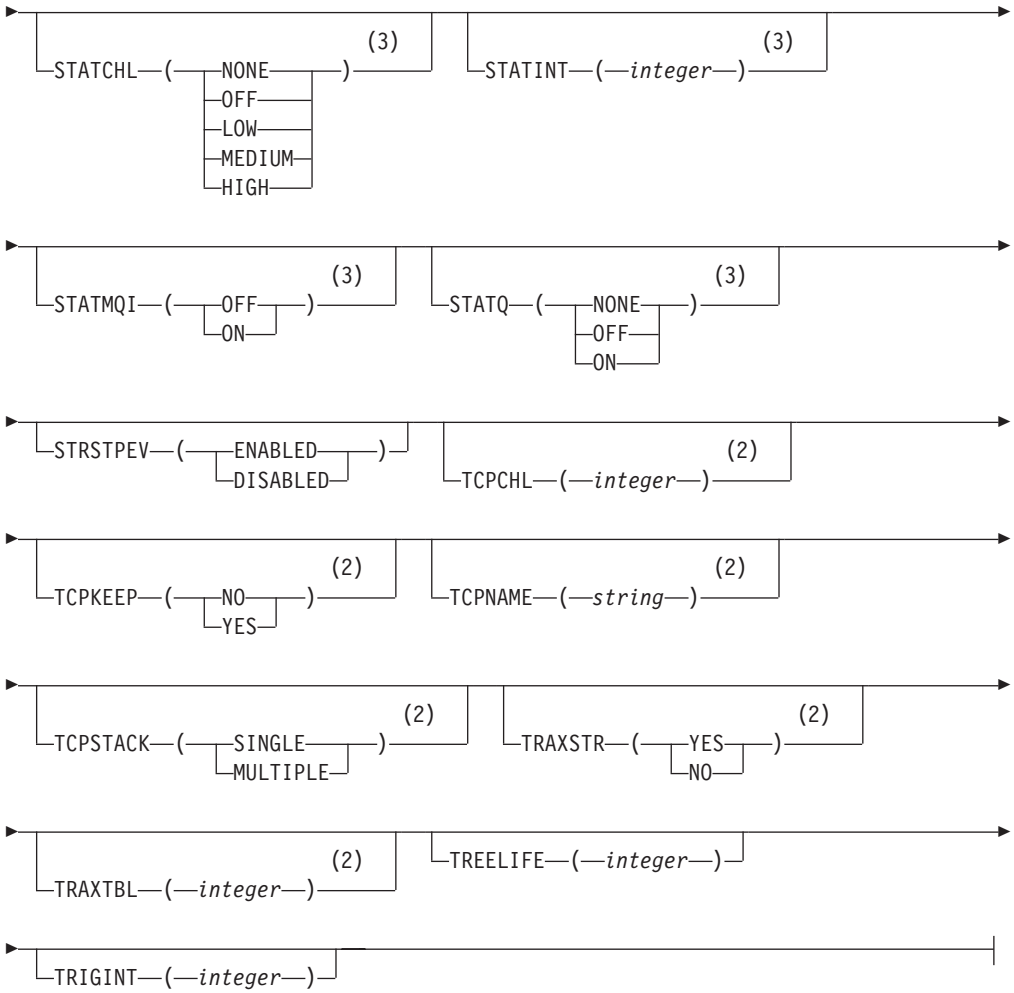
Qmgr attrs:




AUTHOREV
DISABLED
ENABLED
(5)
BRIDGEEV
DISABLED
ENABLED
(2)
CCSID
integer
(6)
CHAD
DISABLED
ENABLED
(7)
CHADEV
DISABLED
ENABLED
(7)
CHADEXIT
string
(8)
CHIADAPS
integer
(2)
CHIDISPS
integer
(2)
CHISERVP
string
(2)
CHLEV
DISABLED
ENABLED
EXCEPTION
(4)
CLWLDATA
string
(8)
CLWLEXIT
string
(8)
CLWLLEN
integer
(8)
CLWLMRUC
integer
CLWLUSEQ
LOCAL
ANY
CMDEV
DISABLED
ENABLED
NODISPLAY
(2)
CONFIGEV
DISABLED
ENABLED
(2)
DEADQ
string
DEFXMITQ
string
DESCR
string
DNSGROUP
string
(2)
DNSWLM
NO
YES
(2)
EXPRYINT
OFF
integer
IGQ
DISABLED
ENABLED
(2)


IGQAUT(DEF | CTX | ONLYIGQ | ALTIGQ) (2)
IGQUSER(string) (2)
INHIBTEV(DISABLED | ENABLED)
IPADDRV(IPV4 | IPV6)
LOCALEV(DISABLED | ENABLED)
LOGGEREV(DISABLED | ENABLED) (3)
LSTRTMR(integer) (2)
LUGROUP(string) (2)
LUNAME(string) (2)
LU62ARM(string) (2)
LU62CHL(integer) (2)
MARKINT(integer | NOLIMIT)
MAXCHL(integer) (2)
MAXHANDS(integer)
MAXMSGL(integer) (7)
MAXPROPL(integer | NOLIMIT)
MAXUMSGS(integer)
MONACLS(QMGR | OFF | LOW | MEDIUM | HIGH)
MONCHL(OFF | NONE | LOW | MEDIUM | HIGH)
MONQ(OFF | NONE | LOW | MEDIUM | HIGH)
OPORTMAX(integer) (2)
OPORTMIN(integer) (2)
PARENT(parentname)
PERFMEV(DISABLED | ENABLED)


PSMODE
ENABLED
COMPAT
DISABLED
PSNPMSG
DISCARD
KEEP
PSNPRES
NORMAL
SAFE
DISCARD
KEEP
PSRTYCNT
integer
PSSYNCPT
YES
IFPER
RCVTIME
integer
(2)
RCVTMIN
integer
(2)
RCVTTYPE
MULTIPLY
ADD
EQUAL
(2)
REMOTEEV
DISABLED
ENABLED
REPOS
clustername
(8)
REPOSNL
nlname
(8)
ROUTEREC
DISABLED
MSG
QUEUE
SCHINIT
QMGR
MANUAL
(3)
SCMDSERV
QMGR
MANUAL
(3)
SCYCASE
UPPER
MIXED
(2)
SQQMNAME
USE
IGNORE
(2)
SSLCRLNL
nlname
SSLCRYP
string
(9)
SSLEV
DISABLED
ENABLED
SSLFIPS
NO
YES
(10)
SSLKEYR
string
SSLRKEYC
integer
SSLTASKS
integer
(2)
STATACLS
QMGR
OFF
LOW
MEDIUM
HIGH
(3)



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

3 Valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

4 Valid only on AIX, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

5 Not valid on z/OS.

6 Valid only on AIX, Compaq NSK, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

7 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

8 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

9 Valid only on AIX, HP-UX, Linux, Solaris, and Windows.

10 Not valid on i5/OS or z/OS.

Parameter descriptions for ALTER QMGR

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

Note:

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

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

Queue manager parameters

Queue manager parameters for the ALTER QMGR command.

ACCTCONO

Specifies whether applications can override the settings of the ACCTQ and ACCTMQI queue manager parameters:

DISABLED

Applications cannot override the settings of the ACCTQ and ACCTMQI parameters.

This is the queue manager's initial default value.

ENABLED

Applications can override the settings of the ACCTQ and ACCTMQI parameters by using the options field of the MQCNO structure of the MQCONNX API call.

Changes to this parameter are effective for connections to the queue manager that occur after the change.

This parameter is valid only on i5/OS, UNIX systems, and Windows.

ACCTINT(*integer*)

The time interval, in seconds, at which intermediate accounting records are written.

Specify a value in the range 1 through 604 800. The queue manager's initial default value is 1800.

Changes to this parameter are effective for connections to the queue manager that occur after the change.

This parameter is valid only on i5/OS, UNIX systems, and Windows.

ACCTMQI

Specifies whether accounting information for MQI data is to be collected:

OFF MQI accounting data collection is disabled. This is the queue manager's initial default value.

ON MQI accounting data collection is enabled.

If queue manager attribute ACCTCONO is set to ENABLED, the value of this parameter can be overridden using the options field of the MQCNO structure.

Changes to this parameter are effective for connections to the queue manager that occur after the change.

This parameter is valid only on i5/OS, UNIX systems, and Windows.

ACCTQ

Specifies whether accounting data is to be collected for all queues. On z/OS, the data collected is class 3 accounting data (thread-level and queue-level accounting).

OFF Accounting data collection is disabled for all queues which specify QMGR as the value for their ACCTQ parameter. This is the queue manager's initial default value, except on z/OS where it is ON.

ON Accounting data collection is enabled for all queues which specify QMGR as the value of their ACCTQ parameter. On z/OS systems, you need to have switched on class 3 accounting by the START TRACE command.

This is the queue manager's initial default value on z/OS.

NONE

Accounting data collection for all queues is disabled regardless of the value of the queue's ACCTQ parameter.

Changes to this parameter are effective only for connections to the queue manager occurring after the change to the parameter.

ACTCHL(*integer***)**

The maximum number of channels that can be active at any time.

Specify a value in the range 1 through 9 999 but no greater than the value of MAXCHL. The queue manager's initial default value is 200.

For an explanation of which channel states are considered active, see *WebSphere MQ Intercommunication*.

If the value of this parameter is reduced, any current channels that exceed the new limit continue to run until they stop.

Changes to the ACTCHL parameter are only honored in a running channel initiator if the new value is less than or equal to the value of ACTCHL when the channel initiator was initialized. If the new value is greater than the value of ACTCHL when the channel initiator was initialized, the new value remains in the queue manager object and takes effect at the next channel initiator restart.

Sharing conversations do not contribute to the total for this parameter.

This parameter is valid on z/OS only.

ACTIVREC

Whether activity reports are generated if requested in the message:

DISABLED

Activity reports are not generated.

MSG Activity reports are generated and sent to the reply queue specified

by the originator in the message causing the report. This is the queue manager's initial default value.

QUEUE
Activity reports are generated and sent to SYSTEM.ADMIN.ACTIVITY.QUEUE.

For more information about activity reports, see Monitoring WebSphere MQ.

ADOPTCHK
Specifies which elements are checked to determine whether an MCA should be adopted when a new inbound channel is detected with the same name as an already active MCA.

ALL Check the queue manager name and the network address. Perform this check to prevent your channels from being inadvertently or maliciously shut down.

This is the queue manager's initial default value.

NETADDR
Check the network address.

NONE
Do no checking.

QMNAME
Check the queue manager name.

This parameter is valid on z/OS only.

Changes to this parameter take effect the next time that a channel attempts to adopt a channel.

ADOPTMCA
Whether an orphaned instance of an MCA should be restarted immediately when a new inbound channel request matching the ADOPTCHK parameter is detetcted:

ALL Adopt all channel types. This is the queue manager's initial default value.

NO Adoption of orphaned channels is not required.

This parameter is valid on z/OS only

Changes to this parameter take effect the next time that a channel attempts to adopt a channel.

AUTHOREV
Whether authorization (Not Authorized) events are generated:

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

ENABLED
Authorization events are generated.

This value is not supported on z/OS.

BRIDGEEV
Whether IMS™ Bridge events are generated.

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

ENABLED
All IMS Bridge events are generated.

This parameter is valid on z/OS only.

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 MQCCSI_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, HP OpenVMS, HP-UX, Linux, i5/OS, 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, HP OpenVMS, HP-UX, Linux, i5/OS, 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, HP OpenVMS, HP-UX, Linux, i5/OS, 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 Windows, it is of the form *dllname(functionname)* where *dllname* is specified without the suffix (".DLL"). The maximum length is 128 characters.
- On i5/OS, it is of the form:

  ```
  progname libname
  ```

  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 of the string is 20 characters.
- On AIX, HP 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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS. On z/OS, it applies only to cluster-sender and cluster-receiver channels.

CHIADAPS(*integer***)**
The number of channel initiator adapter subtasks to use for processing WebSphere MQ calls.

Specify a value in the range zero through 9 999. The queue manager's initial default value is 8.

Suggested settings:

Test system	8 (default)
Production system	30

This parameter is valid on z/OS only.

Changes to this parameter take effect when the channel initiator is restarted.

For a discussion of the relationship between CHIADAPS, CHIDISPS and MAXCHL, see *WebSphere MQ for z/OS System Setup Guide*.

CHIDISPS(*integer***)**
The number of dispatchers to use in the channel initiator.

Specify a value in the range 1 through 9 999. The queue manager's initial default value is 5.

Suggested settings:

Test system	5 (default)
Production system	20

This parameter is valid on z/OS only.

Changes to this parameter take effect when the channel initiator is restarted.

For a discussion of the relationship between CHIADAPS, CHIDISPS and MAXCHL, see *WebSphere MQ for z/OS System Setup Guide*.

CHISERVP
This parameter is reserved for use by IBM® use only; it is not for general use.

This parameter is valid on z/OS only.

CHLEV
Whether channel events are generated.

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

ENABLED
All channel events are generated.

EXCEPTION
All exception channel events are generated.

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, HP OpenVMS, HP-UX, Linux, i5/OS, 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 HP OpenVMS, it is of the form *libraryname(functionname)*. The maximum length is 128 characters.
- On 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 i5/OS, it is of the form:

```
progname libname
```

  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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

CLWLLEN(*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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows and z/OS.

CLWLMRUC(*integer***)**
The maximum number of most recently used outbound cluster channels.

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

The initial default value is 999 999 999.

For more information about this attribute, see WebSphere MQ Queue Manager Clusters.

CLWLUSEQ
For queues whose CLWLUSEQ parameter has a value of QMGR, specifies the behavior of an MQPUT operation when the target queue has a local instance and at least one remote cluster instance (except where the MQPUT originates from a cluster channel).

Specify either:

LOCAL
The local queue is the only target for MQPUT operations. This is the queue manager's initial default value.

ANY The queue manager treats the local queue as another instance of the cluster queue for the purposes of workload distribution.

For more information about this attribute, see WebSphere MQ Queue Manager Clusters.

CMDEV
Specifies whether command events are generated:

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

ENABLED
Command events are generated for all successful commands.

NODISPLAY
Command events are generated for all successful commands, other than DISPLAY commands.

This parameter is 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.

' ' 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 is valid on z/OS only.

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 5.

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 5.

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).

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.

DNSGROUP(*string***)**
The name of the group that the TCP listener handling inbound transmissions for the queue-sharing group should join when using Workload Manager for Dynamic Domain Name Services support (WLM/DNS).

The maximum length of this parameter is eighteen characters.

If this name is blank, the queue-sharing group name is used.

This parameter is valid on z/OS only.

Changes to this parameter take effect for listeners that are subsequently started. Listeners that are currently started are unaffected by changes to this parameter.

DNSWLM
Whether the TCP listener that handles inbound transmissions for the queue-sharing group should register with WLM/DNS:

NO The listener is not to register with Workload Manager. This is the queue manager's initial default value.

YES The listener is to register with Workload Manager.

This parameter is valid on z/OS only.

Changes to this parameter take effect for listeners that are subsequently started. Listeners that are currently started are unaffected by changes to this parameter.

EXPRYINT
Specifies how often queues are scanned to discard 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. Each time that the expiry interval is reached, the queue manager looks for candidate queues that are worth scanning to discard expired messages.

The queue manager maintains information about the expired messages on each queue, and therefore whether a scan for expired messages is worthwhile. So, only a selection of queues is scanned at any time.

The value must be in the range 1 through 99 999 999. The minimum scan interval used is 5 seconds, even if you specify a lower value.

You must set the same EXPRYINT value for all queue managers within a queue-sharing group that support this attribute. Shared queues are scanned by only one queue manager in a queue-sharing group, and this is either the first queue manager to restart or the first queue manager to have had the EXPRYINT value set.

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 valid only on z/OS.

IGQ Whether intra-group queuing is used.

This parameter is valid only on z/OS when the queue manager is a member of a queue-sharing group.

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.

If intra-group queuing has been enabled, but the intra-group queuing agent has stopped for some reason, you can issue ALTER QMGR IGQ(ENABLED) to restart it.

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 is valid only on z/OS when the queue manager is a member of a queue-sharing group.

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 is valid only on z/OS when the queue manager is a member of a queue-sharing group. 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.

Note:

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.

IPADDRV
Specifies which IP protocol is to be used for channel connections.

IPV4 The IPv4 IP address is to be used. This is the queue manager's initial default value.

IPV6 The IPv6 IP address is to be used.

This parameter is used only in systems running IPv4 and IPv6 and it only applies to channels defined with a TRPTYPE of TCP when either of the following two conditions is true:

- The channel's CONNAME parameter contains a hostname that resolves to both an IPv4 and an IPv6 address, and the LOCLADDR parameter is not specified.
- The value of the channel's CONNAME and LOCLADDR parameters is a hostname that resolves to both an IPv4 and IPv6 address.

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.

LOGGEREV
Whether recovery log events are generated:

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

ENABLED
Logger events are generated.

This parameter is valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

LSTRTMR(*integer***)**
The time interval, in seconds, between attempts by WebSphere MQ to restart a listener after an APPC or TCP/IP failure. When the listener is restarted on TCP/IP, it uses the same port and IP address as it used when it first started.

Specify a value in the range 5 through 9 999. The queue manager's initial default value is 60.

This parameter is valid on z/OS only.

Changes to this parameter take effect for listeners that are subsequently started. Listeners that are currently started are unaffected by changes to this parameter.

LUGROUP(*string***)**
The generic LU name to be used by the LU 6.2 listener that handles inbound transmissions for the queue-sharing group. The maximum length of this parameter is 8 characters.

If this name is blank, the listener cannot be used.

This parameter is valid on z/OS only.

Changes to this parameter take effect for listeners that are subsequently started. Listeners that are currently started are unaffected by changes to this parameter.

LUNAME(*string***)**
The name of the LU to use for outbound LU 6.2 transmissions. Set this to be the same as the name of the LU to be used by the listener for inbound transmissions. The maximum length of this parameter is 8 characters.

If this name is blank, the APPC/MVS default LU name should be used. This is variable, so LUNAME should always be set if you are using LU 6.2

This parameter is valid on z/OS only.

Changes to this parameter take effect when the channel initiator is restarted.

LU62ARM(*string***)**
The suffix of the APPCPM member of SYS1.PARMLIB. This suffix nominates the LUADD for this channel initiator. When automatic restart manager (ARM) restarts the channel initiator, the z/OS command SET APPC=xx is issued.

If you do not provide a value for this parameter, no SET APPC=xx command is issued.

The maximum length of this parameter is 2 characters.

This parameter is valid on z/OS only.

Changes to this parameter take effect when the channel initiator is restarted.

LU62CHL(*integer***)**
The maximum number of channels that can be current, or clients that can be connected, that use the LU 6.2 transmission protocol.

Specify a value in the range zero through 9 999 but no greater than the value of MAXCHL. If you specify zero, the LU 6.2 transmission protocol is not used. The queue manager's initial default value is 200.

This parameter is valid on z/OS only.

If the value of this parameter is reduced, any current channels that exceed the new limit continue to run until they stop.

MARKINT(*integer***)**
The time interval, expressed in milliseconds, for which messages that have been marked as browsed by a call to MQGET, with the get message option MQGMO_MARK_BROWSE_CO_OP, are expected to remain mark-browsed.

The queue manager might automatically unmark messages, that have been marked as browsed for the cooperating set of handles, when they have been marked for more than this approximate interval.

This does not affect the state of any message marked as browse by a call to MQGET with the get message option MQGMO_MARK_BROWSE_HANDLE.

Specify a value in the range zero through 999 999 999. The queue manager's initial default value is 5000.

The special value NOLIMIT indicates that the queue manager does not automatically unmark messages by this process.

MAXCHL(*integer***)**
The maximum number of channels that can be current (including server-connection channels with connected clients).

Specify a value in the range zero through 9 999. The queue manager's initial default value is 200. Suggested settings:

Test system	200 (default)
Production system	1000

For an explanation of which channel states are considered current, see *WebSphere MQ Intercommunication*.

If the value of this parameter is reduced, any current channels that exceed the new limit continue to run until they stop.

Changes to the MAXCHL parameter are only honored in a running channel initiator if the new value is less than or equal to the value of MAXCHL when the channel initiator was initialized. If the new value is greater than the value of MAXCHL when the channel initiator was initialized, the new value remains in the queue manager object and takes effect at the next channel initiator restart.

Sharing conversations do not contribute to the total for this parameter.

This parameter is valid on z/OS only.

For a discussion of the relationship between CHIADAPS, CHIDISPS, and MAXCHL, see *WebSphere MQ for z/OS System Setup Guide*.

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

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

On WebSphere MQ for HP NonStop Server, this parameter is ignored.

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 (4 194 304 bytes).

You can use any queue as a shared queue for storing long messages, if its MAXMSGL attribute is set to a value greater than 63 KB. The CFLEVEL parameter of an associated coupling facility structure does not impose any restrictions.

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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

MAXPROPL(*integer***)**
The maximum length of property data in bytes that can be associated with a message.

This is in the range 0 through 100 MB (104 857 600 bytes).

The special value NOLIMIT indicates that the size of the properties is not restricted, except by the upper limit.

This parameter is valid only on i5/OS, UNIX systems, Windows, and z/OS.

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. The limit 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.

Be aware that reducing the value of MAXUMSGS may cause problems to existing applications and queue manager processes, such as clustering on z/OS, if they are already using a higher value.

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

On WebSphere MQ for HP NonStop Server, this parameter is ignored.

MONACLS
Controls the collection of online monitoring data for auto-defined cluster-sender channels:

QMGR
Collection of online monitoring data is inherited from the setting of the queue manager's MONCHL parameter. This is the queue manager's initial default value.

OFF Monitoring for the channel is switched off.

LOW Unless MONCHL is NONE, monitoring is switched on with a low

rate of data collection with a minimal impact on system performance. The data collected is not likely to be the most current.

MEDIUM
Unless MONCHL is NONE, monitoring is switched on with a moderate rate of data collection with limited impact on system performance.

HIGH Unless MONCHL is NONE, monitoring is switched on with a high rate of data collection with a likely impact on system performance. The data collected is the most current available.

A change to this parameter takes effect only on channels started after the change occurs. Any channel started before the change to the parameter continues with the value in force at the time that the channel started.

MONCHL
Controls the collection of online monitoring data for channels.

OFF Online monitoring data collection is turned off for channels specifying a value of QMGR in their MONCHL parameter. This is the queue manager's initial default value.

NONE
Online monitoring data collection is turned off for channels regardless of the setting of their MONCHL parameter.

LOW Online monitoring data collection is turned on, with a low ratio of data collection, for channels specifying a value of QMGR in their MONCHL parameter.

MEDIUM
Online monitoring data collection is turned on, with a moderate ratio of data collection, for channels specifying a value of QMGR in their MONCHL parameter.

HIGH Online monitoring data collection is turned on, with a high ratio of data collection, for channels specifying a value of QMGR in their MONCHL parameter.

A change to this parameter takes effect only on channels started after the change occurs. Any channel started before the change to the parameter continues with the value in force at the time that the channel started.

MONQ
Controls the collection of online monitoring data for queues.

OFF Online monitoring data collection is turned off for queues specifying a value of QMGR in their MONQ parameter. This is the queue manager's initial default value.

NONE
Online monitoring data collection is turned off for queues regardless of the setting of their MONQ parameter.

LOW Online monitoring data collection is turned on for queues specifying a value of QMGR in their MONQ parameter.

MEDIUM
Online monitoring data collection is turned on for queues specifying a value of QMGR in their MONQ parameter.

HIGH Online monitoring data collection is turned on for queues specifying a value of QMGR in their MONQ parameter.

Note that, in contrast to MONCHL, there is no distinction between the values LOW, MEDIUM and HIGH. These values all turn data collection on, but do not affect the rate of collection.

Changes to this parameter are effective only for queues opened after the parameter has been changed.

OPORTMAX(*integer***)**
The maximum value in the range of port numbers to be used when binding outgoing channels. When all the port numbers in the specified range have been used, outgoing channels bind to any available port number.

Specify a value in the range zero through 65 535. The queue manager's initial default value is zero and this means that all outgoing channels bind to any available port number.

Specify a corresponding value for OPORTMIN to define a range of port numbers and ensure that the value you specify for OPORTMAX is greater than or equal to the value you specify for OPORTMIN.

This parameter is valid on z/OS only.

Changes to this parameter take effect for channels that are subsequently started. Channels that are currently started are unaffected by changes to this parameter.

OPORTMIN(*integer***)**
The minimum value in the range of port numbers to be used when binding outgoing channels. When all the port numbers in the specified range have been used, outgoing channels bind to any available port number.

Specify a value in the range zero through 65 535. The queue manager's initial default value is zero.

Specify a corresponding value for OPORTMAX to define a range of port numbers and ensure that the value you specify for OPORTMIN is less than or equal to the value you specify for OPORTMAX.

This parameter is valid on z/OS only.

Changes to this parameter take effect for channels that are subsequently started. Channels that are currently started are unaffected by changes to this parameter.

PARENT(*parentname***)**
The name of the parent queue manager to which the local queue manager is to connect as its child in a hierarchy.

A blank value indicates that the queue manager has no parent queue manager.

If there is an existing parent queue manager it is disconnected.

WebSphere MQ hierarchical connections require that the queue manager attribute PSMODE is set to ENABLED.

The value of PARENT can be set to a blank value if PSMODE is set to DISABLED.

Before a queue manager can connect to a queue manager as its child in a hierarchy, channels must exist in both directions, between the parent queue manager and the child queue manager.

If a parent has already been defined, the ALTER QMGR PARENT command disconnects from the original parent and sends a connection flow to the new parent queue manager.

PERFMEV
Whether performance-related events are generated:

ENABLED
Performance-related events are generated.

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

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

PSMODE
Controls whether the publish/subscribe engine and the queued publish/subscribe interface are running, and therefore controls whether applications can publish or subscribe by using the application programming interface and the queues that are monitored by the queued publish/subscribe interface.

COMPAT
The publish/subscribe engine is running. It is therefore possible to publish or subscribe by using the application programming interface.

The queued publish/subscribe interface is not running. Any publish/subscribe messages put to the queues that are monitored by the queued publish/subscribe interface will not be acted upon.

Use this setting for compatibility with WebSphere Message Broker V6 or earlier versions that use this queue manager, because WebSphere Message Broker needs to read the same queues from which the queued publish/subscribe interface would normally read.

DISABLED
The publish/subscribe engine and the queued publish/subscribe interface are not running. It is therefore not possible to publish or subscribe by using the application programming interface. Any publish/subscribe messages put to the queues that are monitored by the queued publish/subscribe interface will not be acted upon.

ENABLED
The publish/subscribe engine and the queued publish/subscribe interface are running. It is therefore possible to publish or subscribe by using the application programming interface and the queues that are being monitored by the queued publish/subscribe interface. This is the queue manager's initial default value.

Note: Changing the PSMODE attribute can change the PSMODE status. Use "DISPLAY PUBSUB" on page 410, or on i5/OS DSPMQM, to determine the current state of the publish/subscribe engine and the queued publish/subscribe interface.

PSNPMSG

If the Queued Pub/Sub Daemon cannot process a non-persistent input message it might attempt to write the input message to the dead-letter queue (depending on the report options of the input message). If the attempt to write the input message to the dead-letter queue fails, and the MQRO_DISCARD_MSG report option was specified on the input message or the value of PSNPMSG is DISCARD, the daemon discards the input message. If the value of PSNPMSG is KEEP, the daemon will only discard the input message if the MQRO_DISCARD_MSG report option was set in the input message.

Note: If you specify a value of IFPER for PSSYNCPT, you must not specify a value of KEEP for PSNPMSG.

DISCARD
Non-persistent input messages may be discarded if they cannot be processed.

KEEP Non-persistent input messages will not be discarded if they cannot be processed. In this situation the daemon will continue to retry to process this message at appropriate intervals and does not continue processing subsequent messages.

PSNPRES

If the Queued Pub/Sub Daemon attempts to generate a response message in response to a non-persistent input message, and the response message cannot be delivered to the reply-to queue, this attribute indicates whether the daemon should try to write the undeliverable message to the dead-letter queue or whether to discard the message.

For new queue managers, the initial value is NORMAL. If you specify a value of IFPER for PSSYNCPT, you must not specify a value of KEEP or SAFE for PSNPMSG.

For migrated queue managers on Windows, UNIX systems and i5/OS, the value will depend on DLQNonPersistentResponse and DiscardNonPersistentResponse.

NORMAL
Non-persistent responses which cannot be placed on the reply queue are put on the dead-letter queue. If they cannot be placed on the DLQ then they are discarded.

SAFE Non-persistent responses which cannot be placed on the reply queue are put on the dead-letter queue. If the response cannot be sent and cannot be placed on the dead-letter queue then the daemon will back out of the current operation and then retry at appropriate intervals and does not continue processing subsequent messages.

DISCARD
Non-persistent responses which cannot be placed on the reply queue are discarded

KEEP

Non-persistent responses are not placed on the dead-letter queue or discarded. Instead the daemon will back out the current operation and the retry it at appropriate intervals and does not continue processing subsequent messages.

PSRTYCNT

When the Queued Pub/Sub Daemon fails to process a command message under syncpoint (for example a publish message that cannot be delivered to a subscriber because the subscriber queue is full and it is not possible to put the publication on the dead letter queue), the unit of work is backed out and the command retried this number of times before the broker attempts to process the command message according to its report options instead.

The initial value for this parameter on a new queue manager is 5.

Range is 0 to 999 999 999.

PSSYNCPT

Controls whether the Queued Pub/Sub Daemon processes command messages (publishes or delete publication messages) under syncpoint.

YES All messages are processed under syncpoint.

IFPER Persistent messages only are part of the sync point

The queue manager's initial value is IFPER.

QQMNAME

When a queue manager makes an MQOPEN call for a shared queue and the queue manager that is specified in the *ObjectQmgrName* parameter of the MQOPEN call is in the same queue-sharing group as the processing queue manager, the SQQMNAME attribute specifies whether the *ObjectQmgrName* is used or whether the processing queue manager opens the shared queue directly.

USE The *ObjectQmgrName* is used, and the appropriate transmission queue is opened.

IGNORE

The processing queue manager opens the shared queue directly. This can reduce the traffic in your queue manager network.

This parameter is valid only on z/OS.

RCVTIME(*integer***)**

The approximate length of time that a TCP/IP channel waits to receive data, including heartbeats, from its partner before returning to the inactive state. This parameter applies only to message channels (and not to MQI channels).

This number can be qualified as follows:

- To specify that this number is a multiplier to be applied to the negotiated HBINT value to determine how long a channel is to wait, set RCVTTYPE to MULTIPLY. Specify an RCVTIME value of zero or in the range 2 through 99. If you specify zero, the channel does not time out its wait to receive data from its partner.
- To specify that this number is a value, in seconds, to be added to the negotiated HBINT value to determine how long a channel is to wait, set RCVTTYPE to ADD. Specify an RCVTIME value in the range 1 through 999 999.
- To specify that this number is a value, in seconds, that the channel is to wait, set RCVTTYPE to EQUAL. Specify an RCVTIME value in the range zero through 999 999. If you specify zero, the channel does not time out its wait to receive data from its partner.

The queue manager's initial default is zero.

This parameter is valid on z/OS only.

Changes to this parameter take effect for channels that are subsequently started. Channels that are currently started are unaffected by changes to this parameter.

RCVTMIN(*integer***)**
The minimum length of time that a TCP/IP channel waits to receive data, including heartbeats, from its partner before returning to an inactive state. This parameter applies only to message channels (and not to MQI channels).

If you use RCVTTYPE to specify that the TCP/IP channel wait time is relative to the negotiated value of HBINT and the resultant value is less than the value of this parameter, then the value of this parameter is used.

Specify a value, in seconds, between zero and 999 999. The queue manager's initial default is zero.

This parameter is valid on z/OS only.

Changes to this parameter take effect for channels that are subsequently started. Channels that are currently started are unaffected by changes to this parameter.

RCVTTYPE
The qualifier to apply to the value in RCVTIME.

MULTIPLY
Specifies that RCVTIME is a multiplier to be applied to the negotiated HBINT value to determine how long a channel will wait. This is the queue manager's initial default.

ADD Specifies that RCVTIME is a value, in seconds, to be added to the negotiated HBINT value to determine how long a channel will wait.

EQUAL
Specifies that RCVTIME is a value, in seconds, representing how long the channel will wait.

This parameter is valid on z/OS only.

Changes to this parameter take effect for channels that are subsequently started. Channels that are currently started are unaffected by changes to this parameter.

REMOTEEV
Whether remote error events are generated:

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

ENABLED
Remote error events are generated.

If you are using the reduced function form of WebSphere MQ for z/OS supplied with WebSphere Application Server, only DISABLED is valid.

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 and REPOSNL can be nonblank.

If you use the REPOS parameter to create a full repository queue manager, use cluster-sender channels to connect it to at least one other full repository queue manager in the cluster. See the information in WebSphere MQ Queue Manager Clusters for details about using cluster-sender channels with full repository queue managers.

This parameter is valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, 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 REPOS and 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.

If you use the REPOSNL parameter to create a full repository queue manager, use cluster-sender channels to connect it to at least one other full repository queue manager in each cluster specified in the namelist. See the information in WebSphere MQ Queue Manager Clusters for details about using cluster-sender channels with full repository queue managers.

This parameter is valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

ROUTEREC
Whether trace-route information is recorded if requested in the message. If participation in route tracing is enabled by this queue manager parameter not being set to DISABLED, the value of the parameter is only important should a reply be generated. The reply goes either to SYSTEM.ADMIN.TRACE.ROUTE.QUEUE, or to the destination specified by the message itself. Provided ROUTEREC is not DISABLED, messages not yet at the final destination may have information added to them.

DISABLED
Trace-route information is not recorded.

MSG Trace-route information is recorded and sent to the destination specified by the originator of the message causing the trace route record. This is the queue manager's initial default value.

QUEUE
Trace-route information is recorded and sent to SYSTEM.ADMIN.TRACE.ROUTE.QUEUE.

For more information about trace-route records, see Monitoring WebSphere MQ.

SCHINIT
Whether the channel initiator should start automatically when the queue manager starts.

QMGR
The channel initiator will start automatically when the queue manager starts. This is the default value.

MANUAL
The channel initiator will not start automatically.

This parameter is valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

SCMDSERV
Whether the command server should start automatically when the queue manager starts.

QMGR
The command server will start automatically when the queue manager starts. This is the default value.

MANUAL
The command server will not start automatically.

This parameter is valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

SCYCASE
Whether the security profiles are uppercase or mixed case.

UPPER
The security profiles are uppercase only. This is the default value.

MIXED
The security profiles are mixed case.

Changes to SCYCASE become effective after you run the following command:

```
REFFRESH SECURITY(*) TYPE(CLASSES)
```

This parameter is valid only on z/OS

SSLCRLNL*(nlname)*
The name of a namelist of authentication information objects to be used for Certificate Revocation List (CRL) checking by the queue manager.

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 authentication information objects become effective either:

- On i5/OS, Windows, and UNIX systems when a new channel process is started.
- For channels that run as threads of the channel initiator on i5/OS, Windows, and UNIX systems, when the channel initiator is restarted.
- For channels that run as threads of the listener on i5/OS, Windows, and UNIX systems, when the listener is restarted.
- On z/OS, when the channel initiator is restarted.
- When a REFRESH SECURITY TYPE(SSL) command is issued.
- On i5/OS queue managers, this parameter is ignored. However, it is used to determine which authentication information objects are written to the AMQCLCHL.TAB file.

SSLCRYP*(string)*
Sets the name of the parameter string required to configure the cryptographic hardware present on the system.

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 PKCS #11 driver path and filename>;<the PKCS #11 token label>;<the PKCS #11 token password>;<symmetric cipher setting>;*

The strings containing RAINBOW enable or disable the Rainbow CryptoSwift cryptographic hardware.

The strings containing NCIPHER enable or disable the nCipher nFast cryptographic hardware.

To use cryptographic hardware which is accessed using the PKCS #11 interface, you must specify the string containing PKCS11. The PKCS #11 driver path is an absolute path to the shared library providing support for the PKCS #11 card. The PKCS #11 driver filename is the name of the shared library. An example of the value required for the PKCS #11 driver path and filename is `/usr/lib/pkcs11/PKCS11_API.so`

To access symmetric cipher operations through GSKit, specify the symmetric cipher setting parameter on the PKCS11 string. The value of this parameter is either:

SYMMETRIC_CIPHER_OFF
Do not access symmetric cipher operations.

SYMMETRIC_CIPHER_ON
Access symmetric cipher operations.

If the symmetric cipher setting parameter is not specified, this has the same effect as specifying SYMMETRIC_CIPHER_OFF.

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.
- When a REFRESH SECURITY TYPE(SSL) command is issued.

SSLEV
Whether SSL events are generated.

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

ENABLED
All SSL events are generated.

SSLFIPS
This parameter is valid only on Windows and UNIX platforms.

SSSLFIPS specifies whether only FIPS-certified algorithms are to be used if cryptography is carried out in WebSphere MQ. If cryptographic hardware is configured, the cryptographic modules used are those provided by the hardware product, and these may, or may not, be FIPS-certified to a particular level. This depends on the hardware product in use. For more information about FIPS, see the WebSphere MQ Security manual.

NO WebSphere MQ provides an implementation of SSL cryptography, which supplies FIPS-certified modules for all Version 7.0 platforms. If you set SSLFIPS to NO, any CipherSpec supported on a particular platform can be used. This is the queue manager's initial default value.

If the queue manager runs without using cryptographic hardware, the following CipherSpecs run using FIPS 140–2 certified cryptography:

- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- FIPS_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA

YES Specifies that only FIPS-certified algorithms are to be used in the CipherSpecs allowed on all SSL connections from and to this queue manager.

Inbound and outbound SSL channel connections succeed only if one of the following CipherSpecs is used:

- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- FIPS_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA

SSLKEYR*(string)*

The name of the Secure Sockets Layer key repository.

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 i5/OS, 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`.

  If you specify *SYSTEM, WebSphere MQ utilizes the system certificate store as the key repository for the queue manager. As a result, the queue manager is registered as a server application in Digital Certificate Manager (DCM) and you can assign any server/client certificate in the system store to this application.

  If you change the SSLKEYR parameter to a value other than *SYSTEM, WebSphere MQ deregisters the queue manager as an application with DCM.
- On UNIX it is of the form *pathname/keyfile* and on Windows *pathname\keyfile*, where *keyfile* is specified without the suffix (.kdb), and identifies a GSKit key database file. The default value for UNIX platforms is `/var/mqm/qmgrs/QMGR/ssl/key`, and on Windows it is `C:\Program Files\IBM\WebSphere MQ\qmgrs\QMGR\ssl\key`, where `QMGR` is replaced by the queue manager name (on UNIX and Windows).

On i5/OS, Windows and UNIX systems, 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 i5/OS, Windows and UNIX systems, when a new channel process is started.
- For channels that run as threads of the channel initiator on i5/OS, Windows and UNIX systems, when the channel initiator is restarted.
- For channels that run as threads of the listener on i5/OS, Windows and UNIX systems, when the listener is restarted.
- For channels that run as threads of the process pooling process (amqrmppa), when the queue manager is restarted.
- On z/OS, when the channel initiator is restarted.
- When a REFRESH SECURITY TYPE(SSL) command is issued.

SSLRKEYC*(integer)*

The number of unencrypted bytes to be sent and received within an SSL conversation before the secret key is renegotiated. The number of bytes includes control information.

This value is used only by SSL channels which initiate communication from the queue manager (for example, the sender channel in a sender and receiver channel pairing).

If a value greater than zero is specified and channel heartbeats are enabled for the channel, the secret key is also renegotiated before message data is sent or received following a channel heartbeat. The count of bytes until the next secret key renegotiation is reset after each successful renegotiation.

Specify a value in the range zero through 999 999 999. A value of zero (the queue manager's initial default value) means that the secret key is never renegotiated.

Attention: Non-zero values less than 4096 (4Kb) may cause channels to fail to start, or may cause inconsistencies in the values of SSLKEYDA, SSLKEYTI, and SSLRKEYS.

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. To avoid problems with storage allocation, do not set the SSLTASKS parameter to a value greater than 50.

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

STATACLS

Whether statistics data is to be collected for auto-defined cluster-sender channels:

QMGR

Collection of statistics data is inherited from the setting of the queue manager's STATCHL parameter. This is the queue manager's initial default value.

OFF Statistics data collection for the channel is switched off.

LOW Unless STATCHL is NONE, statistics data collection is switched on with a low ratio of data collection with a minimal impact on system performance.

MEDIUM
Unless STATCHL is NONE, statistics data collection is switched on with a moderate ratio of data collection.

HIGH Unless STATCHL is NONE, statistics data collection is switched on with a high ratio of data collection.

This parameter is valid only on i5/OS, UNIX systems, and Windows.

A change to this parameter takes effect only on channels started after the change occurs. Any channel started before the change to the parameter continues with the value in force at the time that the channel started.

STATCHL
Whether statistics data is to be collected for channels:

NONE
Statistics data collection is turned off for channels regardless of the setting of their STATCHL parameter.

OFF Statistics data collection is turned off for channels specifying a value of QMGR in their STATCHL parameter. This is the queue manager's initial default value.

LOW Statistics data collection is turned on, with a low ratio of data collection, for channels specifying a value of QMGR in their STATCHL parameter.

MEDIUM
Statistics data collection is turned on, with a moderate ratio of data collection, for channels specifying a value of QMGR in their STATCHL parameter.

HIGH Statistics data collection is turned on, with a high ratio of data collection, for channels specifying a value of QMGR in their STATCHL parameter.

This parameter is valid only on i5/OS, UNIX systems, and Windows.

A change to this parameter takes effect only on channels started after the change occurs. Any channel started before the change to the parameter continues with the value in force at the time that the channel started.

STATINT(*integer***)**
The time interval, in seconds, at which statistics monitoring data is written to the monitoring queue.

Specify a value in the range 1 through 604 800. The queue manager's initial default value is 1800.

This parameter is valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

Changes to this parameter take immediate effect on the collection of monitoring and statistics data.

STATMQI
Whether statistics monitoring data is to be collected for the queue manager:

OFF Data collection for MQI statistics is disabled. This is the queue manager's initial default value.

ON Data collection for MQI statistics is enabled.

This parameter is valid only on i5/OS, UNIX systems, and Windows.

Changes to this parameter take immediate effect on the collection of monitoring and statistics data.

STATQ

Whether statistics data is to be collected for queues:

NONE
Statistics data collection is turned off for queues regardless of the setting of their STATQ parameter. This is the queue manager's initial default value.

OFF Statistics data collection is turned off for queues specifying a value of QMGR in their STATQ parameter.

ON Statistics data collection is turned on for queues specifying a value of QMGR in their STATQ parameter.

This parameter is valid only on i5/OS, UNIX systems, and Windows.

Changes to this parameter are effective only for connections to the queue manager occurring after the change to the parameter.

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.

TCPCHL(*integer***)**

The maximum number of channels that can be current, or clients that can be connected, that use the TCP/IP transmission protocol.

The maximum number of sockets used is the sum of the values in TCPCHL and CHIDISPS. The z/OS OpenEdition MAXFILEPROC parameter (specified in the BPXPRMxx member of SYS1.PARMLIB) controls how many sockets each task is allowed, and thus how many channels each dispatcher is allowed. In this case, the number of channels using TCP/IP is limited to the value of MAXFILEPROC multiplied by the value of CHIDISPS.

Specify a value in the range zero through 9 999 but no greater than the value of MAXCHL (although note that TCP/IP might not support as many as 9 999 channels). If you specify zero, the TCP/IP transmission protocol is not used. The queue manager's initial default is 200.

If the value of this parameter is reduced, any current channels that exceed the new limit continue to run until they stop.

Sharing conversations do not contribute to the total for this parameter.

This parameter is valid on z/OS only.

TCPKEEP
Specifies whether the KEEPALIVE facility is to be used to check that the other end of the connection is still available. If it is not available, the channel is closed.

NO The TCP KEEPALIVE facility is not to be used. This is the queue manager's initial default value.

YES The TCP KEEPALIVE facility is to be used as specified in the TCP profile configuration data set. The interval is specified in the KAINT channel attribute.

This parameter is valid on z/OS only.

Changes to this parameter take effect for channels that are subsequently started. Channels that are currently started are unaffected by changes to this parameter.

TCPNAME(*string*)
The name of either the only, or default, TCP/IP system to be used, depending on the value of TCPSTACK. This is the name of the z/OS OpenEdition stack for TCP/IP, as specified in the SUBFILESYSTYPE NAME parameter in the BPXPRMxx member of SYS1.PARMLIB. The queue manager's initial default value is 'TCPIP'.

The maximum length of this parameter is 8 characters.

This parameter is valid on z/OS only.

Changes to this parameter take effect when the channel initiator is restarted.

TCPSTACK
Specifies whether the channel initiator may use only the TCP/IP address space specified in TCPNAME, or may optionally bind to any selected TCP/IP address.

SINGLE
The channel initiator may only use the TCP/IP address space specified in TCPNAME. This is the queue manager's default value.

MULTIPLE
The channel initiator may use any TCP/IP address space available to it. It defaults to the one specified in TCPNAME if no other is specified for a channel or listener.

This parameter is valid on z/OS only.

Changes to this parameter take effect when the channel initiator is restarted.

TRAXSTR
Specifies whether the channel initiator trace should start automatically:

YES Channel initiator trace is to start automatically. This is the queue manager's default value.

NO Channel initiator trace is not to start automatically.

This parameter is valid on z/OS only.

Changes to this parameter take effect when the channel initiator is restarted. If you wish to start or stop channel initiator trace without restarting the channel initiator, use the START TRACE or STOP TRACE commands after the channel initiator has started.

TRAXTBL(*integer***)**
The size, in megabytes, of the channel initiator's trace data space.

Specify a value in the range 2 through 2048. The queue manager's initial default value is 2.

This parameter is valid on z/OS only.

Note:

1. Changes to this parameter take effect immediately; any existing trace table contents are lost.
2. When you use large z/OS data spaces, ensure that sufficient auxiliary storage is available on your system to support any related z/OS paging activity. You might also need to increase the size of your SYS1.DUMP data sets.

TREELIFE(*integer***)**
The lifetime, in seconds of non-administrative topics.

Non-administrative topics are those created when an application publishes to, or subscribes on, a topic string that does not exist as an administrative node. When this non-administrative node no longer has any active subscriptions, this parameter determines how long the queue manager will wait before removing that node. Only non-administrative topics that are in use by a durable subscription remain after the queue manager is recycled.

Specify a value in the range 0 through 604 000. A value of 0 means that non-administrative topics are not removed by the queue manager. The queue manager's initial default value is 1800.

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 232 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

Use the MQSC command ALTER command to alter the parameters of a local queue (ALTER QLOCAL), alias queue (ALTER QALIAS), model queue (ALTER QMODEL), a remote queue, a queue-manager alias, or a reply-to queue alias (ALTER QREMOTE).

This section contains the following commands:

- "ALTER QALIAS" on page 79
- "ALTER QLOCAL" on page 81
- "ALTER QMODEL" on page 85
- "ALTER QREMOTE" on page 88

These commands are supported on the following platforms:

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

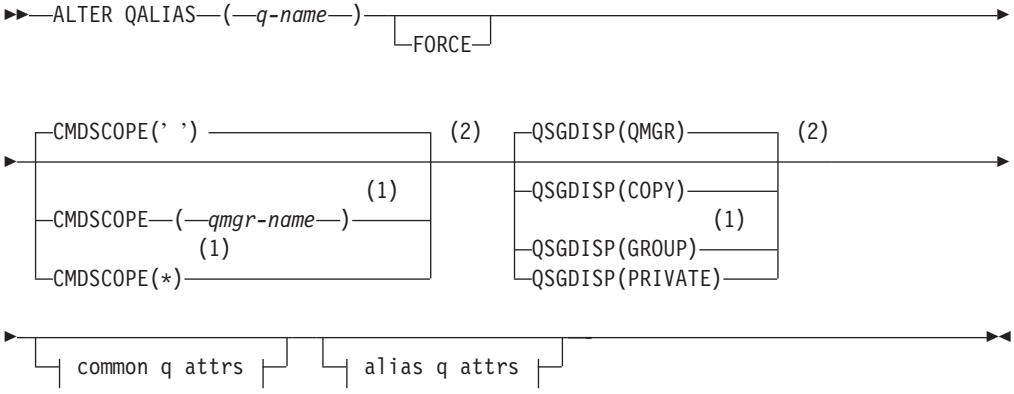
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

ALTER QALIAS

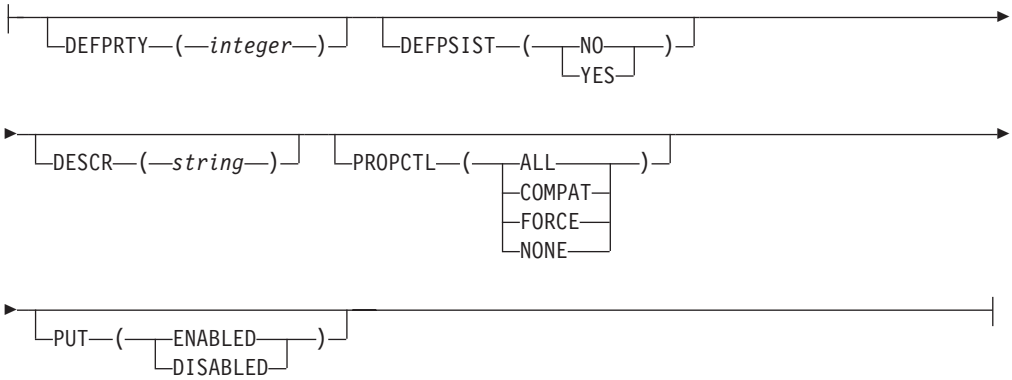
Use the MQSC command ALTER QALIAS to alter the parameters of an alias queue.

Synonym: ALT QA

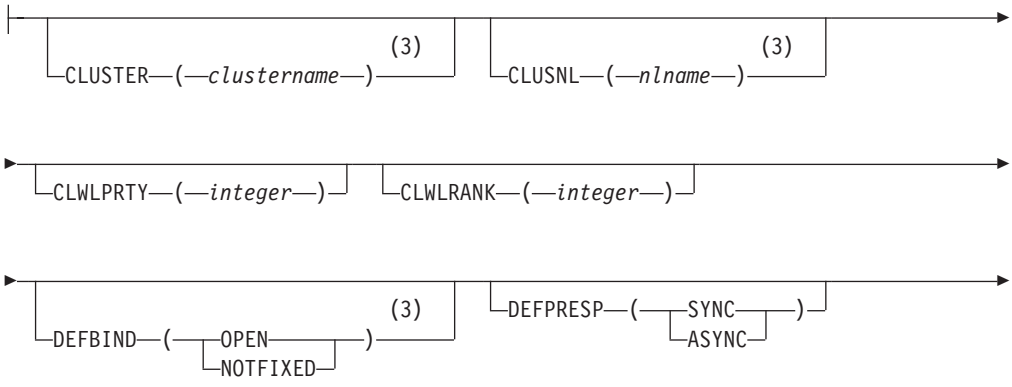
ALTER QALIAS

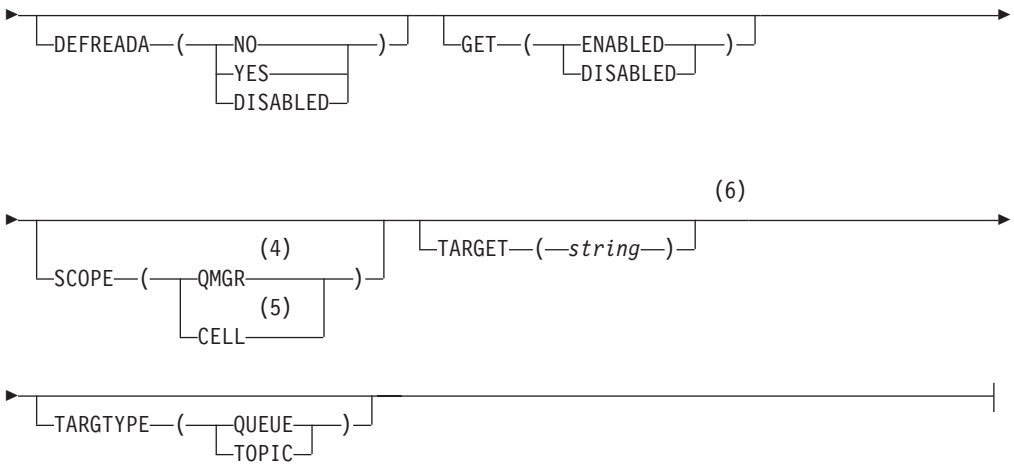


Common q attrs:



Alias q attrs:





Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

3 Valid only on AIX, HP-UX, z/OS, i5/OS, Solaris, and Windows.

4 Valid only on HP OpenVMS, i5/OS, UNIX systems, and Windows.

5 Valid only on HP OpenVMS, UNIX systems, and Windows.

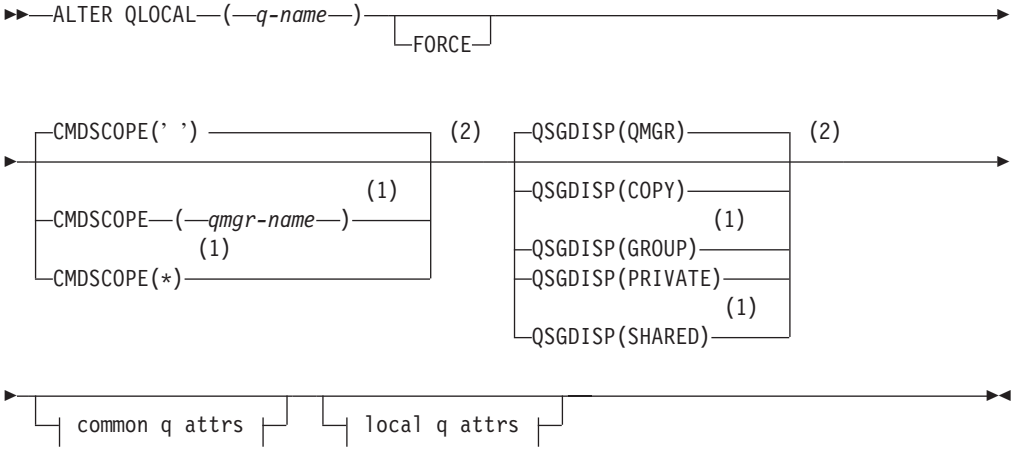
6 The TARGQ parameter is available for compatibility with previous releases. It is a synonym of TARGET; you cannot specify both parameters.

ALTER QLOCAL

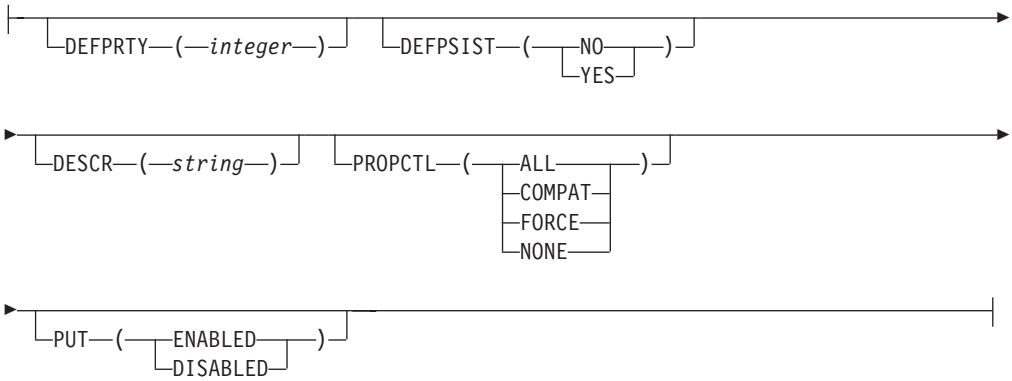
Use the MQSC command ALTER QLOCAL to alter the parameters of a local queue.

Synonym: ALT QL

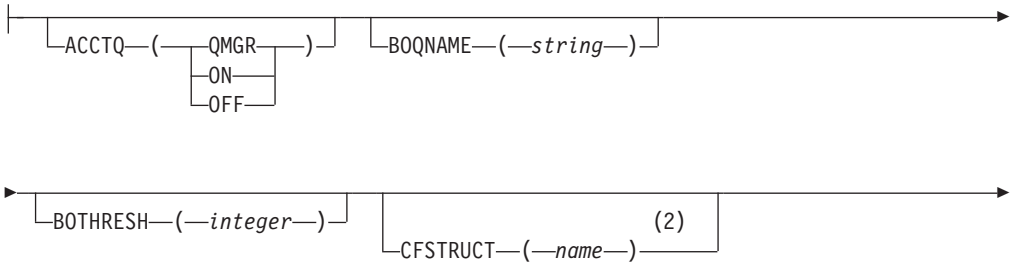
ALTER QLOCAL

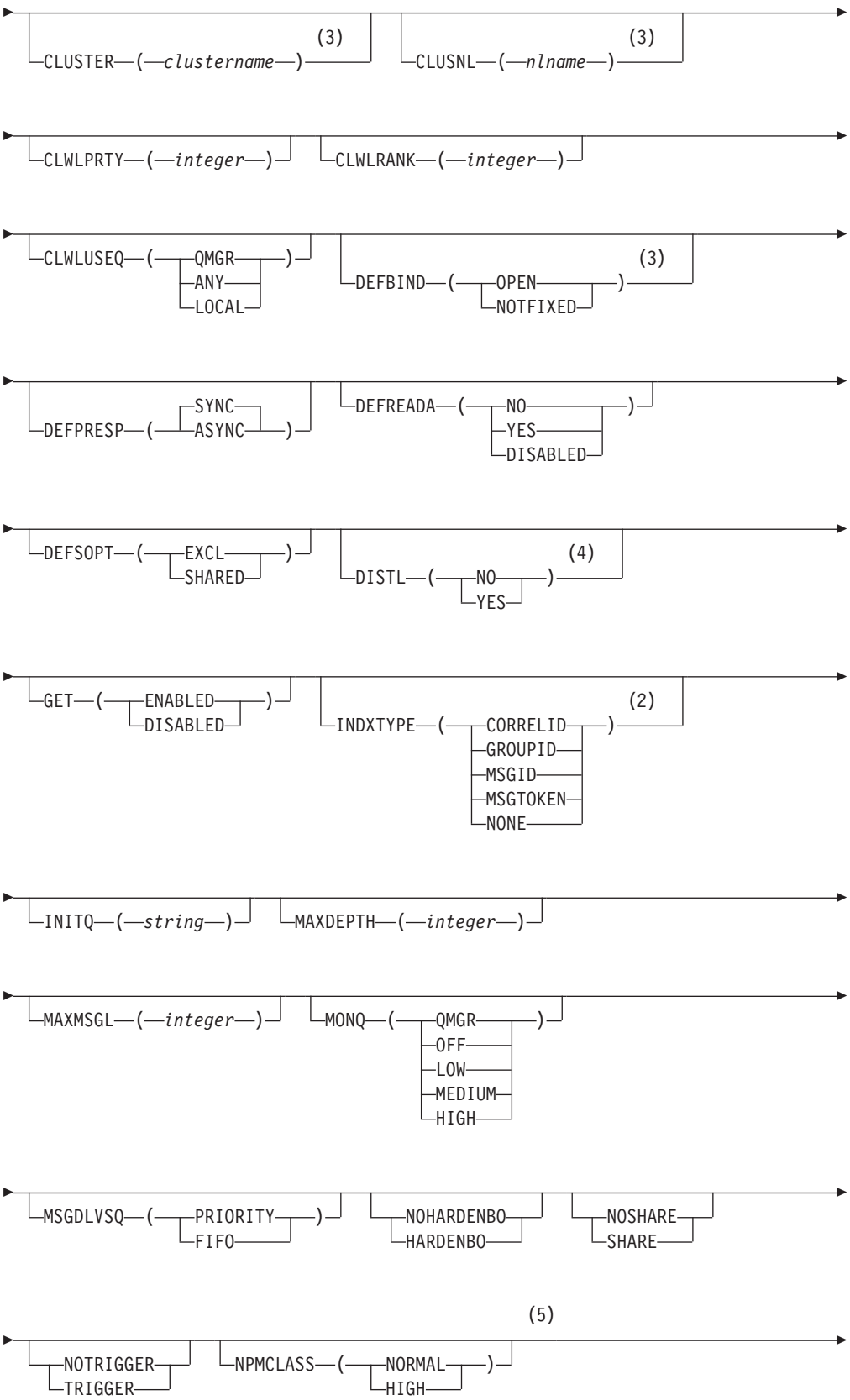


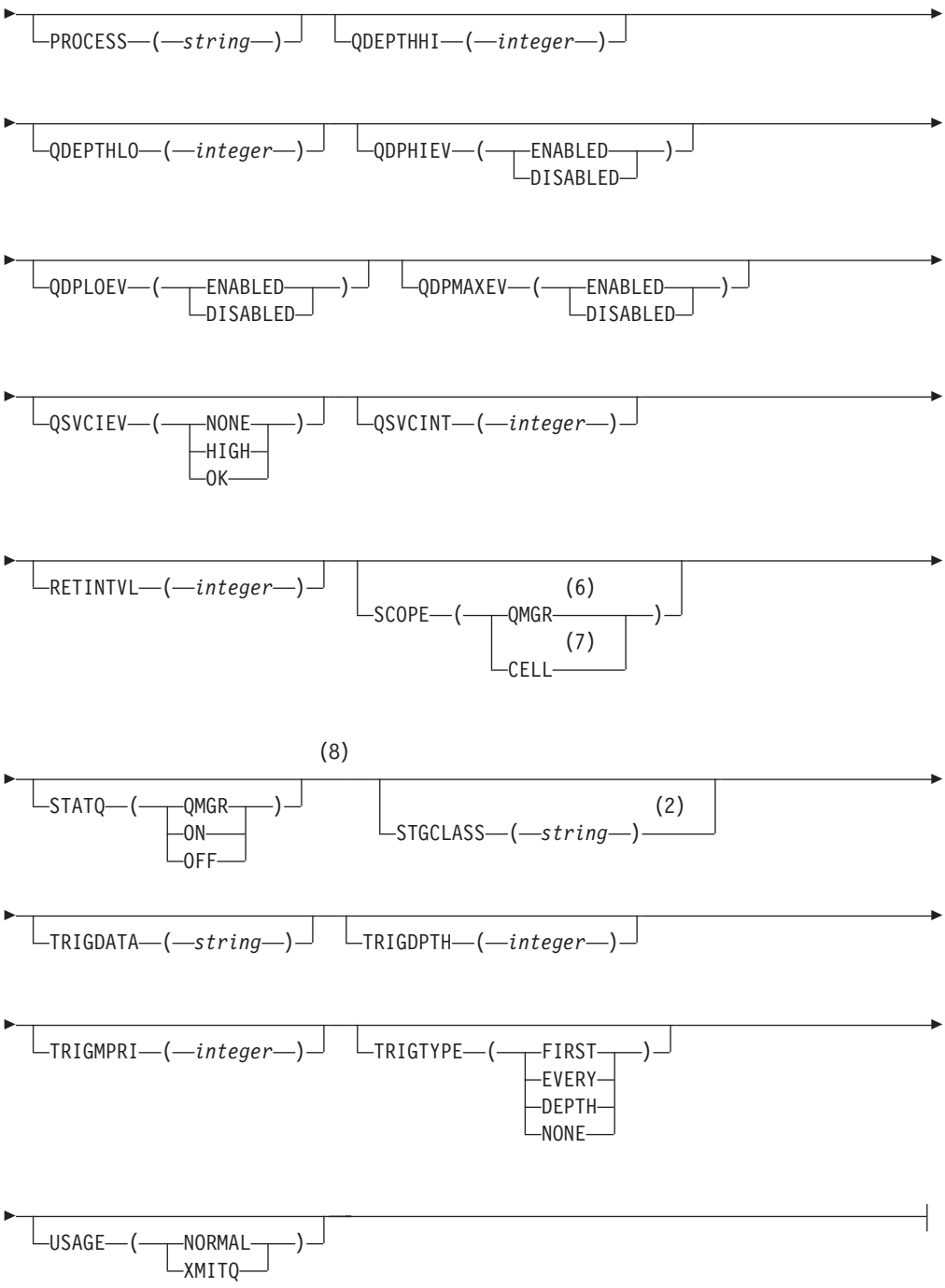
Common q attrs:



Local q attrs:




CLUSTER(clustername) (3)
CLUSNL(nlname) (3)
CLWLPRTY(integer)
CLWLRANK(integer)
CLWLUSEQ(QMGR | ANY | LOCAL)
DEFBIND(OPEN | NOTFIXED) (3)
DEFPRESP(SYNC | ASYNC)
DEFREADA(NO | YES | DISABLED)
DEFSOPT(EXCL | SHARED)
DISTL(NO | YES) (4)
GET(ENABLED | DISABLED)
INDXTYPE(CORRELID | GROUPID | MSGID | MSGTOKEN | NONE) (2)
INITQ(string)
MAXDEPTH(integer)
MAXMSGL(integer)
MONQ(QMGR | OFF | LOW | MEDIUM | HIGH)
MSGDLVSQ(PRIORITY | FIFO)
NOHARDENBO | HARDENBO
NOSHARE | SHARE
NOTRIGGER | TRIGGER
NPMCLASS(NORMAL | HIGH) (5)



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

3 Valid only on AIX, HP-UX, z/OS, i5/OS, Solaris, and Windows.

4 Valid only on AIX, HP-UX, i5/OS, Solaris, and Windows.

5 Not valid on z/OS.

6 Valid only on HP OpenVMS, i5/OS, UNIX systems, and Windows.

7 Valid only on HP OpenVMS, UNIX systems, and Windows.

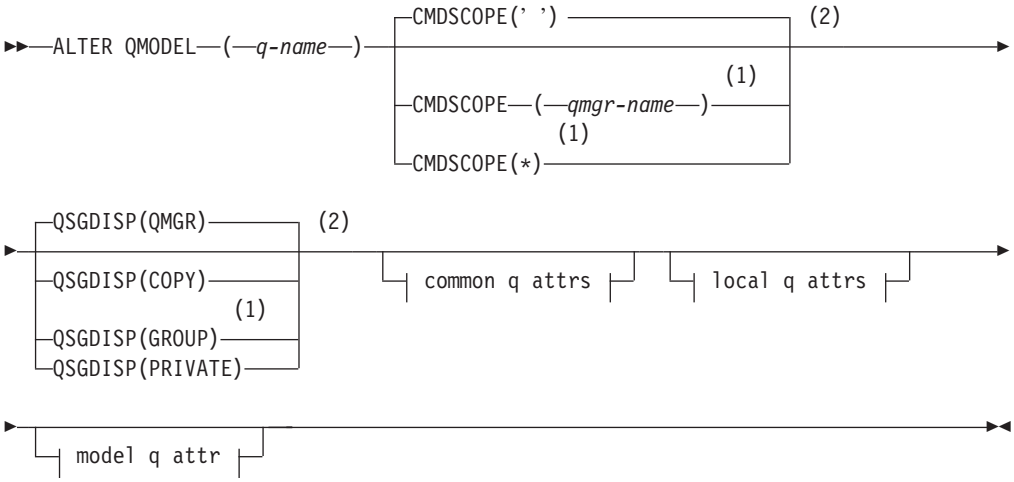
8 Valid only on i5/OS, UNIX systems, and Windows.

ALTER QMODEL

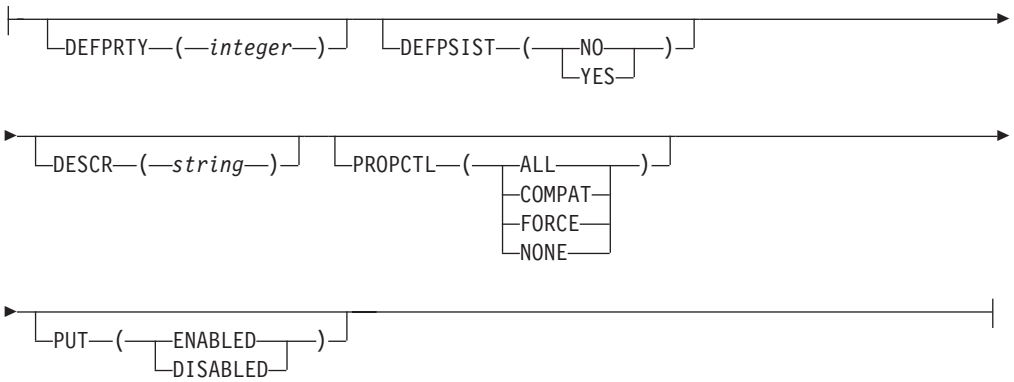
Use the MQSC command ALTER QMODEL to alter the parameters of a model queue.

Synonym: ALT QM

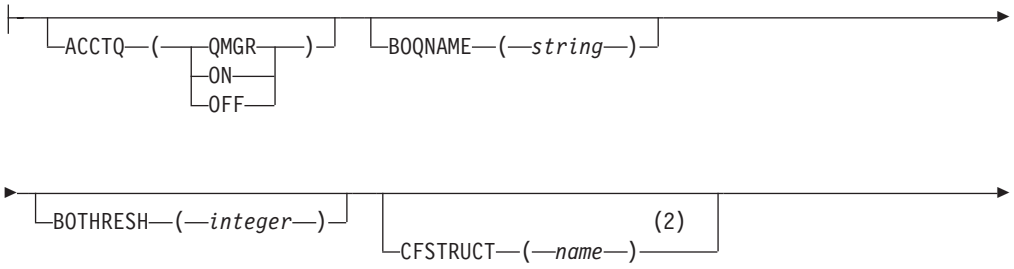
ALTER QMODEL

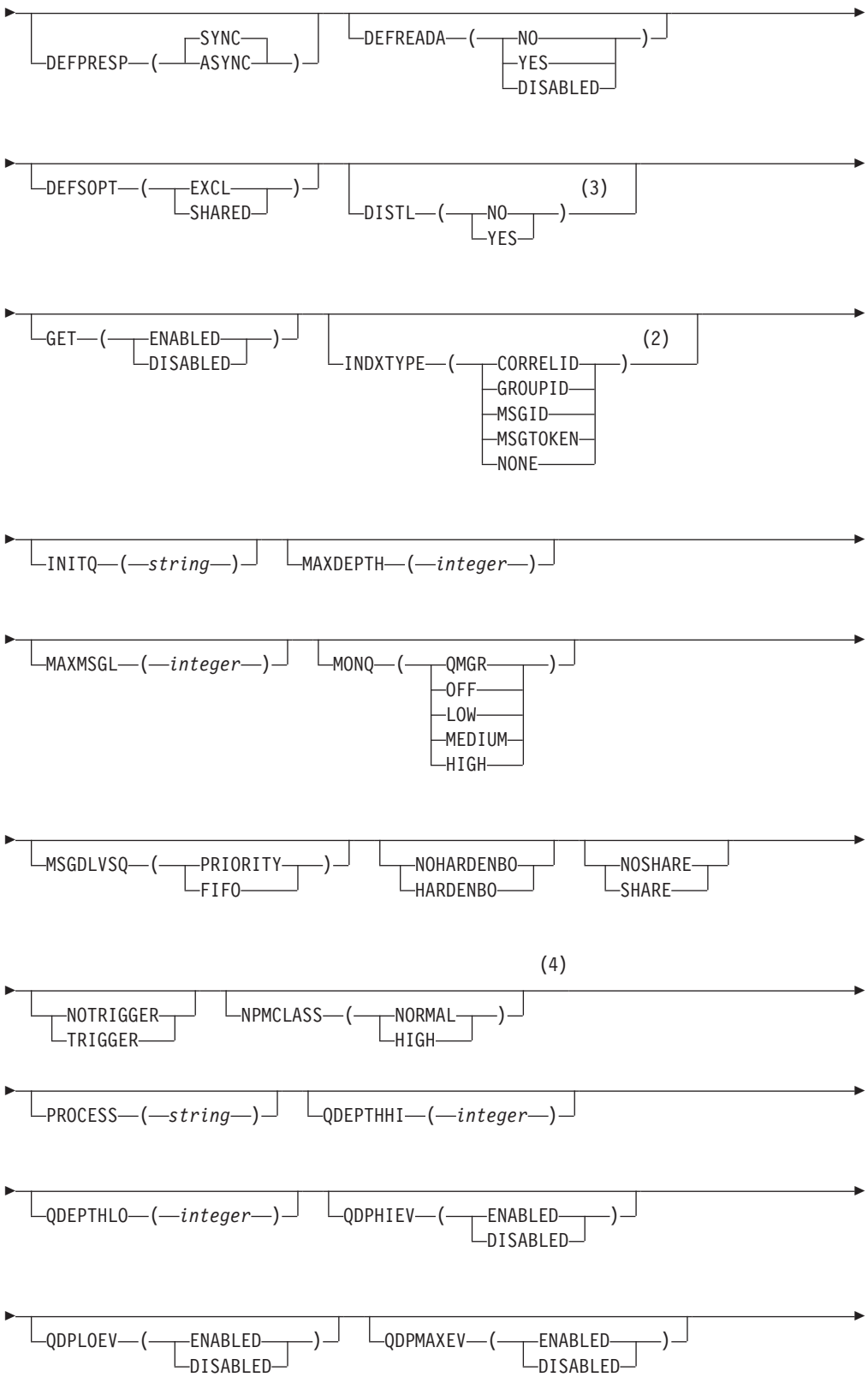


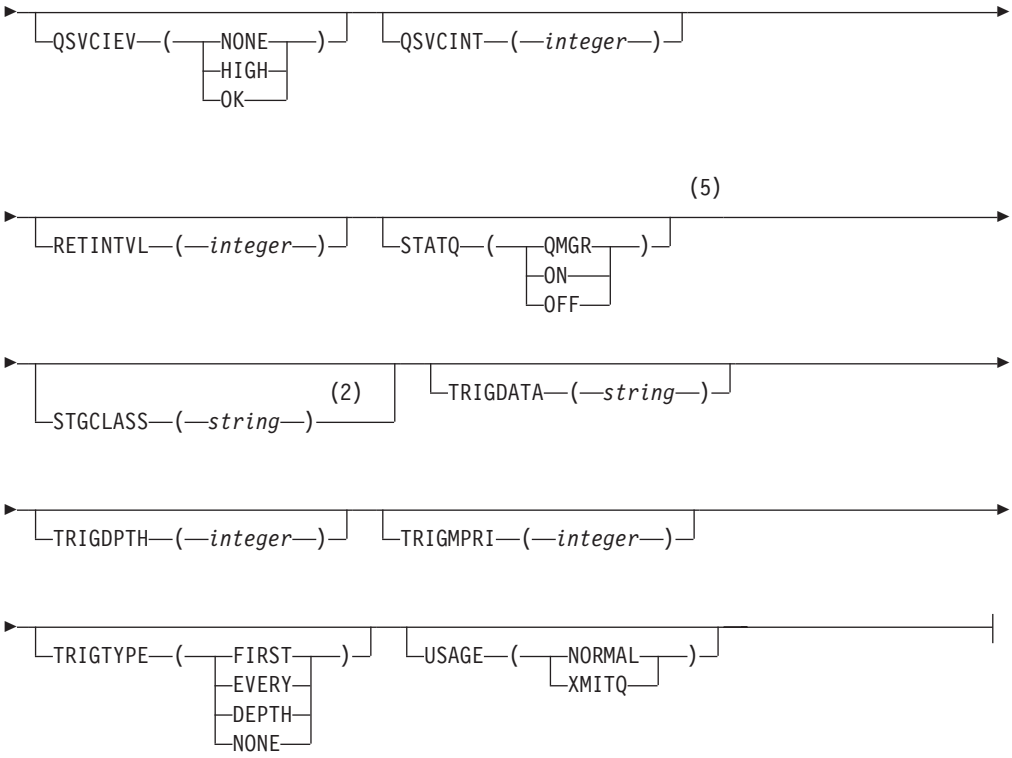
Common q attrs:



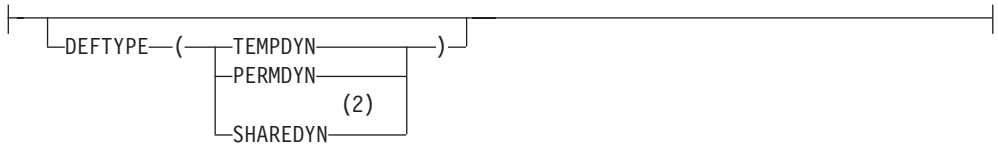
Local q attrs:




DEFPRESP
SYNC
ASYNC
DEFREADA
NO
YES
DISABLED
DEFSOPT
EXCL
SHARED
DISTL
NO
YES
(3)
GET
ENABLED
DISABLED
INDXTYPE
CORRELID
GROUPID
MSGID
MSGTOKEN
NONE
(2)
INITQ
string
MAXDEPTH
integer
MAXMSGL
integer
MONQ
QMGR
OFF
LOW
MEDIUM
HIGH
MSGDLVSQ
PRIORITY
FIFO
NOHARDENBO
HARDENBO
NOSHARE
SHARE
NOTRIGGER
TRIGGER
NPMCLASS
NORMAL
HIGH
(4)
PROCESS
string
QDEPTHHI
integer
QDEPTHLO
integer
QDPHIEV
ENABLED
DISABLED
QDPLOEV
ENABLED
DISABLED
QDPMAXEV
ENABLED
DISABLED



Model q attr:



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

3 Valid only on AIX, HP-UX, i5/OS, Solaris, and Windows.

4 Not valid on z/OS.

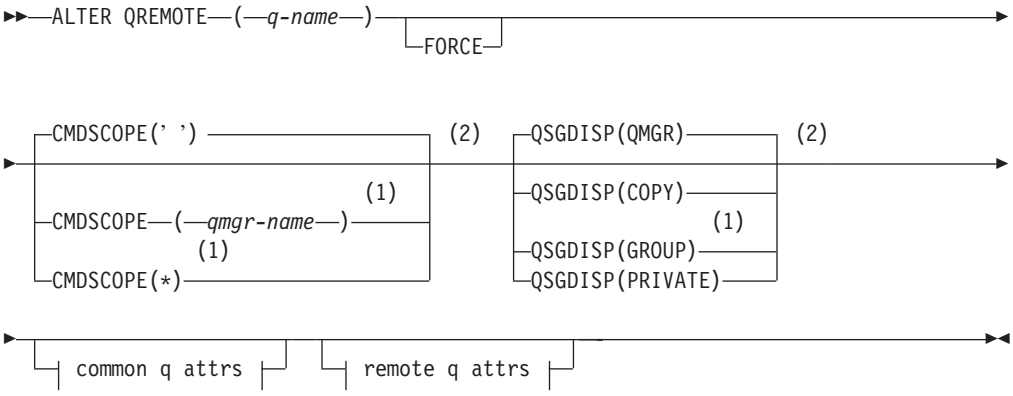
5 Valid only on i5/OS, UNIX systems, and Windows.

ALTER QREMOTE

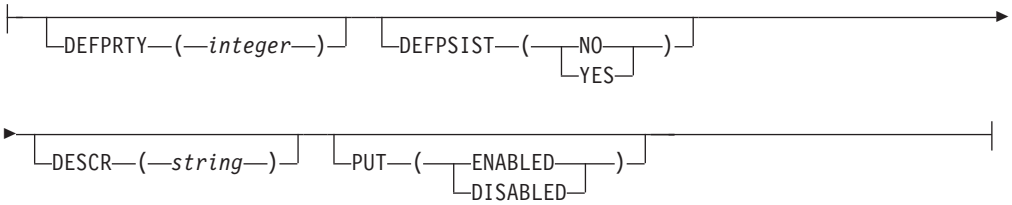
Use the MQSC command ALTER QREMOTE to alter the parameters of a local definition of a remote queue, a queue-manager alias, or a reply-to queue alias.

Synonym: ALT QR

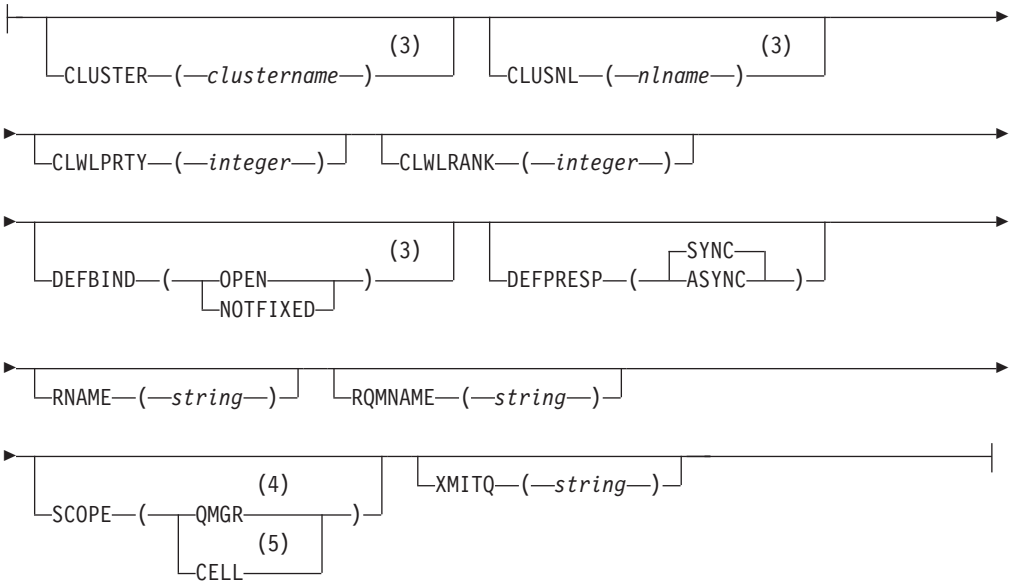
ALTER QREMOTE



Common q attrs:



Remote q attrs:



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

3 Valid only on AIX, HP-UX, z/OS, i5/OS, Solaris, and Windows.

4 Valid only on HP OpenVMS, i5/OS, UNIX systems, and Windows.

5 Valid only on HP OpenVMS, UNIX systems, and Windows.

Parameter descriptions for DEFINE QUEUE and ALTER QUEUE.

These parameter descriptions apply to the DEFINE QUEUE and 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** parameters apply only to the DEFINE QUEUE commands.

Table 2 shows the parameters that are relevant for each type of queue. There is a description of each parameter after the table.

Table 2. DEFINE and ALTER QUEUE parameters

	Local queue	Model queue	Alias queue	Remote queue
ACCTQ	X	X		
BOQNAME	X	X		
BOTHRESH	X	X		
CFSTRUCT	X	X		
CLUSNL	X		X	X
CLUSTER	X		X	X
CLWLPRTY	X		X	X
CLWLRANK	X		X	X
CLWLUSEQ	X			
CMDSCOPE	X	X	X	X
DEFBIND	X		X	X
DEFPRESP	X	X	X	X
DEFPRTY	X	X	X	X
DEFPSIST	X	X	X	X
DEFREADA	X	X	X	
DEFSOPT	X	X		
DEFTYPE	X	X		
DESCR	X	X	X	X
DISTL	X	X		
FORCE	X		X	X
GET	X	X	X	

Table 2. DEFINE and ALTER QUEUE parameters (continued)

	Local queue	Model queue	Alias queue	Remote queue
HARDENBO or NOHARDENBO	X	X		
INDXTYPE	X	X		
INITQ	X	X		
LIKE	X	X	X	X
MAXDEPTH	X	X		
MAXMSGL	X	X		
MONQ	X	X		
MSGDLVSQ	X	X		
NOREPLACE	X	X	X	X
NPMCLASS	X	X		
PROCESS	X	X		
PROPCTL	X	X	X	
PUT	X	X	X	X
q-name	X	X	X	X
QDEPTHHI	X	X		
QDEPTHLO	X	X		
QDPHIEV	X	X		
QDPLOEV	X	X		
QDPMAXEV	X	X		
QSGDISP	X	X	X	X
QSVCIEV	X	X		
QSVCINT	X	X		
REPLACE	X	X	X	X
RETINTVL	X	X		
RNAME				X
RQMNAME				X
SCOPE	X		X	X
SHARE or NOSHARE	X	X		
STATQ	X	X		
STGCLASS	X	X		
TARGET			X	
TARGQ			X	
TARGTYPE			X	
TRIGDATA	X	X		
TRIGDPTH	X	X		
TRIGGER or NOTRIGGER	X	X		
TRIGMPRI	X	X		
TRIGTYPE	X	X		

Table 2. DEFINE and ALTER QUEUE parameters (continued)

	Local queue	Model queue	Alias queue	Remote queue
USAGE	X	X		
XMITQ				X

(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 5.

ACCTQ
Specifies whether accounting data collection is to be enabled for the queue. On z/OS, the data collected is class 3 accounting data (thread-level and queue-level accounting). In order for accounting data to be collected for this queue, accounting data for this connection must also be enabled by either the ACCTQ queue manager attribute or the options field in the MQCNO structure on the MQCONNX call.

QMGR
The collection of accounting data is based on the setting of the ACCTQ parameter on the queue manager definition.

ON Accounting data collection is enabled for the queue unless the ACCTQ queue manager parameter has a value of NONE. On z/OS systems, you need to have switched on class 3 accounting by the START TRACE command.

OFF Accounting data collection is disabled for the queue.

BOQNAME(*string***)**
The excessive backout requeue name.

This parameter is supported only on local and model queues.

Use this parameter to set or change the value of the *BackoutRequeueQName* attribute of a local or model queue. Apart from allowing its value to be queried, the queue manager takes no action based on the value of this attribute. However, the Application Server Facilities (ASF) of WebSphere MQ classes for JMS use the attribute to determine where to transfer a message that has already been backed out the maximum number of times as specified by the *BackoutThreshold* attribute.

BOTHRESH(*integer***)**
The backout threshold.

This parameter is supported only on local and model queues.

Use this parameter to set or change the value of the *BackoutThreshold* attribute of a local or model queue. Apart from allowing its value to be queried, the queue manager takes no action based on the value of this attribute. However, the Application Server Facilities (ASF) of WebSphere MQ classes for JMS use the attribute to determine how many times to allow a message to be backed out before transferring the message to the queue specified by the *BackoutRequeueQName* attribute.

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.

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, HP OpenVMS, HP-UX, Linux, 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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

CLWLPRTY(*integer***)**
Specifies the priority of the queue for the purposes of cluster workload distribution. This parameter is valid only for local, remote, and alias queues. The value must be in the range zero through 9 where zero is the lowest priority and 9 is the highest. For more information about this attribute, see WebSphere MQ Queue Manager Clusters.

CLWLRANK(*integer***)**
Specifies the rank of the queue for the purposes of cluster workload distribution. This parameter is valid only for local, remote, and alias queues. The value must be in the range zero through 9 where zero is the lowest rank and 9 is the highest. For more information about this attribute, see WebSphere MQ Queue Manager Clusters.

CLWLUSEQ
Specifies the behavior of an MQPUT operation when the target queue has a local instance and at least one remote cluster instance (except where the MQPUT originates from a cluster channel). This parameter is valid only for local queues.

QMGR
The behavior will be as specified by the CLWLUSEQ parameter of the queue manager definition.

ANY The queue manager is to treat the local queue as another instance of the cluster queue for the purposes of workload distribution.

LOCAL
The local queue is the only target of the MQPUT operation.

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 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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

DEFPRESP
Specifies the behavior to be used by applications when the put response type, within the MQPMO options, is set to MQPMO_RESPONSE_AS_Q_DEF.

SYNC Ensures that the put operations to the queue specifying MQPMO_RESPONSE_AS_Q_DEF are issued as if MQPMO_SYNC_RESPONSE had been specified instead. This is the default supplied with WebSphere MQ, but your installation might have changed it.

ASYNC
Ensures that the put operations to the queue specifying MQPMO_RESPONSE_AS_Q_DEF are issued as if MQPMO_ASYNC_RESPONSE had been specified instead.

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.)

DEFPSIST
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**.

DEFREADA
Specifies the default read ahead behavior for non-persistent messages delivered to the client. Enabling read ahead can improve the performance of client applications consuming non-persistent messages.

NO Non-persistent messages are not read ahead unless the client application is configured to request read ahead.

YES Non-persistent messages are sent to the client before an application requests them. Non-persistent messages can be lost if the client ends abnormally or if the client does not consume all the messages it is sent.

DISABLED
Read ahead of non-persistent messages in not enabled for this queue. Messages are not sent ahead to the client regardless of whether read ahead is requested by the client application.

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 DEFPSIST parameter of YES.

If you specify this option, do not specify INDXTYPE(MSGTOKEN).

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, HP OpenVMS, HP-UX, Linux, i5/OS, 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
- 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).

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.

Note: This parameter affects only WebSphere MQ for z/OS. It can be set on other platforms but has no effect.

INDXTYPE
The type of index maintained by the queue manager to expedite **MQGET** operations on the queue. For shared queues, the type of index determines the type of **MQGET** operations that can be used.

This parameter is supported only on local and model queues.

Messages can be retrieved using a selection criterion only if an appropriate index type is maintained, as the following table shows:

Retrieval selection criterion	Index type required	
	Shared queue	Other queue
None (sequential retrieval)	Any	Any
Message identifier	MSGID or NONE	Any
Correlation identifier	CORRELID	Any
Message and correlation identifiers	MSGID or CORRELID	Any
Group identifier	GROUPID	Any
Grouping	GROUPID	GROUPID
Message token	Not allowed	MSGTOKEN

where the value of INDXTYPE parameter has the following values:

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.

Note:

1. You cannot set INDXTYPE to GROUPID if the queue is a transmission queue.
2. 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)

For queues that are not shared and do not use grouping or message tokens, the index type does not restrict the type of retrieval selection. However, the index is used to expedite GET operations on the queue, so choose the type that corresponds to the most common retrieval selection.

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:

Queue type		**NON-SHARED**			**SHARED**	
Queue state		**Uncommitted activity**	**No uncommitted activity, messages present**	**No uncommitted activity, and empty**	**Open or messages present**	**Not open, and empty**
Change INDXTYPE from:	**To:**	**Change allowed?**				
NONE	MSGID	No	Yes	Yes	No	Yes
NONE	CORRELID	No	Yes	Yes	No	Yes
NONE	MSGTOKEN	No	No	Yes	-	-

Queue type		NON-SHARED			SHARED	
NONE	GROUPID	No	No	Yes	No	Yes
MSGID	NONE	No	Yes	Yes	No	Yes
MSGID	CORRELID	No	Yes	Yes	No	Yes
MSGID	MSGTOKEN	No	No	Yes	-	-
MSGID	GROUPID	No	No	Yes	No	Yes
CORRELID	NONE	No	Yes	Yes	No	Yes
CORRELID	MSGID	No	Yes	Yes	No	Yes
CORRELID	MSGTOKEN	No	No	Yes	-	-
CORRELID	GROUPID	No	No	Yes	No	Yes
MSGTOKEN	NONE	No	Yes	Yes	-	-
MSGTOKEN	MSGID	No	Yes	Yes	-	-
MSGTOKEN	CORRELID	No	Yes	Yes	-	-
MSGTOKEN	GROUPID	No	No	Yes	-	-
GROUPID	NONE	No	No	Yes	No	Yes
GROUPID	MSGID	No	No	Yes	No	Yes
GROUPID	CORRELID	No	No	Yes	No	Yes
GROUPID	MSGTOKEN	No	No	Yes	-	-

This parameter is supported only on z/OS. On other platforms, all queues are automatically indexed.

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 5.

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:

SYSTEM.DEFAULT.ALIAS.QUEUE	Alias queue
SYSTEM.DEFAULT.LOCAL.QUEUE	Local queue
SYSTEM.DEFAULT.MODEL.QUEUE	Model queue
SYSTEM.DEFAULT.REMOTE.QUEUE	Remote queue

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 z/OS, the queue manager searches for an object with the name and queue type you specify with a disposition of QMGR, COPY, or SHARED. The disposition of the LIKE object is not copied to the object you are defining.

Note:

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, HP OpenVMS, HP-UX, Linux, i5/OS, 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, HP OpenVMS, HP-UX, Linux, i5/OS, 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 (104 857 600 bytes).

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

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.

MONQ
Controls the collection of online monitoring data for queues.

This parameter is supported only on local and model queues.

QMGR
Collect monitoring data according to the setting of the queue manager parameter MONQ. This is the default value.

OFF Online monitoring data collection is turned off for this queue.

LOW If the value of the queue manager's MONQ parameter is not NONE, online monitoring data collection is turned on for this queue.

MEDIUM
If the value of the queue manager's MONQ parameter is not NONE, online monitoring data collection is turned on for this queue.

HIGH If the value of the queue manager's MONQ parameter is not NONE, online monitoring data collection is turned on for this queue.

Note that there is no distinction between the values LOW, MEDIUM and HIGH. These values all turn data collection on, but do not affect the rate of collection.

When this parameter is used in an ALTER queue command, the change is effective only when the queue is next opened.

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.

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.

NPMCLASS
The level of reliability to be assigned to non-persistent messages that are put to the queue:

NORMAL
Non-persistent messages are lost after a failure, or queue manager shutdown. These messages are discarded on a queue manager restart.

HIGH The queue manager attempts to retain non-persistent messages on this queue over a queue manager restart.

You cannot set this parameter on z/OS.

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 5.

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 definition contains the name of the channel to be started. This parameter is optional for transmission queues on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS; if you do not specify it, the channel name is taken from the value specified for the TRIGDATA parameter.

PROPCTL
Property control attribute.

This parameter is applicable to Local, Alias and Model queues.

This parameter is optional.

Permissable values are:

ALL All properties of the message are included with the message when it is sent to the remote queue manager. The properties, except those in the message descriptor (or extension), are placed in one or more MQRFH2 headers in the message data.

COMPAT
If the message contains a property with a prefix of mcd., jms., usr. or mqext., all message properties are delivered to the application in an MQRFH2 header. Otherwise all properties of the message, except those contained in the message descriptor (or extension), are discarded and are no longer accessible to the application.

This is the default value; it allows applications which expect JMS related properties to be in an MQRFH2 header in the message data to continue to work unmodified.

FORCE
Properties are always returned in the message data in an MQRFH2 header regardless of whether the application specifies a message handle.

A valid message handle supplied in the MsgHandle field of the MQGMO structure on the MQGET call is ignored. Properties of the message are not accessible via the message handle.

NONE
All properties of the message, except those in the message descriptor (or extension), are removed from the message before the message is sent to the remote queue manager.

Note, that for a transmission queue, which is a local queue with USAGE(XMITQ), the PROPCTL attribute of the queue is irrelevant and it is the PROPCTL attribute on the corresponding channel object that controls the message property behavior.

PUT Whether messages can be put on the queue.

ENABLED
Messages can be added to the queue (by suitably authorized applications).

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 Monitoring WebSphere MQ 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 Monitoring WebSphere MQ 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 Monitoring WebSphere MQ 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 Monitoring WebSphere MQ 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 Monitoring WebSphere MQ 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.

QSGDISP	ALTER	DEFINE
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.	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.

QSGDISP	ALTER	DEFINE

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), 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 attempt to refresh local copies on page set zero:
	`DEFINE QUEUE(q-name)`
	`REPLACE QSGDISP(COPY)`
	
	The ALTER for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.
		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 attempt to make or refresh local copies on page set zero:
		`DEFINE QUEUE(q-name)`
		`REPLACE QSGDISP(COPY)`
		
		The DEFINE for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.

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.	Not permitted.
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. For local queues, messages are stored on the page sets of each queue manager and are available only through that queue manager.
SHARED	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.	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: • CFSTRUCT is nonblank • INDXTYPE is not MSGTOKEN • The queue is not one of the following: – SYSTEM.CHANNEL.INITQ – SYSTEM.COMMAND.INPUT 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.

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 Monitoring WebSphere MQ 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 QSVCIEV parameter.

The value is in units of milliseconds, and must bein 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.

This parameter applies only to the DEFINE Queue command.

(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 HP OpenVMS, UNIX systems, 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.

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***)**
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.

  In a queue manager cluster, this definition applies only to the queue manager that made it. To advertise the alias to the whole cluster, add the CLUSTER attribute to the remote queue definition.
- 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 5).

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 5).

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 has been configured (note that the DCE name service is no longer supported).

This parameter is valid only on HP OpenVMS, 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

STATQ

Whether statistics data collection is enabled:

QMGR
Statistics data collection is based on the setting of the queue manager's STATQ parameter.

ON If the value of the queue manager's STATQ parameter is not NONE, statistics data collection for the queue is enabled.

OFF Statistics data collection for the queue is disabled.

When this parameter is used in an ALTER queue command, the change is effective only only for connections to the queue manager made after the change to the parameter.

This parameter is valid only on i5/OS, UNIX systems, and Windows.

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.

If you specify QSGDISP(SHARED) or DEFTYPE(SHAREDYN), this parameter is ignored.

TARGET(*string***)**
The name of the queue or topic object being aliased. (See "Rules for naming WebSphere MQ objects" on page 5.) The object can be a queue or a topic as defined by TARGTYPE. The maximum length is 48 characters.

This parameter is supported only on alias queues.

This object needs to be defined only when an application process opens the alias queue.

This is a synonym of the parameter TARGQ which has been retained for compatibility purposes. If you specify TARGET, you cannot also specify TARGQ.

TARGTYPE(*string***)**
The type of object to which the alias resolves.

QUEUE
The alias resolves to a queue.

TOPIC
The alias resolves to a topic.

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 NonStop Kernel, HP OpenVMS, HP-UX, Linux, i5/OS, 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 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 414 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.

ALTER SECURITY

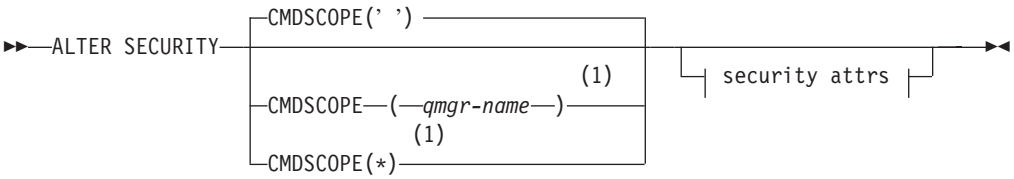
Use the MQSC command ALTER SECURITY to define system-wide security options.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

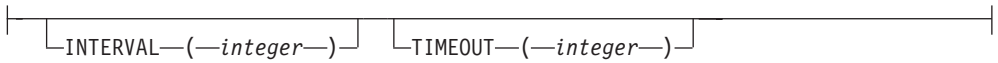
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: ALT SEC

ALTER SECURITY



Security attrs:



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for ALTER SECURITY

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 accociated 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.

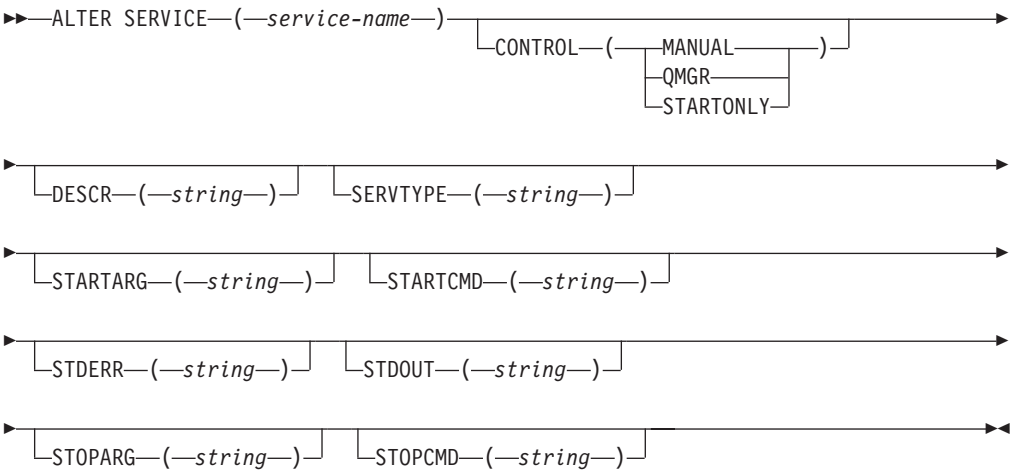
ALTER SERVICE

Use the MQSC command ALTER SERVICE to alter the parameters of an existing WebSphere MQ service definition.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	

Synonym:

ALTER SERVICE



Parameter descriptions for ALTER SERVICE

For a description of the parameters see "DEFINE SERVICE" on page 265.

ALTER STGCLASS

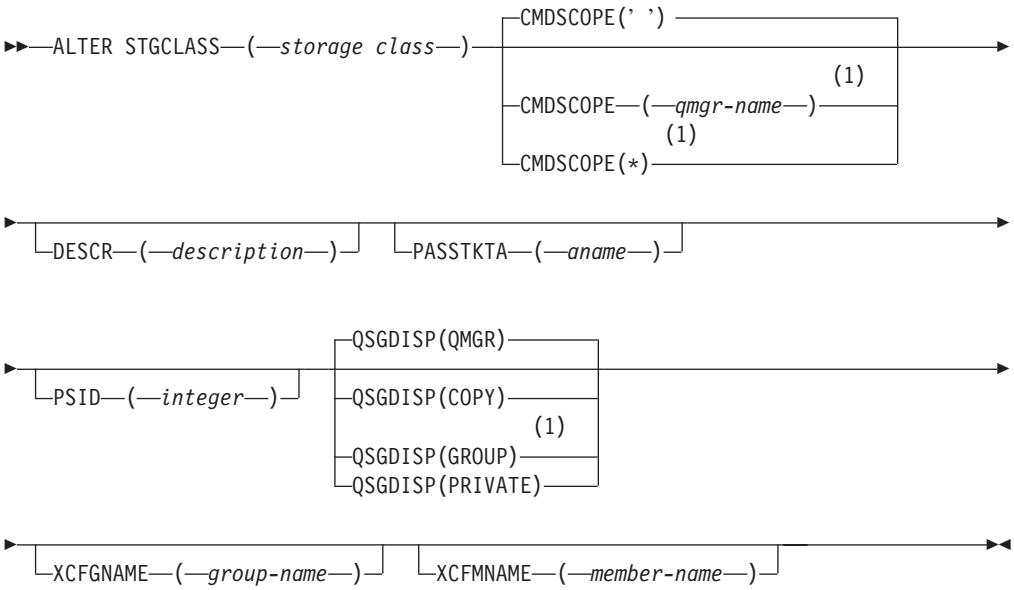
Use the MQSC command ALTER STGCLASS to alter the characteristics of a storage class.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: ALT STC

ALTER STGCLASS



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for ALTER STGCLASS

For a description of the parameters see "DEFINE STGCLASS" on page 268.

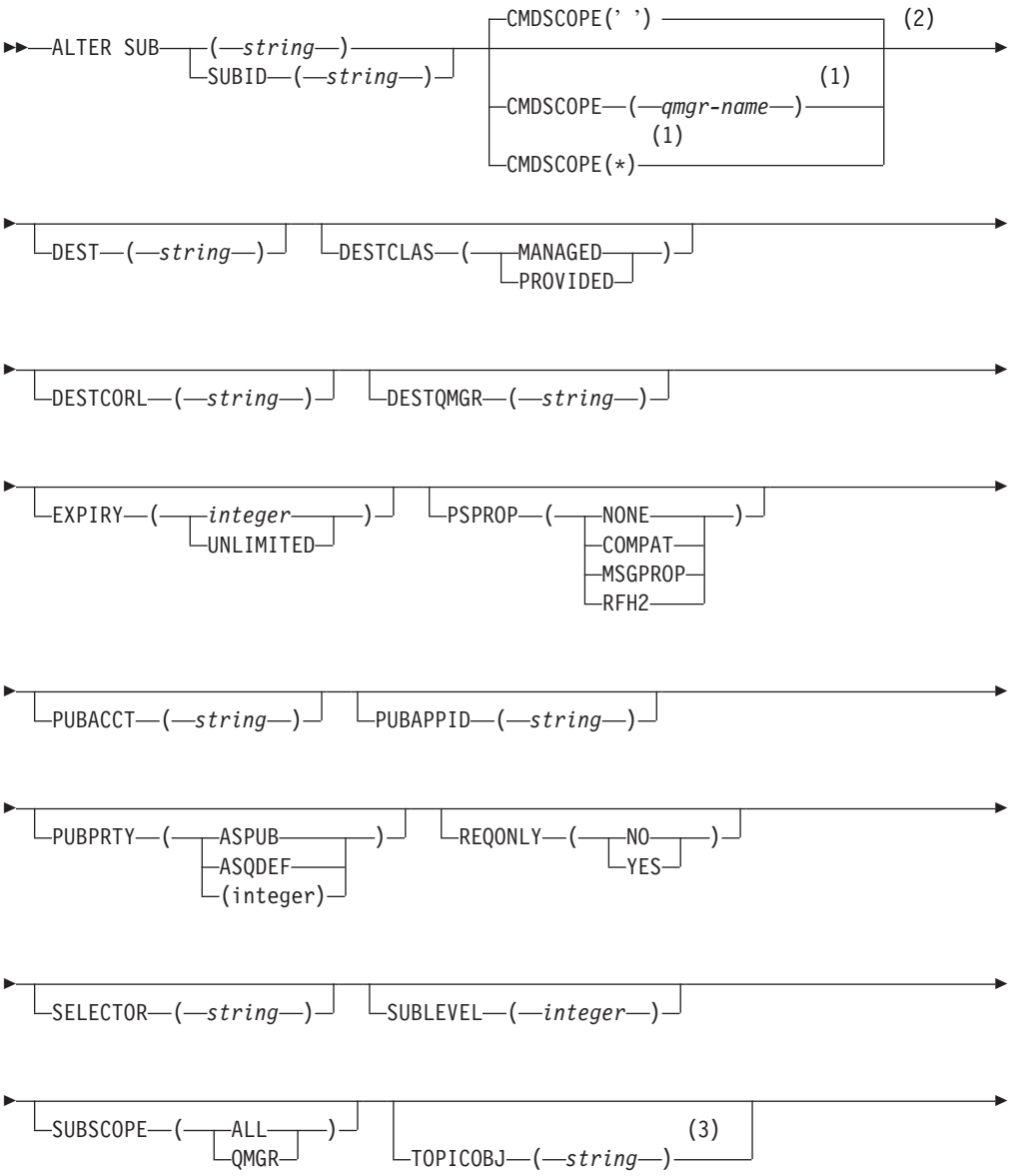
ALTER SUB

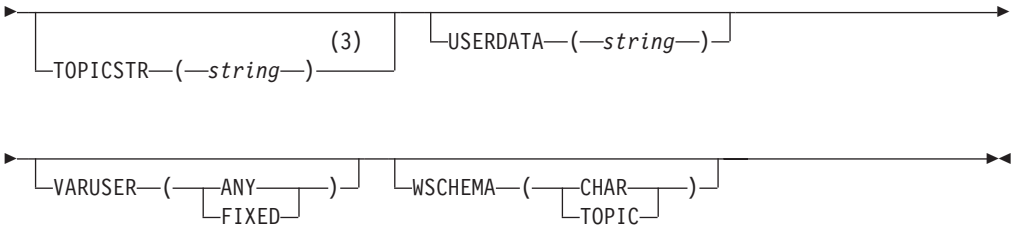
Use the MQSC command ALTER SUB to alter the characteristics of an existing subscription.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

Synonym: ALT SUB

ALTER SUB





Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

3 At least one of TOPICSTR and TOPICOBJ must be present on DEFINE.

Parameter descriptions for ALTER SUB

Describes the ALTER SUB (ALT SUB) parameters.

(string)
This is an optional parameter that specifies the subscription name of the subscription to be altered. If you do not specify this, you must specify a value in the SUBID parameter.

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

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

qmgr-name
The command is processed 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 processed 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.

You cannot use CMDSCOPE as a filter keyword.

DEST(*string***)**
The destination for messages published to this subscription; the destination is the name of a queue.

DESTCLAS
System managed destination.

PROVIDED
The destination is a queue.

MANAGED
The destination is managed.

DESTCORL(*string***)**
The CorrelId used for messages published to this subscription.

DESTQMGR(*string***)**
The destination queue manager for messages published to this subscription.

EXPIRY
The time to expiry from the creation date and time.

(*integer***)**
The time to expiry, in tenths of a second, from the creation date and time.

UNLIMITED
There is no expiry time. This is the default option supplied with the product.

PSPROP
The manner in which publish subscribe related message properties are added to messages published to this subscription.

NONE
Do not add publish subscribe properties to the message. This is the default option supplied with the product.

COMPAT
Publish subscribe properties are added as PCF attributes.

MSGPROP
Publish subscribe properties are added within an MQRFH version 1 header.

RFH2 Publish subscribe properties are added within an MQRFH version 2 header.

PUBACCT(*string***)**
Accounting token passed by the subscriber, for propagation into messages published to this subscription in the *AccountingToken* field of the MQMD.

PUBAPPID(*string***)**
Identity data passed by the subscriber for propagation into messages sent to this subscription in the *ApplId* field of the MQMD.

PUBPRTY
The priority of the message published to this subscription.

ASPUB
Priority of the message published to this subscription is taken from that supplied in the published message.

ASQDEF
Priority of the message published to this subscription is taken from the default priority of the queue defined as a destination.

0-9 An integer providing an explicit priority for messages published to this subscription.

REQONLY
Indicates whether the subscriber polls for updates using the MQSUBRQ API call, or whether all publications are delivered to this subscription.

NO All publications on the topic are delivered to this subscription.

YES Publications are only delivered to this subscription in response to an MQSUBRQ API call.

SELECTOR(*string***)**
A selector applied to messages published on the named topic, to select whether they are eligible for this subscription.

SUBID(*string***)**
The internal, unique key that identifies a subscription.

SUBLEVEL(*integer***)**
The level within the subscription hierarchy at which this subscription is made. The range is zero through 9.

SUBSCOPE
Determines whether this subscription should be forwarded to other brokers so that the subscriber receives messages published at those other brokers.

ALL The subscription is forwarded to all queue managers directly connected through a publish subscribe collective or hierarchy.

QMGR
The subscription forwards messages published on the topic only within this queue manager.

TOPICOBJ(*string***)**
The name of a topic object used by this subscription.

TOPICSTR(*string***)**
Specifies a fully qualified topic name, or a topic set using wildcard characters for the subscription.

USERDATA(*string***)**
Specifies the user data associated with the subscription. The string is a variable length value that can be retrieved by the application on an MQSUB API call, and passed in a message sent to this subscription as a message property.

VARUSER
Specifies whether a user, other than the subscription creator, can connect to and take over ownership of the subscription.

ANY Any user can connect to and takeover ownership of the subscription.

FIXED
Takeover by another **USERID** is not permitted.

WSCHEMA
The schema to be used when interpreting any wildcard characters in the topic string.

CHAR
Wildcard characters represent portions of strings.

TOPIC
Wildcard characters represent portions of the topic hierarchy. This is the default value supplied with the product.

Usage notes for ALTER SUB

1. The following are valid forms of the command:

   ```
   ALT SUB(xyz)
   ALT SUB SUBID(123)
   ALT SUB(xyz) SUBID(123)
   ```

2. Although permitted on the command, you can not alter the following fields using DEF SUB (REPLACE) or ALTER SUB:
   - SELECTOR
   - TOPICOBJ
   - TOPICSTR
   - WSCHEMA
3. At the time the ALT SUB command processes, no check is performed that the named DEST or DESTQMGR exists. These names are used at publishing time as the *ObjectName* and *ObjectQMgrName* for an MQOPEN call. These names are resolved according to the WebSphere MQ name resolution rules.

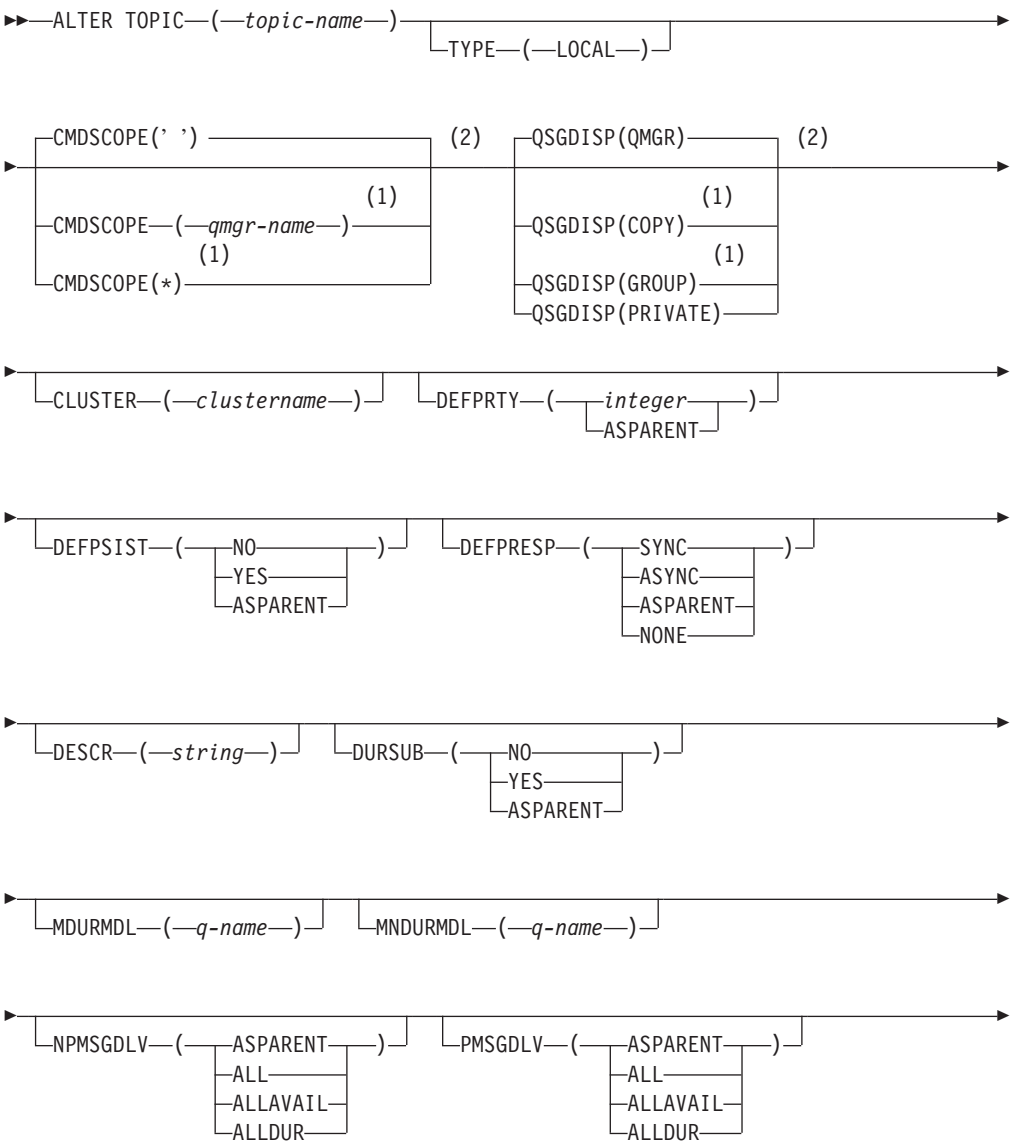
ALTER TOPIC

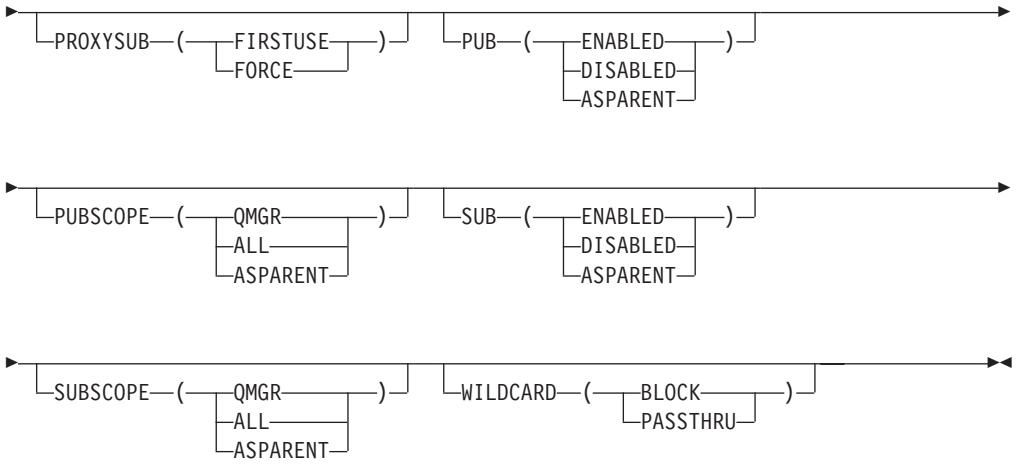
Use ALTER TOPIC to alter the parameters of an existing WebSphere MQ topic object.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

Synonym: ALT TOPIC

ALTER TOPIC





Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for ALTER TOPIC

For a description of the parameters see "DEFINE TOPIC" on page 278.

ALTER TRACE

Use the MQSC command ALTER TRACE to change the trace events being traced for a particular active queue manager trace. ALTER TRACE stops the specified trace, and restarts it with the altered parameters.

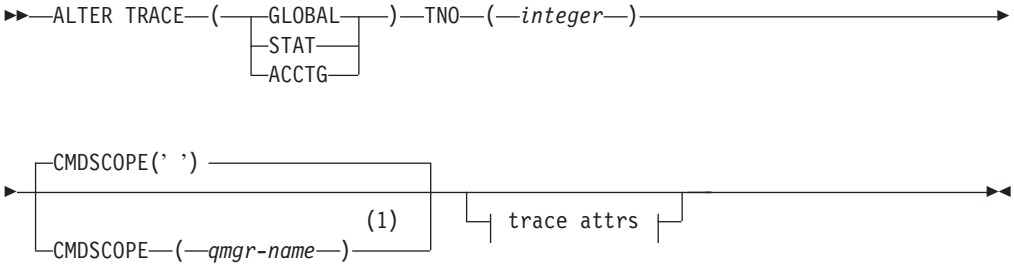
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

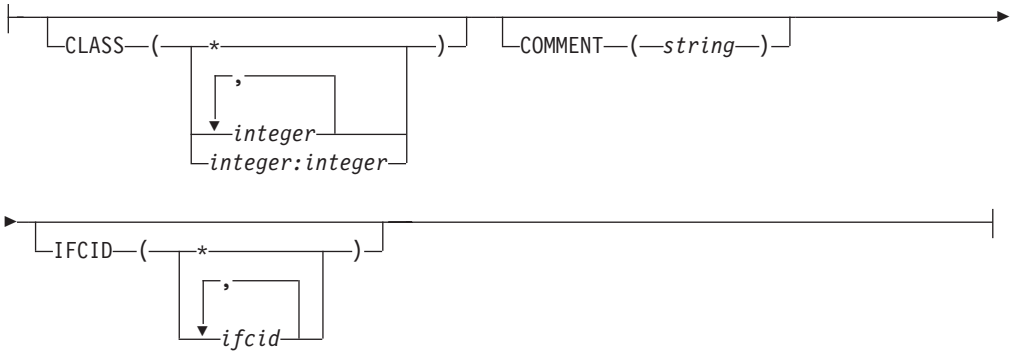
Note: Channel initiator traces cannot be altered.

Synonym: ALT TRACE

ALTER TRACE



Trace attrs:



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for ALTER TRACE

Specify one of the following trace types:

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 (1 through 32). You can specify only one trace number.

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 new trace class. See "START TRACE" on page 573 for a list of allowed classes. A range of classes can be specified as *m:n* (for example, CLASS(01:03)). CLASS(*) activates all 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***)**
Reserved for IBM Service.

ARCHIVE LOG

Use the MQSC command 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).

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

In detail, ARCHIVE LOG:

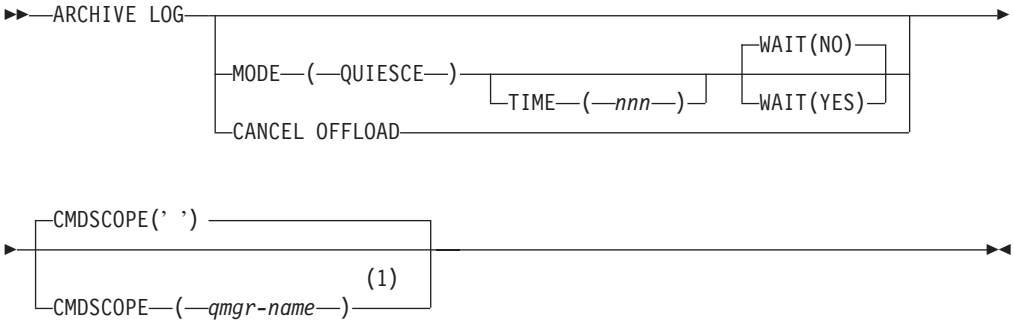
1. Truncates the current active log data sets.
2. Continues logging, switching to the next active log data set.
3. Starts a task to off-load the data sets.
4. Archives previous active log data sets not yet archived.

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



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for ARCHIVE LOG

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.

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(*nnn*)

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

nnn 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 for ARCHIVE LOG

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 LOG 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.

BACKUP CFSTRUCT

Use the MQSC command BACKUP CFSTRUCT to initiate a CF application structure backup.

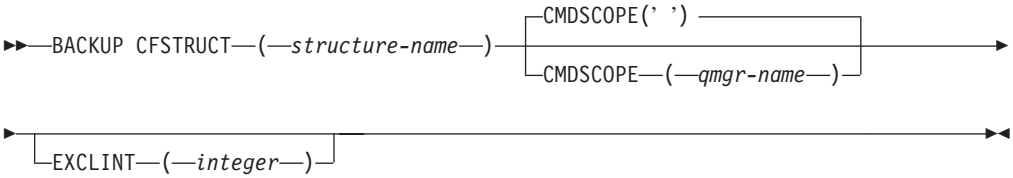
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note: This command is valid only on z/OS when the queue manager is a member of a queue-sharing group.

Synonym: None

BACKUP CFSTRUCT



Keyword and parameter descriptions for BACKUP CFSTRUCT

The keywords and parameters for BACKUP CFSTRUCT are described.

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

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 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 for BACKUP CFSTRUCT

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.

CLEAR QLOCAL

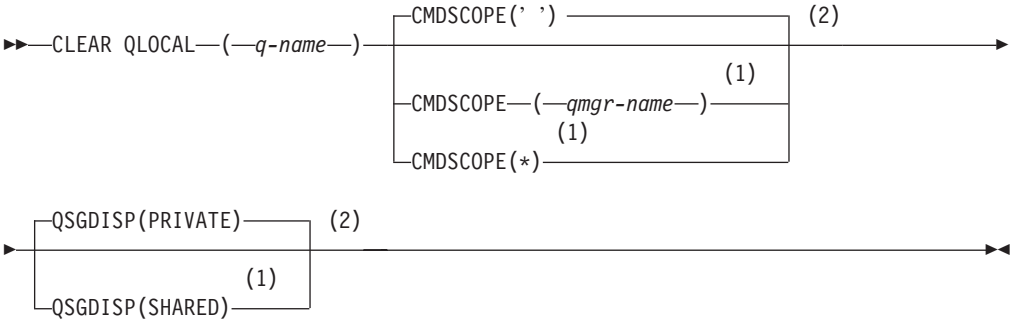
Use the MQSC command CLEAR QLOCAL to clear the messages from a local queue.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: CLEAR QL

CLEAR QLOCAL



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for CLEAR QLOCAL

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.

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 for CLEAR QLOCAL

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.

CLEAR TOPICSTR

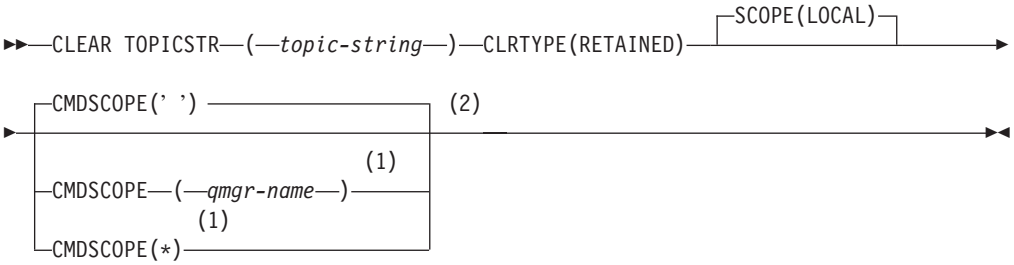
Use the MQSC command CLEAR TOPICSTR to clear the retained message which is stored for the specified topic string.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: None.

CLEAR TOPIC



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for CLEAR TOPICSTR

You must specify which topic string you want to remove the retained publication from.

(topic-string)
The topic string to be cleared. This string can represent several topics to be cleared by using wildcards as shown in the table below.

Table 3.

Special Character	Behaviour
#	Wildcard, multiple topic level
+	Wildcard, single topic level
Note: the '+' and '#' are not treated as wildcards if they are mixed in with other characters (including themselves) within a topic level. In the string below the '#' and '+' characters are treated as ordinary characters. `level0/level1/#+/level3/level#`	

To illustrate this, the following example is used.

Clearing the following topic;

```
/a/b/#/z
```

will clear the following topics;

```
/a/b/z
/a/b/c/z
/a/b/c/y/z
```

CLRTYPE
This is a mandatory parameter.

The value must be:

RETAINED
Remove the retained publication from the specified topic string.

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.

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.

SCOPE
The scope of the deletion of retained messages.

The value can be:

LOCAL
The retained message is removed from the specified topic string at the local queue manager only. This is the default value.

Usage notes for CLEAR TOPICSTR

1. If the topic string specified has no retained message the command will complete successfully. You can find out whether a topic string has a retained message by using the DISPLAY TPSTATUS command. The RETAINED field shows whether there is a retained message.
2. 2. The topic-string input parameter on this command must match the topic you wish to act upon. You are advised to keep the character strings in your topic strings as characters that can be used from location issuing the command. If you issue commands using MQSC, you will have fewer characters available to you than if you are using an application submitting PCF messages, such as the WebSphere MQ Explorer.

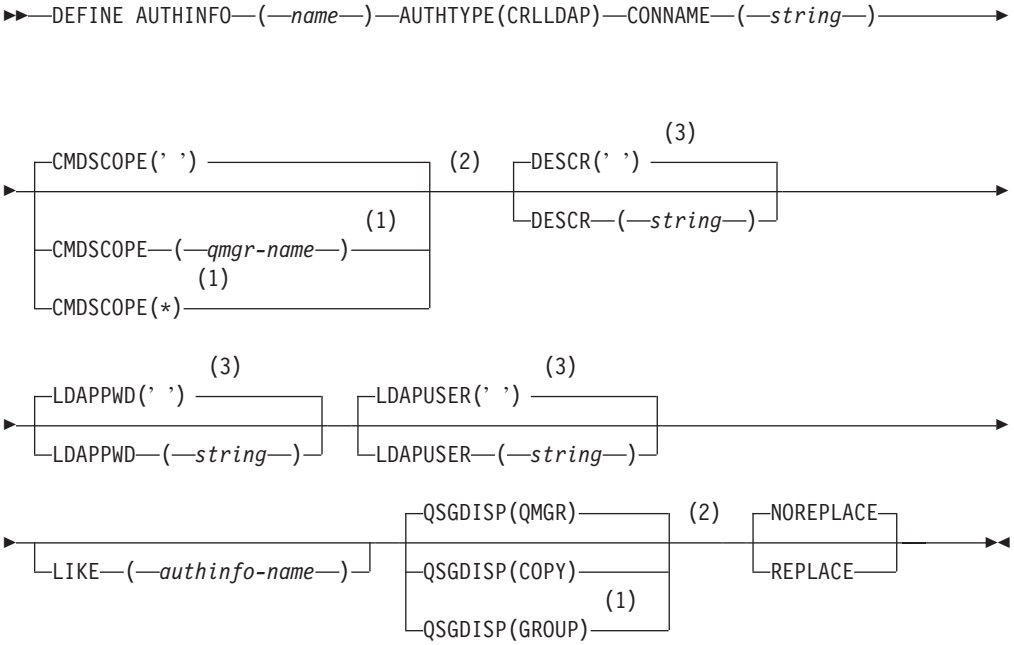
DEFINE AUTHINFO

Use the MQSC command 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 i5/OS where these are defined by the Digital Certificate Manager for each Certification Authority.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

Synonym: DEF AUTHINFO

DEFINE 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 for DEFINE AUTHINFO

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.

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 5.

AUTHTYPE

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

This parameter is valid only on the DEFINE AUTHINFO command.

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 hostname, IPv4 dotted decimal address, or IPv6 hexadecimal notation of the host on which the LDAP server is running, with an optional port number. This keyword is required.

If you specify the connection name as an IPv6 address, only systems with an IPv6 stack are able to resolve this address. If the AUTHINFO object is part of the queue manager's CRL namelist, ensure that any clients using the client channel table generated by the queue manager are capable of resolving the connection name.

On z/OS, if a CONNAME is to resolve to an IPv6 network address, a level of z/OS that supports IPv6 for connection to an LDAP server is required.

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

```
conname('hostname(nnn)')
```

where *nnn* is the port number. If *nnn* is not provided, the default port number 389 is used.

The maximum length for the field is 264 characters on i5/OS, 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 315).

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.

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. On z/OS, the LDAPPWD used for accessing the LDAP server may not be the one defined in the AUTHINFO object. If more than one AUTHINFO object is placed in the namelist referred to by the QMGR parameter SSLCRLNL, the LDAPPWD in the first AUTHINFO object is used for accessing all LDAP Servers.

LDAPUSER(*string***)**
The Distinguished Name of the user who is accessing the LDAP server. (See the SSLPEER parameter for more information about distinguished names.)

The maximum size for the user name is 1024 characters on i5/OS, UNIX systems, and Windows, and 256 characters on z/OS.

On z/OS, the LDAPUSER used for accessing the LDAP Server may not be the one defined in the AUTHINFO object. If more than one AUTHINFO object is placed in the namelist referred to by the QMGR parameter SSLCRLNL, the LDAPUSER in the first AUTHINFO object is used for accessing all LDAP Servers.

On i5/OS, UNIX systems, and Windows, the maximum accepted line length is defined to be BUFSIZ, which can be found in stdio.h.

LIKE(*authinfo-name***)**
The name of an authentication information object, whose parameters will be used to model this definition. This parameter is valid only on the DEFINE AUTHINFO 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 an object of this type. This is equivalent to specifying LIKE(SYSTEM.DEFAULT.AUTHINFO.CRLLDAP).

This default authentication information object definition can be altered by the installation to the default values required.

On z/OS, the queue manager searches for an object with the name you specify and a disposition of QMGR or COPY. The disposition of the LIKE object is not copied to the object you are defining.

This parameter is valid only on the DEFINE AUTHINFO command.

Note:

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.

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).

QSGDISP	ALTER	DEFINE
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.	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.

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 attempt to refresh local copies on page set zero:
	`DEFINE AUTHINFO(name)`
	`REPLACE QSGDISP(COPY)`
	
	The ALTER for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.
		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 AUTHINFO(name)`
		`REPLACE QSGDISP(COPY)`
		
		The DEFINE for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.

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.	Not permitted.
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.

This parameter is valid only on the DEFINE AUTHINFO command.

Usage Notes for DEFINE AUTHINFO

1. On i5/OS, authentication information objects are only used for channels of type CLNTCONN through use of the AMQCLCHL.TAB. Certificates are defined by Digital Certificate Manager for each Certification Authority, and are verified against the LDAP servers.

DEFINE BUFFPOOL

Use the MQSC command DEFINE BUFFPOOL to define a buffer pool that is used for holding messages in main storage.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					1

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

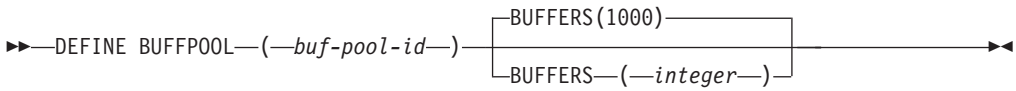
Specify DEFINE BUFFPOOL commands in a data set identified by the CSQINP1 DD concatenation in the queue manager started task procedure. These definitions are not retained so you must define them at each queue manager start using a data set referenced from CSQINP1.

Note:

1. You can use the DISPLAY USAGE TYPE(PAGESET) command to display buffer pool information (see "DISPLAY USAGE" on page 507).
2. You can use the ALTER BUFFPOOL command to dynamically add or remove buffers in a predefined buffer pool (see "ALTER BUFFPOOL" on page 13).

Synonym: DEF BP

DEFINE BUFFPOOL



Parameter descriptions for DEFINE BUFFPOOL

Parameter descriptions for the DEFINE BUFFPOOL command

If more than one DEFINE BUFFPOOL command is issued for the same buffer pool, only the last one is processed.

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

This is an integer in the range zero through 15.

BUFFERS(*integer***)**

This is a required parameter and is the number of 4096 byte buffers to be used in this buffer pool. The minimum value is 100. The default number of buffers is 1000. The maximum value 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.

Attention: The queue manager records the current buffer pool sizes in checkpoint log records. These are automatically restored when a queue manager is subsequently restarted. This occurs after processing of the CSQINP1 data set, therefore if you have used ALTER BUFFPOOL since the buffer pool was last

defined, it appears that any DEFINE BUFFPOOL command in CSQINP1 has been ignored at restart.

DEFINE CFSTRUCT

Use the MQSC command DEFINE CFSTRUCT to define queue manager CF level capability, and backup and recovery parameters for a Coupling Facility application structure.

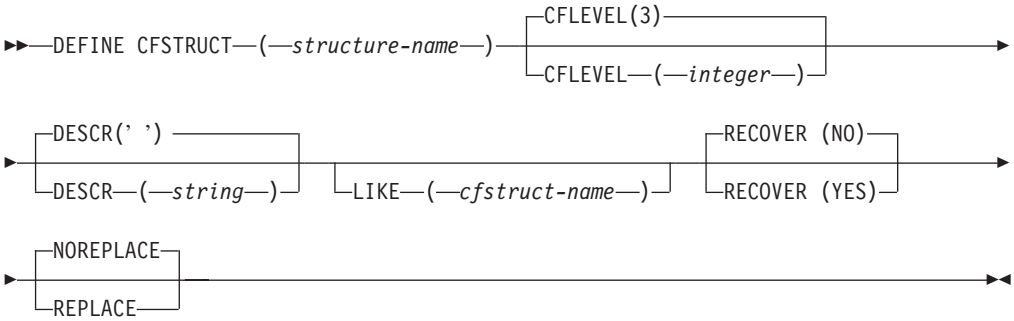
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

This command is valid only on z/OS when the queue manager is a member of a queue-sharing group.

Synonym: DEF CFSTRUCT

DEFINE CFSTRUCT



Keyword and parameter descriptions for DEFINE CFSTRUCT

(structure-name)
Name of the Coupling Facility application structure whose queue manager CF level capability and 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 `NY03PRODUCT7`. 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 or greater.

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.

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 or greater - this is to ensure that there are no latent command level 520 connections to queues referencing the structure.

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.

This is the default CFLEVEL for queue managers at command level 600.

4

This CFLEVEL supports all the CFLEVEL(3) functions. CFLEVEL(4) allows queues defined with CF structures at this level to have messages with a length greater than 63 KB.

Only a queue manager with a command level of 600 can connect to a CF structure at CFLEVEL(4).

You can only increase the value of CFLEVEL to 4 if all the queue managers in the queue-sharing group are at command level 600 or greater.

You can only decrease the value of CFLEVEL from 4 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.

This parameter is valid only on the DEFINE CFSTRUCT command.

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 or higher.

You can only change RECOVER(NO) to RECOVER(YES) if all the queue managers in the queue-sharing group are at command level 530 or greater ; 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.

NOREPLACE

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

This parameter is valid only on the DEFINE CFSTRUCT command.

Usage notes for DEFINE CFSTRUCT

1. This command cannot specify the CF administration structure (CSQ_ADMIN).
2. 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.
3. Only CF structures with RECOVER(YES) defined can be backed up and recovered.

DEFINE CHANNEL

Use the MQSC command DEFINE CHANNEL to define a new channel, and set its parameters.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

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

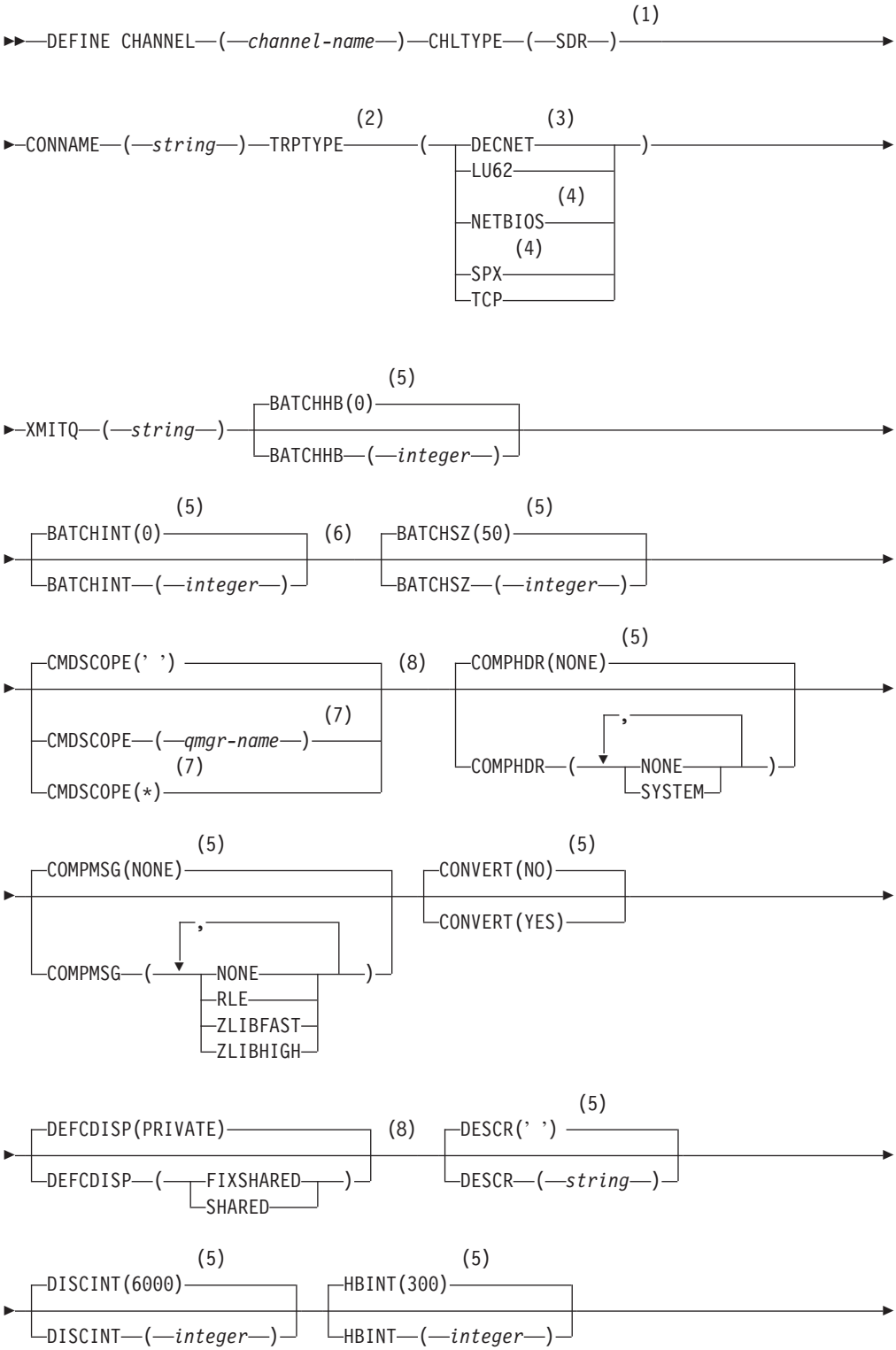
Synonym: DEF CHL

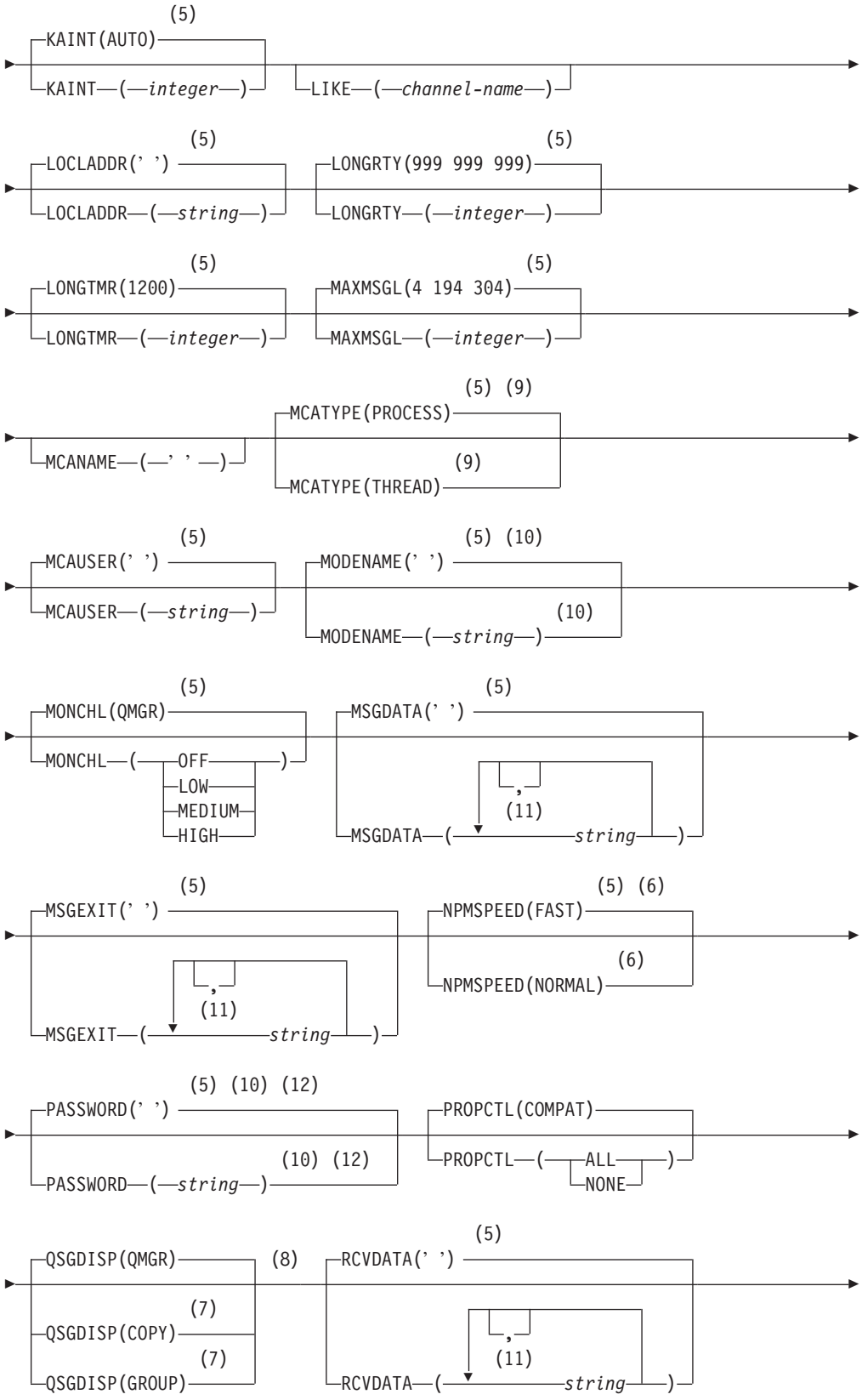
There is a separate syntax diagram for each type of channel:

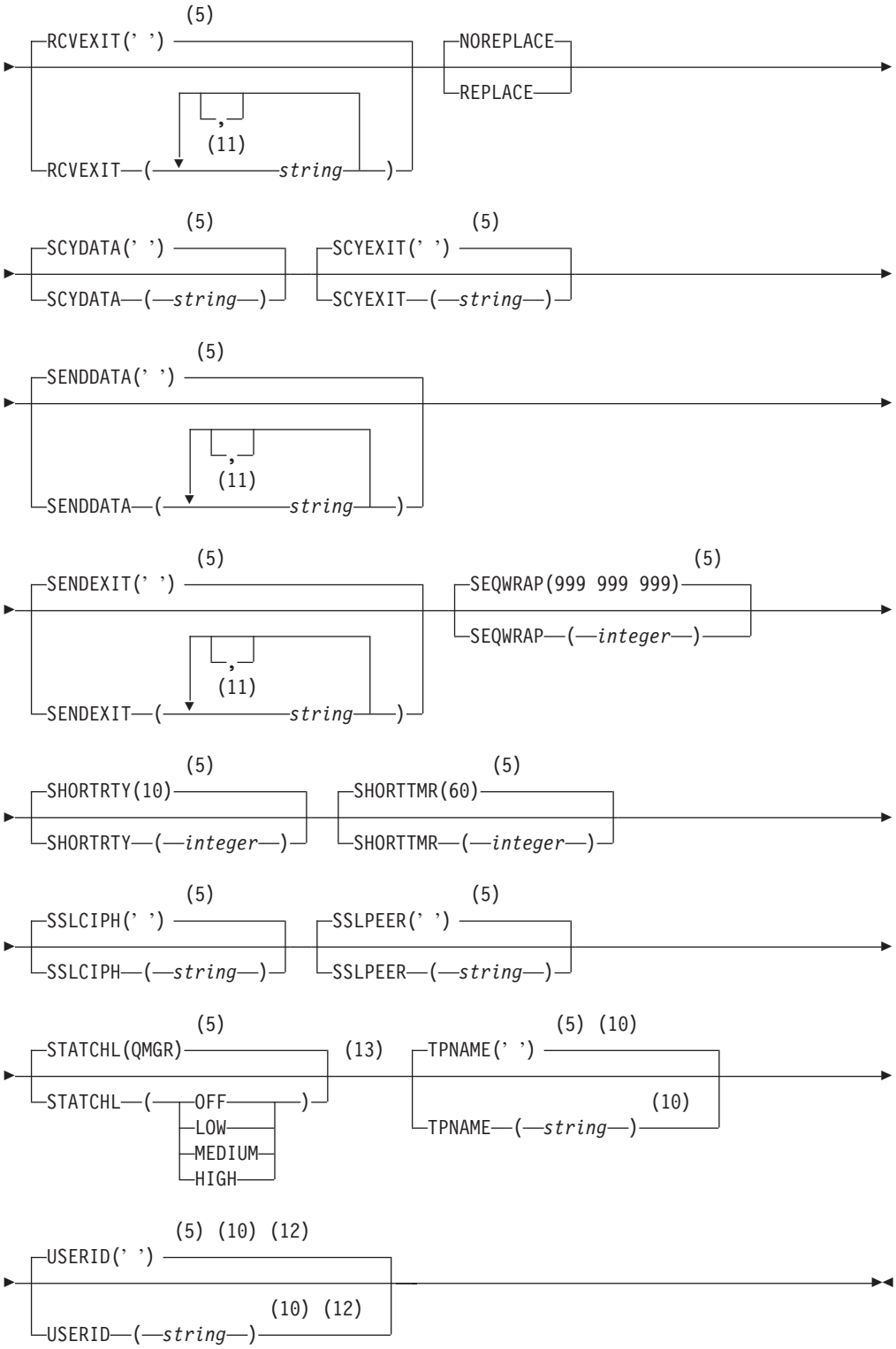
- "Sender channel" on page 145
- "Server channel" on page 149
- "Receiver channel" on page 153
- "Requester channel" on page 156
- "Client-connection channel" on page 160
- "Server-connection channel" on page 163
- "Cluster-sender channel" on page 166
- "Cluster-receiver channel" on page 170

Sender channel

DEFINE CHANNEL




(5)
KAINT(AUTO)
KAINT—(—integer—)
LIKE—(—channel-name—)
(5)
LOCLADDR(' ')
LOCLADDR—(—string—)
(5)
LONGRTY(999 999 999)
LONGRTY—(—integer—)
(5)
LONGTMR(1200)
LONGTMR—(—integer—)
(5)
MAXMSGL(4 194 304)
MAXMSGL—(—integer—)
MCANAME—(—' '—)
(5) (9)
MCATYPE(PROCESS)
(9)
MCATYPE(THREAD)
(5)
MCAUSER(' ')
MCAUSER—(—string—)
(5) (10)
MODENAME(' ')
(10)
MODENAME—(—string—)
(5)
MONCHL(QMGR)
MONCHL—(—OFF—)
LOW
MEDIUM
HIGH
(5)
MSGDATA(' ')
,
(11)
MSGDATA—(—string—)
(5)
MSGEXIT(' ')
,
(11)
MSGEXIT—(—string—)
(5) (6)
NPMSPEED(FAST)
(6)
NPMSPEED(NORMAL)
(5) (10) (12)
PASSWORD(' ')
(10) (12)
PASSWORD—(—string—)
(5)
PROPCTL(COMPAT)
PROPCTL—(—ALL—)
NONE
QSGDISP(QMGR)
(8)
(7)
QSGDISP(COPY)
(7)
QSGDISP(GROUP)
(5)
RCVDATA(' ')
,
(11)
RCVDATA—(—string—)



Notes:

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

2 This is not mandatory on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

3 Valid only on HP OpenVMS.

4 Valid only on Windows.

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

6 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

7 Valid only on WebSphere MQ for z/OS when the queue manager is a member of a queue-sharing group.

8 Valid only on z/OS.

9 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

10 Valid only if TRPTYPE is LU62.

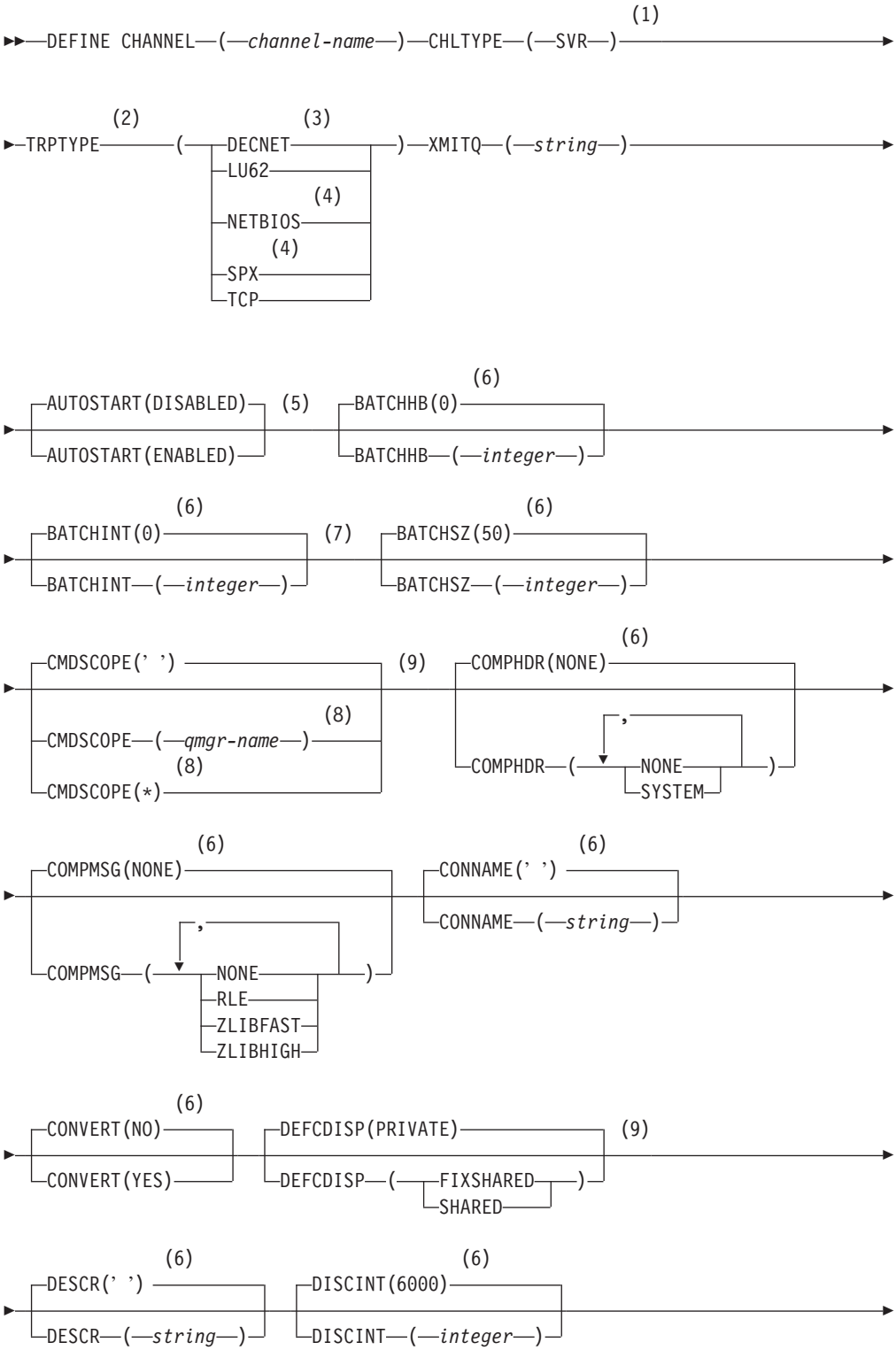
11 You can specify more than one value only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, z/OS, Solaris, and Windows.

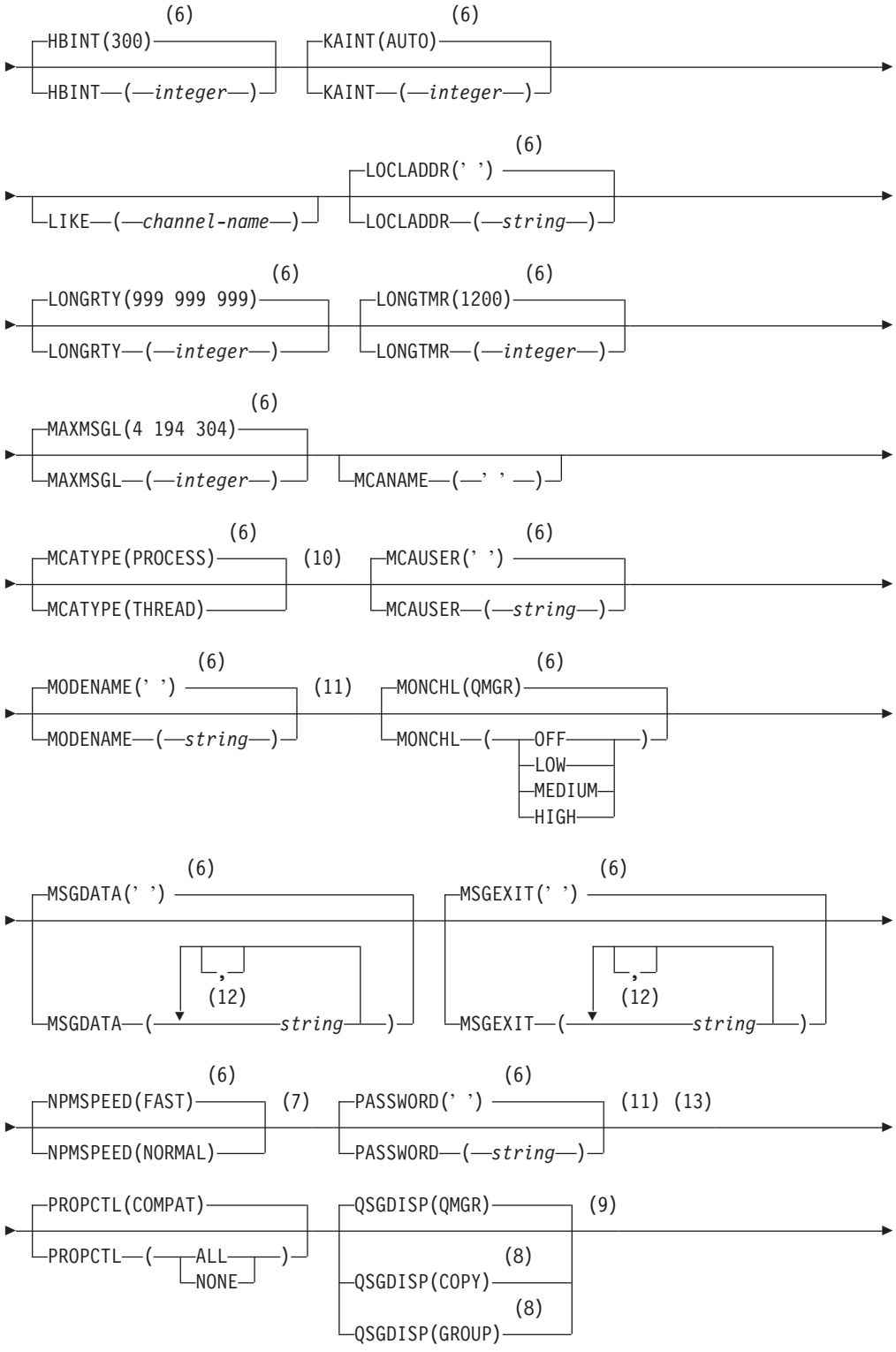
12 Not valid on z/OS.

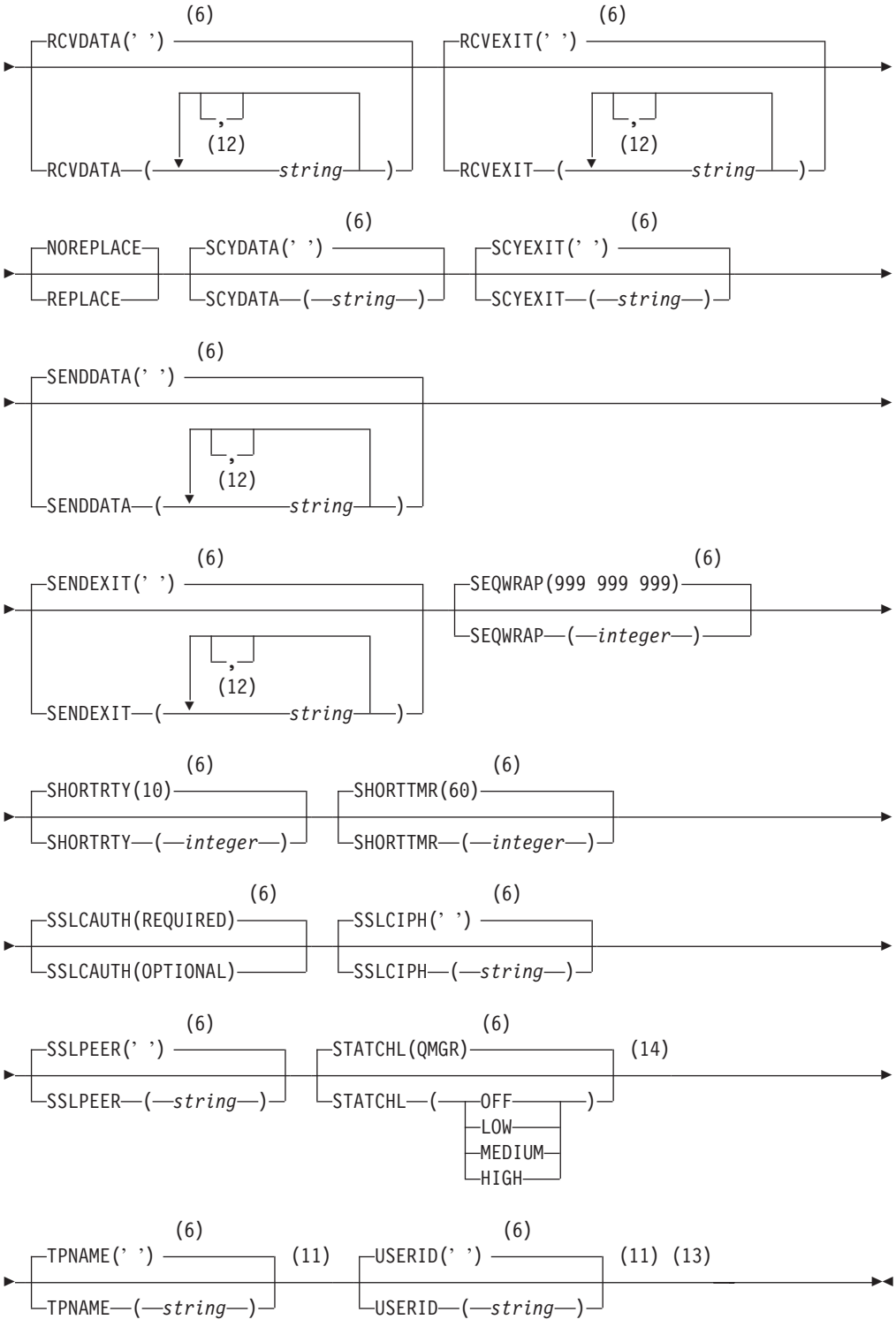
13 Valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

Server channel

DEFINE CHANNEL




(6)
HBINT(300)
HBINT—(—integer—)
(6)
KAINT(AUTO)
KAINT—(—integer—)
LIKE—(—channel-name—)
(6)
LOCLADDR(' ')
LOCLADDR—(—string—)
(6)
LONGRTY(999 999 999)
LONGRTY—(—integer—)
(6)
LONGTMR(1200)
LONGTMR—(—integer—)
(6)
MAXMSGL(4 194 304)
MAXMSGL—(—integer—)
MCANAME—(—' '—)
(6)
MCATYPE(PROCESS)
MCATYPE(THREAD)
(10)
(6)
MCAUSER(' ')
MCAUSER—(—string—)
(6)
MODENAME(' ')
MODENAME—(—string—)
(11)
(6)
MONCHL(QMGR)
MONCHL—(—OFF—)
LOW
MEDIUM
HIGH
(6)
MSGDATA(' ')
,
(12)
MSGDATA—(—string—)
(6)
MSGEXIT(' ')
,
(12)
MSGEXIT—(—string—)
(6)
NPMSPEED(FAST)
NPMSPEED(NORMAL)
(7)
(6)
PASSWORD(' ')
PASSWORD—(—string—)
(11) (13)
PROPCTL(COMPAT)
PROPCTL—(—ALL—)
NONE
QSGDISP(QMGR)
(9)
(8)
QSGDISP(COPY)
(8)
QSGDISP(GROUP)



Notes:

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

2 This is not mandatory on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

3 Valid only on HP OpenVMS.

4 Valid only on Windows.

5 Valid only on Compaq NSK.

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

7 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

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

9 Valid only on z/OS.

10 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

11 Valid only if TRPTYPE is LU62.

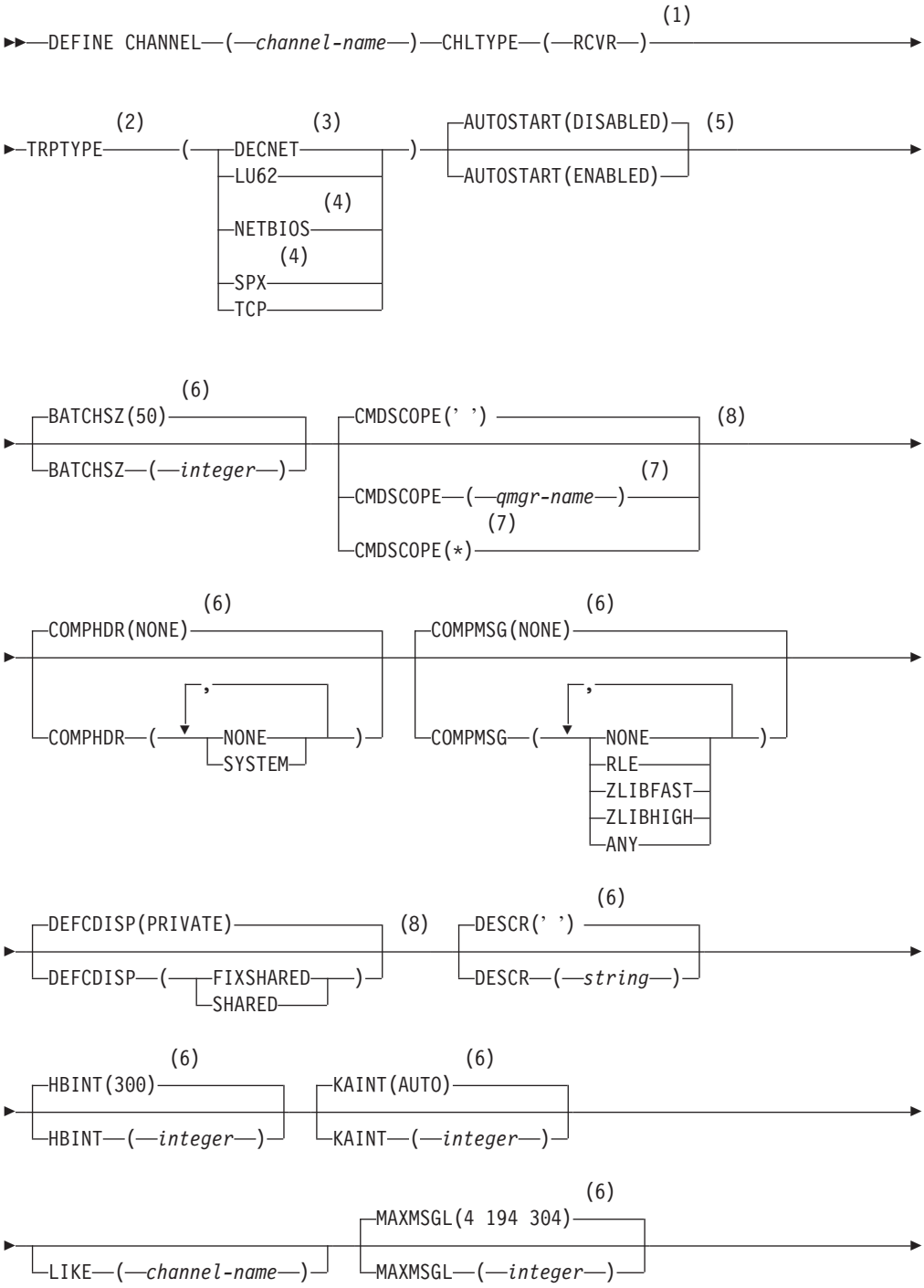
12 You can specify more than one value only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

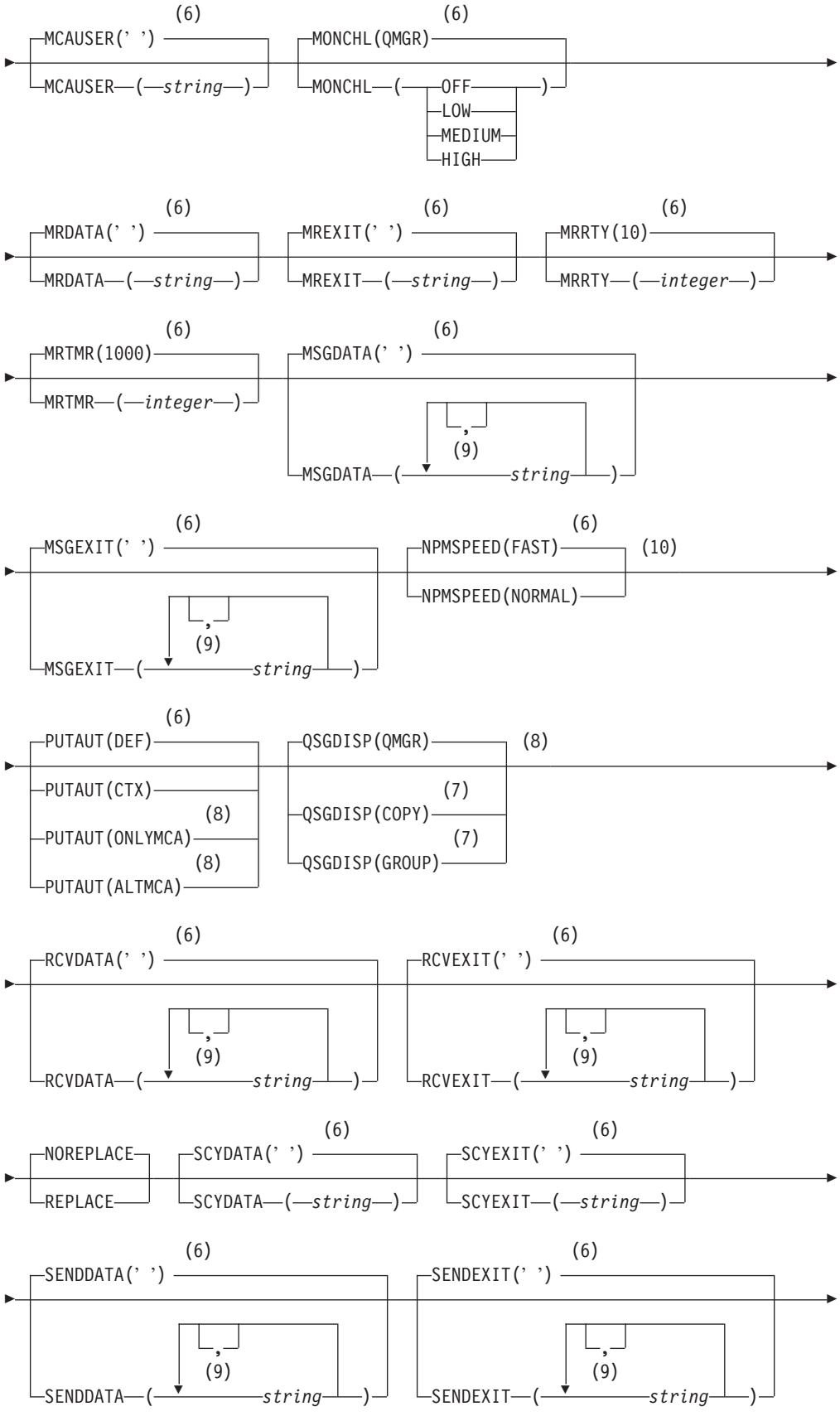
13 Not valid on z/OS.

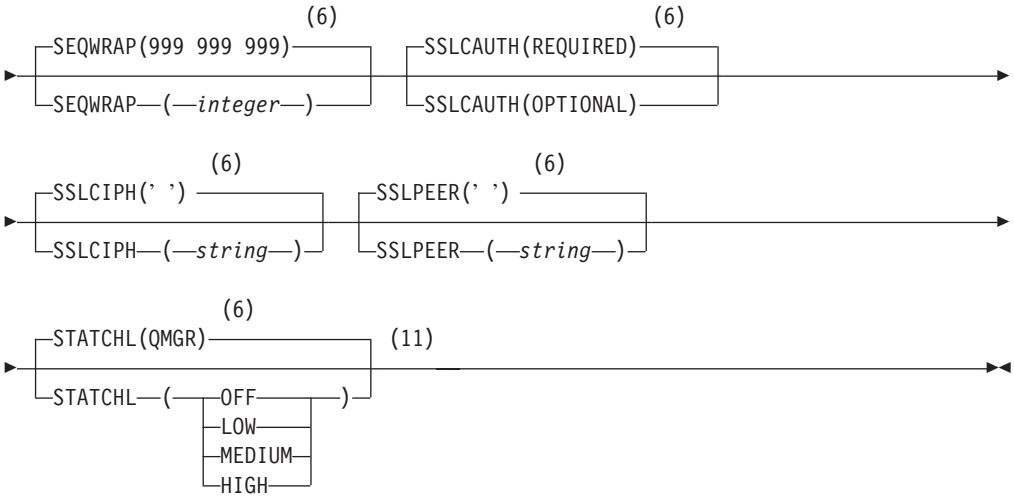
14 Valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

Receiver channel

DEFINE CHANNEL




(6)
MCAUSER(' ')
MCAUSER—(—string—)—
(6)
MONCHL(QMGR)
MONCHL—(—OFF—)—
LOW
MEDIUM
HIGH
(6)
MRDATA(' ')
MRDATA—(—string—)—
(6)
MREXIT(' ')
MREXIT—(—string—)—
(6)
MRRTY(10)
MRRTY—(—integer—)—
(6)
MRTMR(1000)
MRTMR—(—integer—)—
(6)
MSGDATA(' ')
,
(9)
MSGDATA—(—string—)—
(6)
MSGEXIT(' ')
,
(9)
MSGEXIT—(—string—)—
(6)
NPMSPEED(FAST)
NPMSPEED(NORMAL)
(10)
(6)
PUTAUT(DEF)
PUTAUT(CTX)
(8)
PUTAUT(ONLYMCA)
(8)
PUTAUT(ALTMCA)
QSGDISP(QMGR)
(8)
(7)
QSGDISP(COPY)
(7)
QSGDISP(GROUP)
(6)
RCVDATA(' ')
,
(9)
RCVDATA—(—string—)—
(6)
RCVEXIT(' ')
,
(9)
RCVEXIT—(—string—)—
NOREPLACE
REPLACE
(6)
SCYDATA(' ')
SCYDATA—(—string—)—
(6)
SCYEXIT(' ')
SCYEXIT—(—string—)—
(6)
SENDDATA(' ')
,
(9)
SENDDATA—(—string—)—
(6)
SENDEXIT(' ')
,
(9)
SENDEXIT—(—string—)—



Notes:

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

2 This is not mandatory on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

3 Valid only on HP OpenVMS.

4 Valid only on Windows.

5 Valid only on Compaq NSK.

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

7 Valid only on WebSphere MQ for z/OS when the queue manager is a member of a queue-sharing group.

8 Valid only on z/OS.

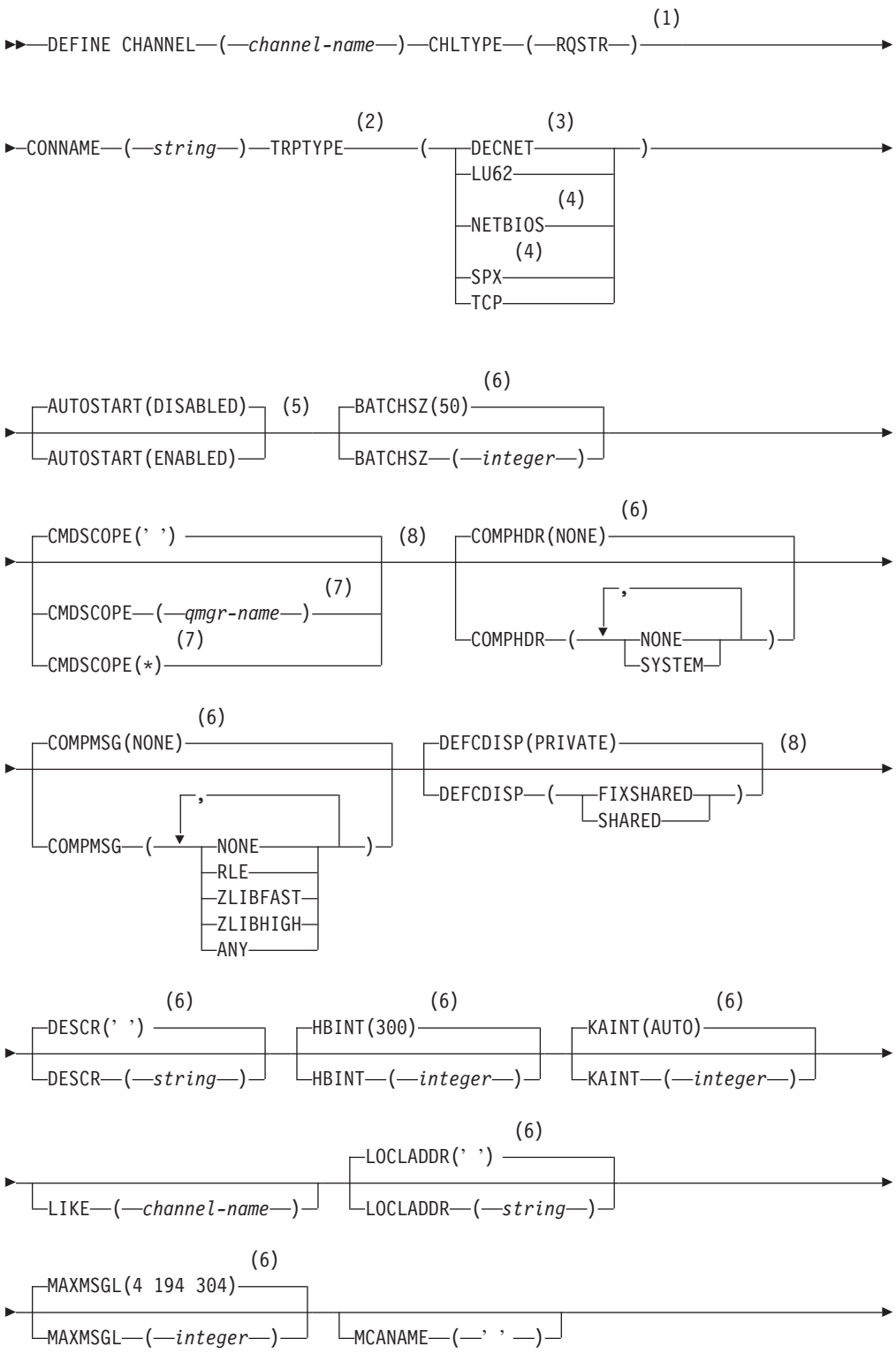
9 You can specify more than one value only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

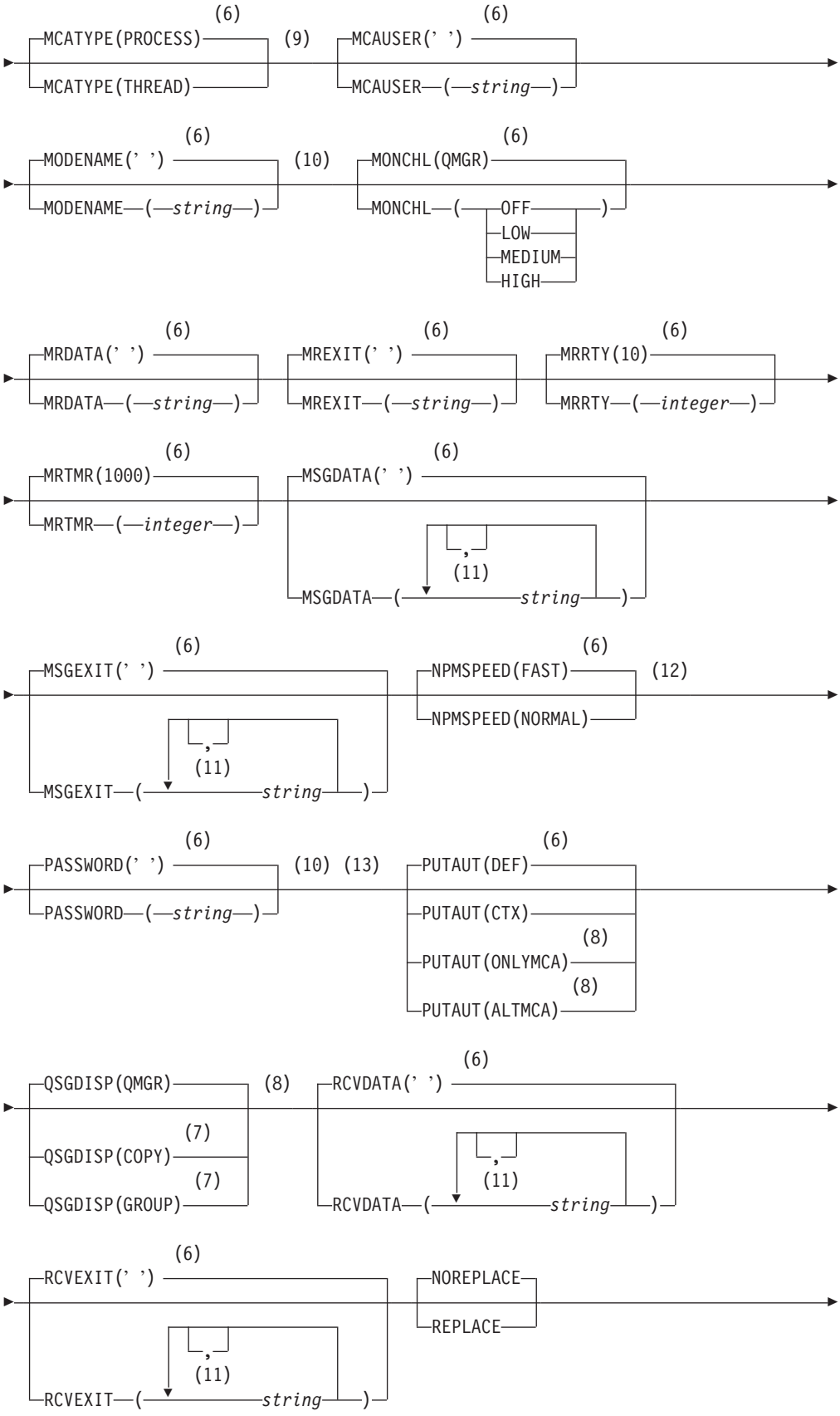
10 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

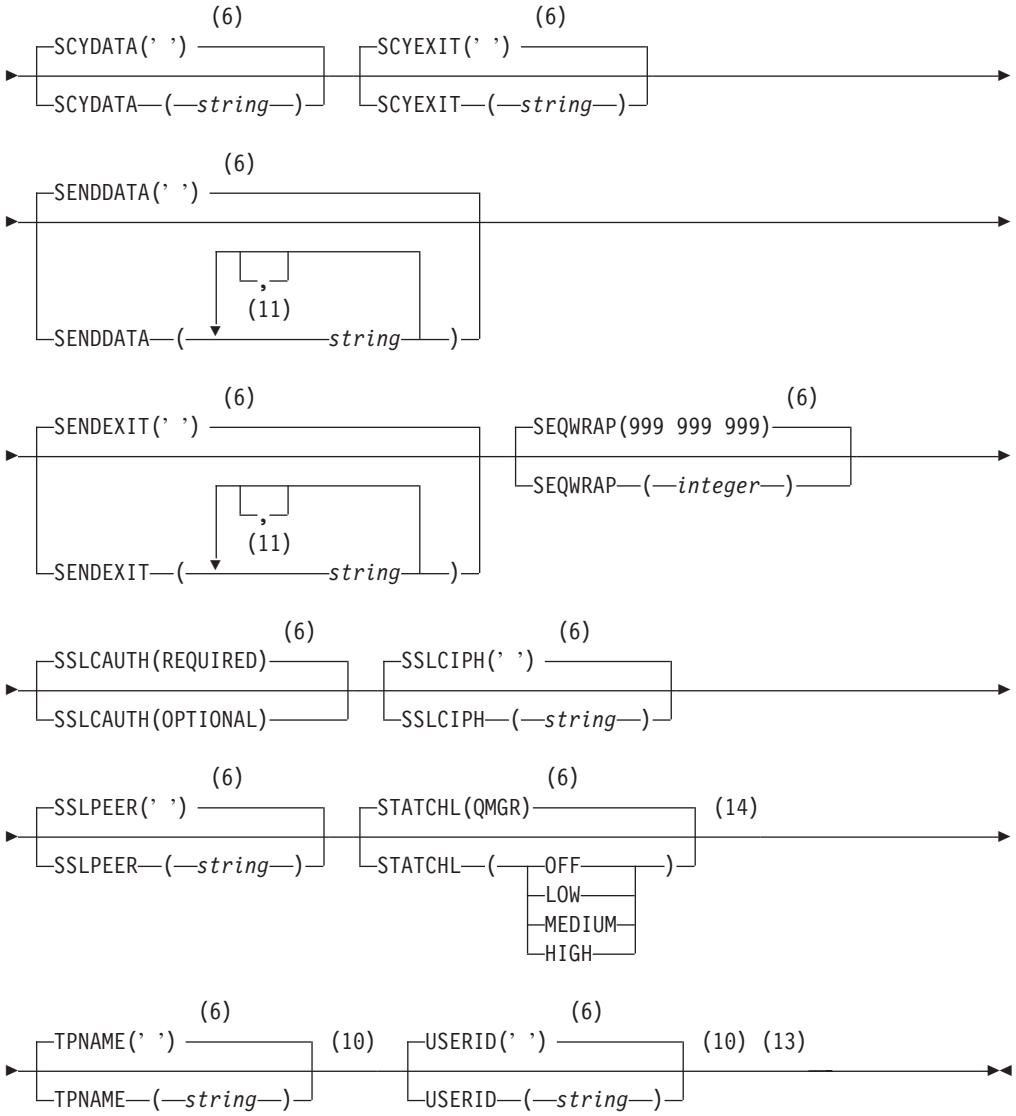
11 Valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

Requester channel

DEFINE CHANNEL




(6)
MCATYPE(PROCESS)
(9)
MCATYPE(THREAD)
(6)
MCAUSER(' ')
MCAUSER—(—string—)
(6)
MODENAME(' ')
(10)
MODENAME—(—string—)
(6)
MONCHL(QMGR)
MONCHL—(—OFF—)
LOW
MEDIUM
HIGH
(6)
MRDATA(' ')
MRDATA—(—string—)
(6)
MREXIT(' ')
MREXIT—(—string—)
(6)
MRRTY(10)
MRRTY—(—integer—)
(6)
MRTMR(1000)
MRTMR—(—integer—)
(6)
MSGDATA(' ')
,
(11)
MSGDATA—(—string—)
(6)
MSGEXIT(' ')
,
(11)
MSGEXIT—(—string—)
(6)
NPMSPEED(FAST)
(12)
NPMSPEED(NORMAL)
(6)
PASSWORD(' ')
(10)
(13)
PASSWORD—(—string—)
(6)
PUTAUT(DEF)
PUTAUT(CTX)
(8)
PUTAUT(ONLYMCA)
(8)
PUTAUT(ALTMCA)
QSGDISP(QMGR)
(8)
(7)
QSGDISP(COPY)
(7)
QSGDISP(GROUP)
(6)
RCVDATA(' ')
,
(11)
RCVDATA—(—string—)
(6)
RCVEXIT(' ')
,
(11)
RCVEXIT—(—string—)
NOREPLACE
REPLACE



Notes:

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

2 This is not mandatory on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

3 Valid only on HP OpenVMS.

4 Valid only on Windows.

5 Valid only on Compaq NSK.

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

7 Valid only on WebSphere MQ for z/OS when the queue manager is a member of a queue-sharing group.

8 Valid only on z/OS.

9 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

10 Valid only if TRPTYPE is LU62.

11 You can specify more than one value only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

12 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

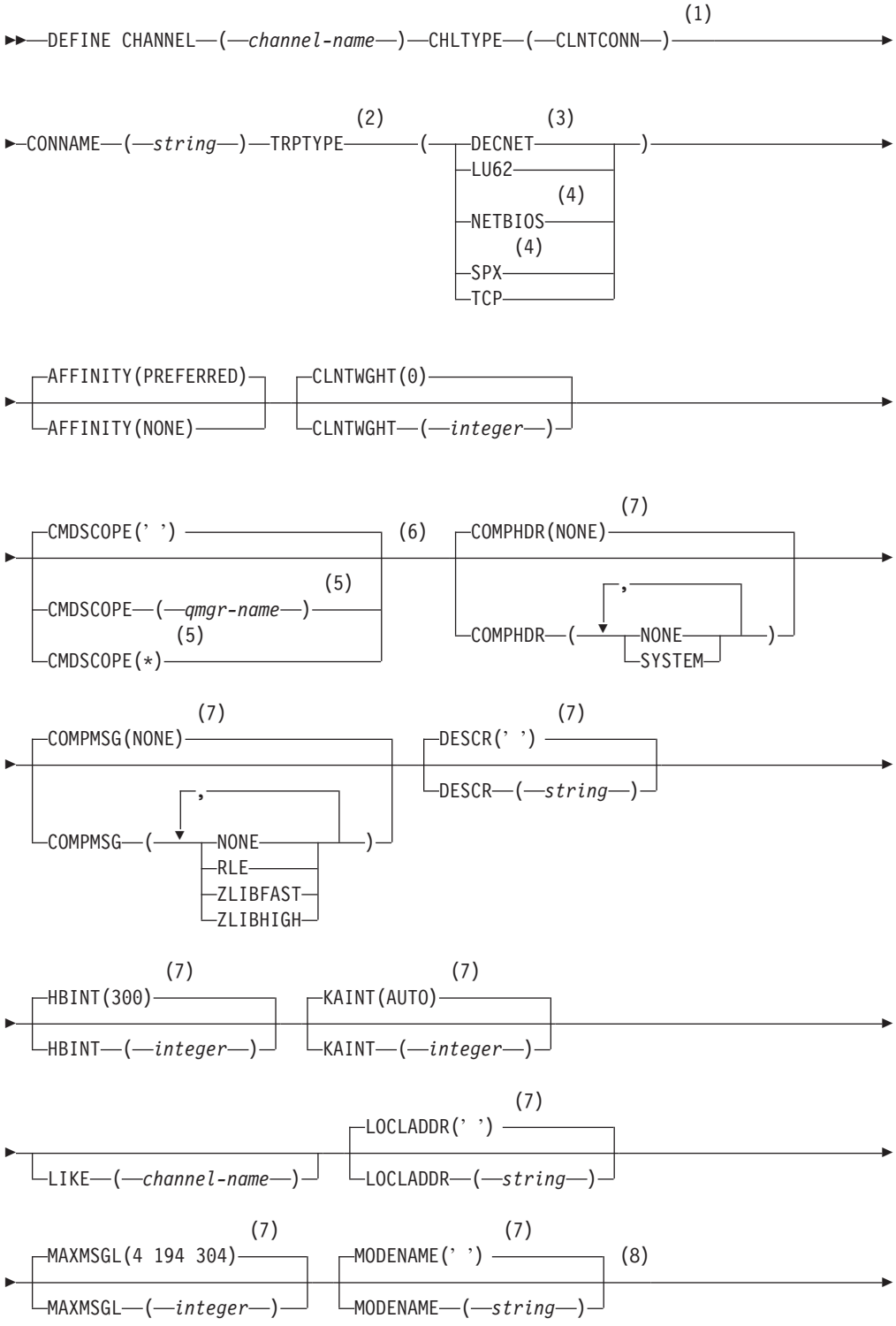
13 Not valid on z/OS.

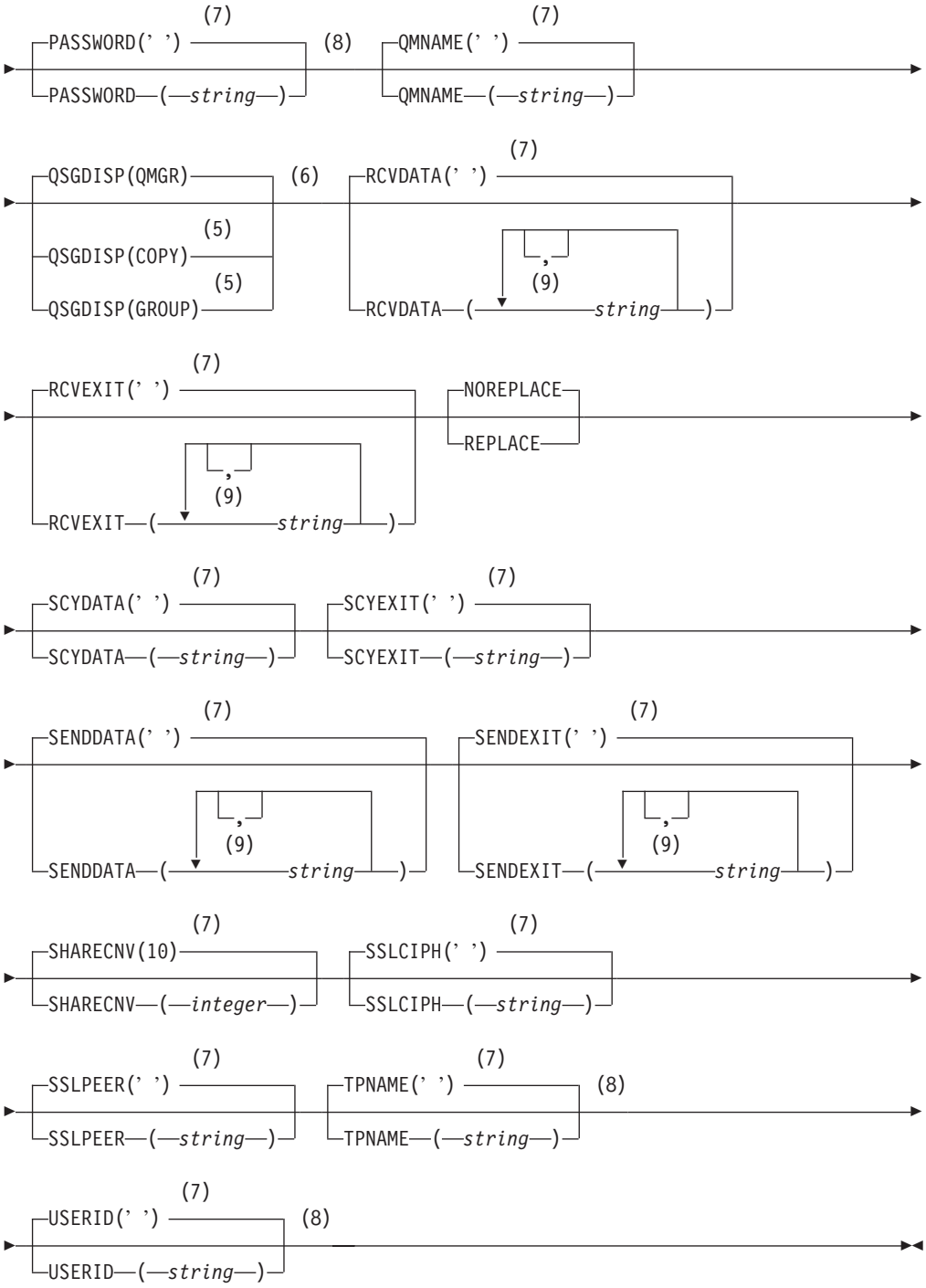
14 Valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

Client-connection channel

Syntax diagram for a client-connection channel when using the DEFINE CHANNEL command.

DEFINE CHANNEL





Notes:

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

2 This is not mandatory on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

3 Valid only on HP OpenVMS.

4 Valid only for clients to be run on DOS or Windows.

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

6 Valid only on z/OS.

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

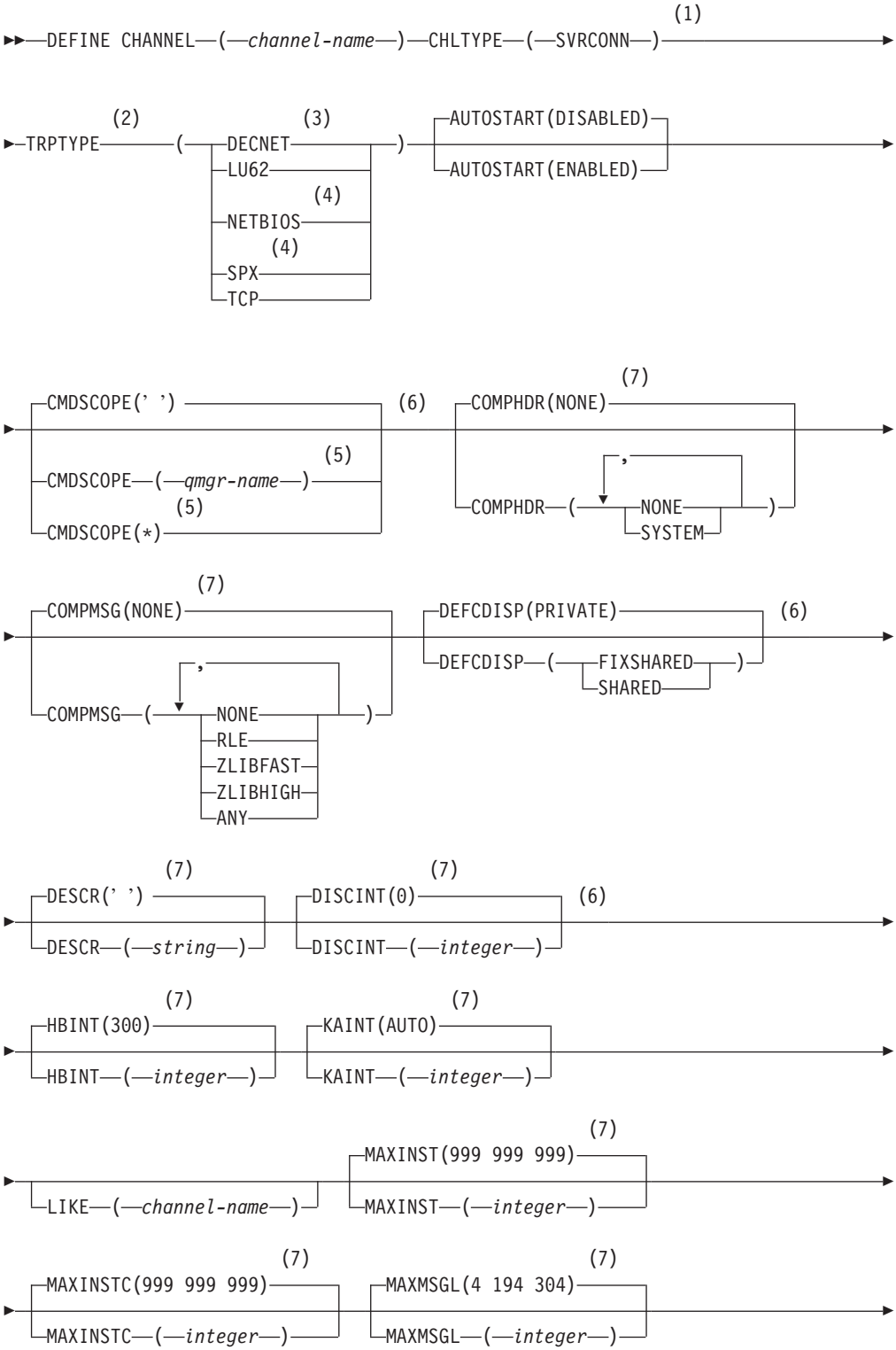
8 Valid only if TRPTYPE is LU62.

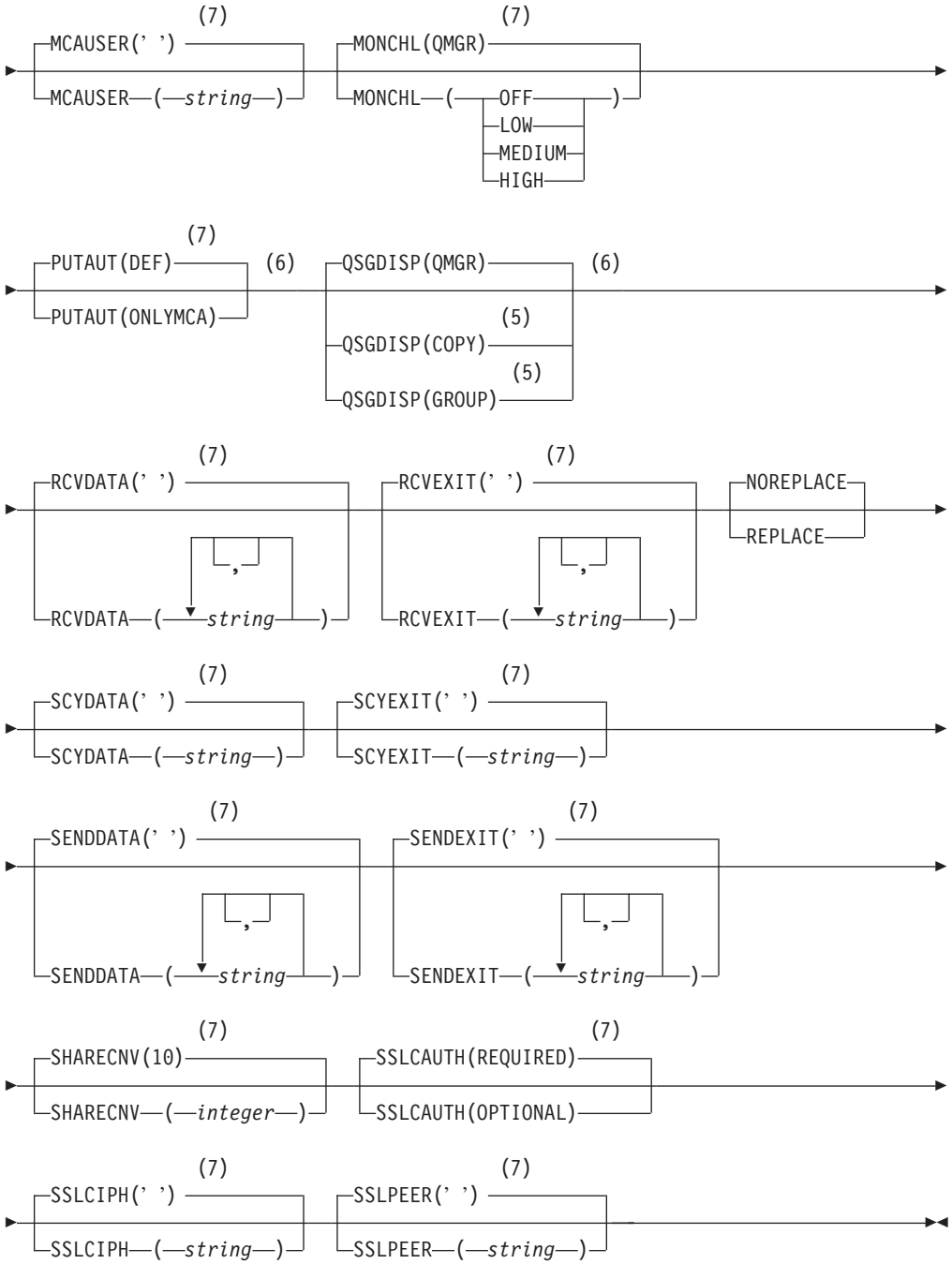
9 You can specify more than one value only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

Server-connection channel

Syntax diagram for a server-connection channel when using the DEFINE CHANNEL command.

DEFINE CHANNEL





Notes:

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

2 This is not mandatory.

3 Valid only on HP OpenVMS.

4 Valid only for clients to be run on Windows.

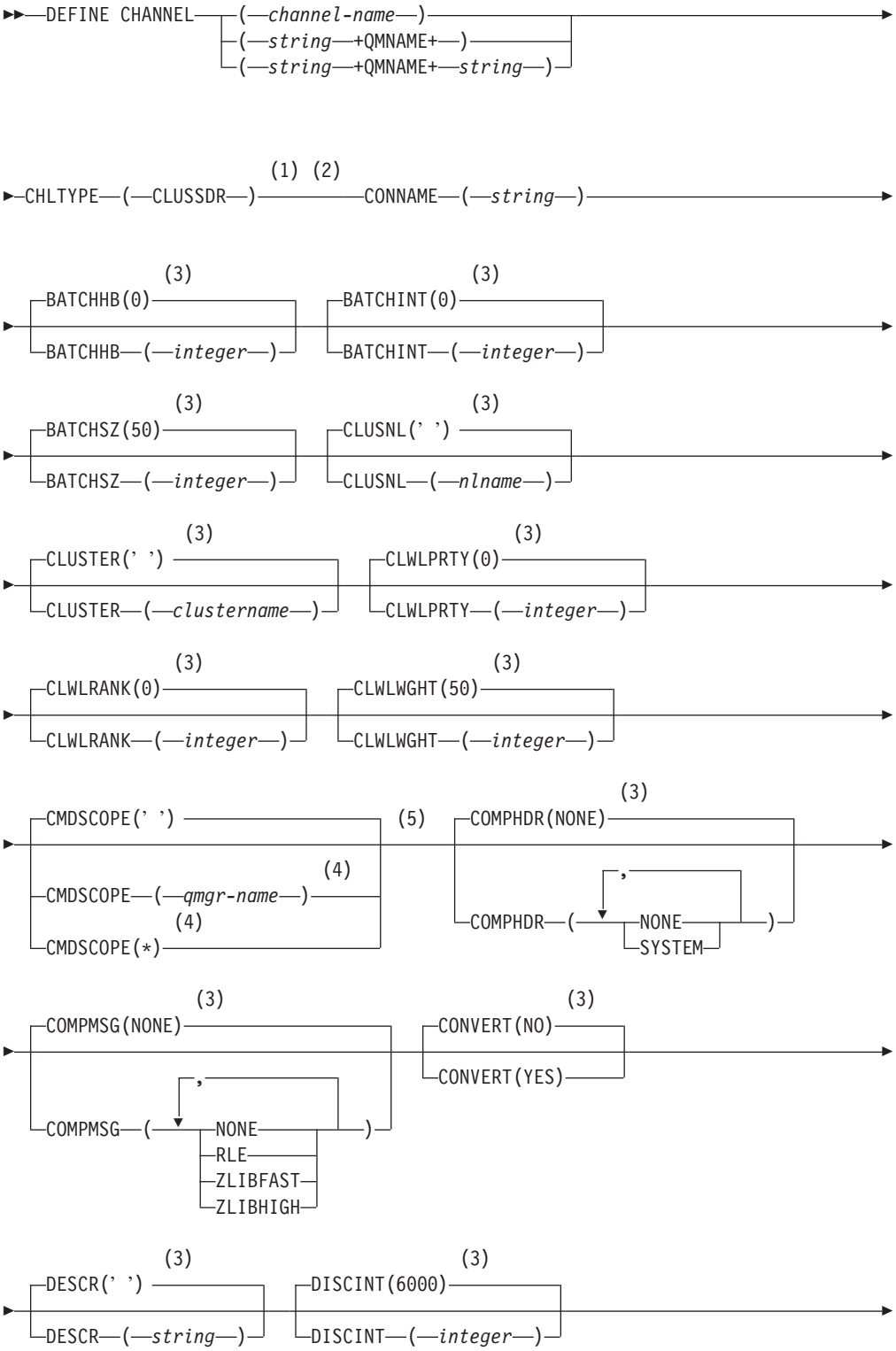
5 Valid only on WebSphere MQ for z/OS when the queue manager is a member of a queue-sharing group.

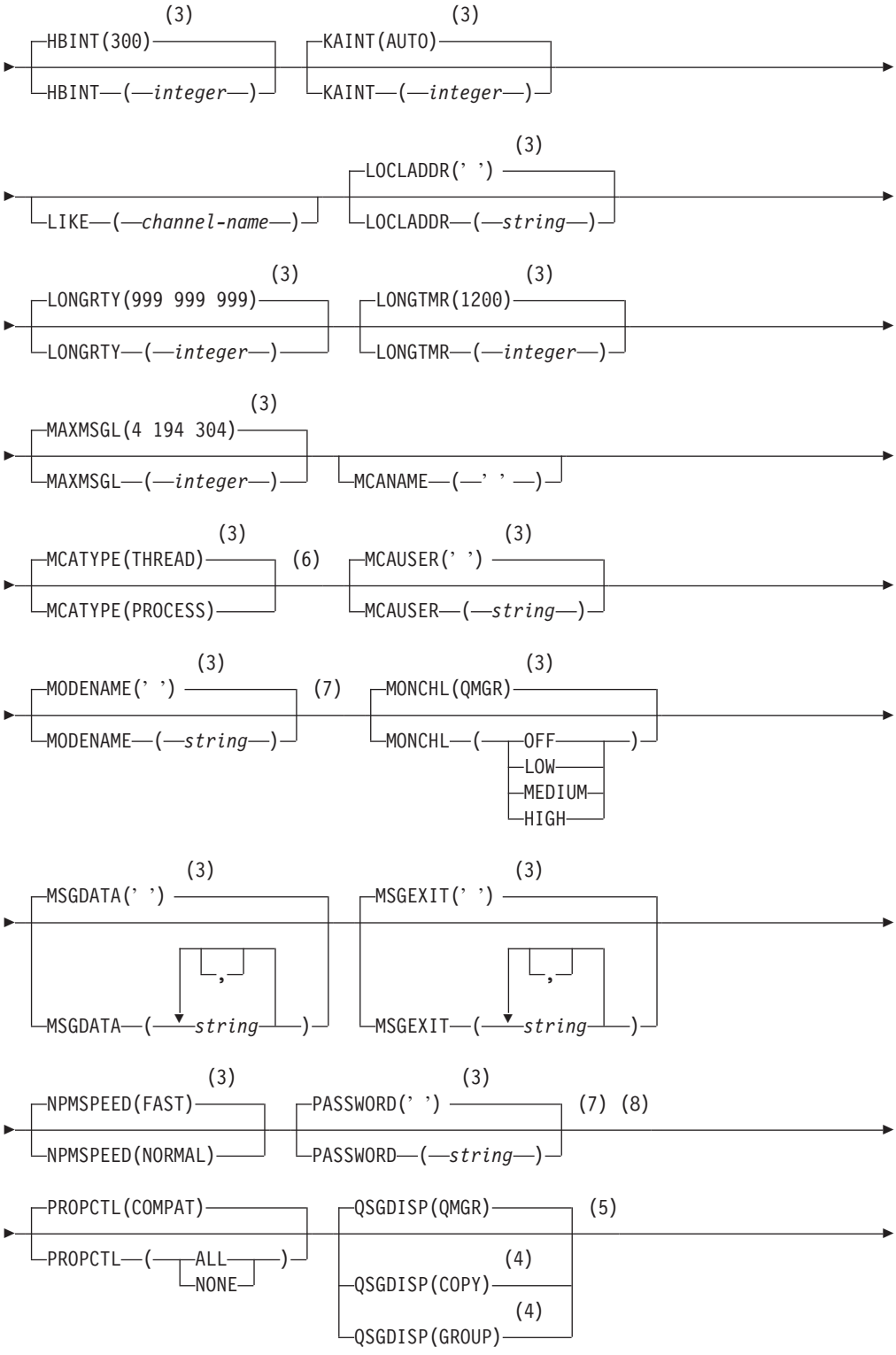
6 Valid only on z/OS.

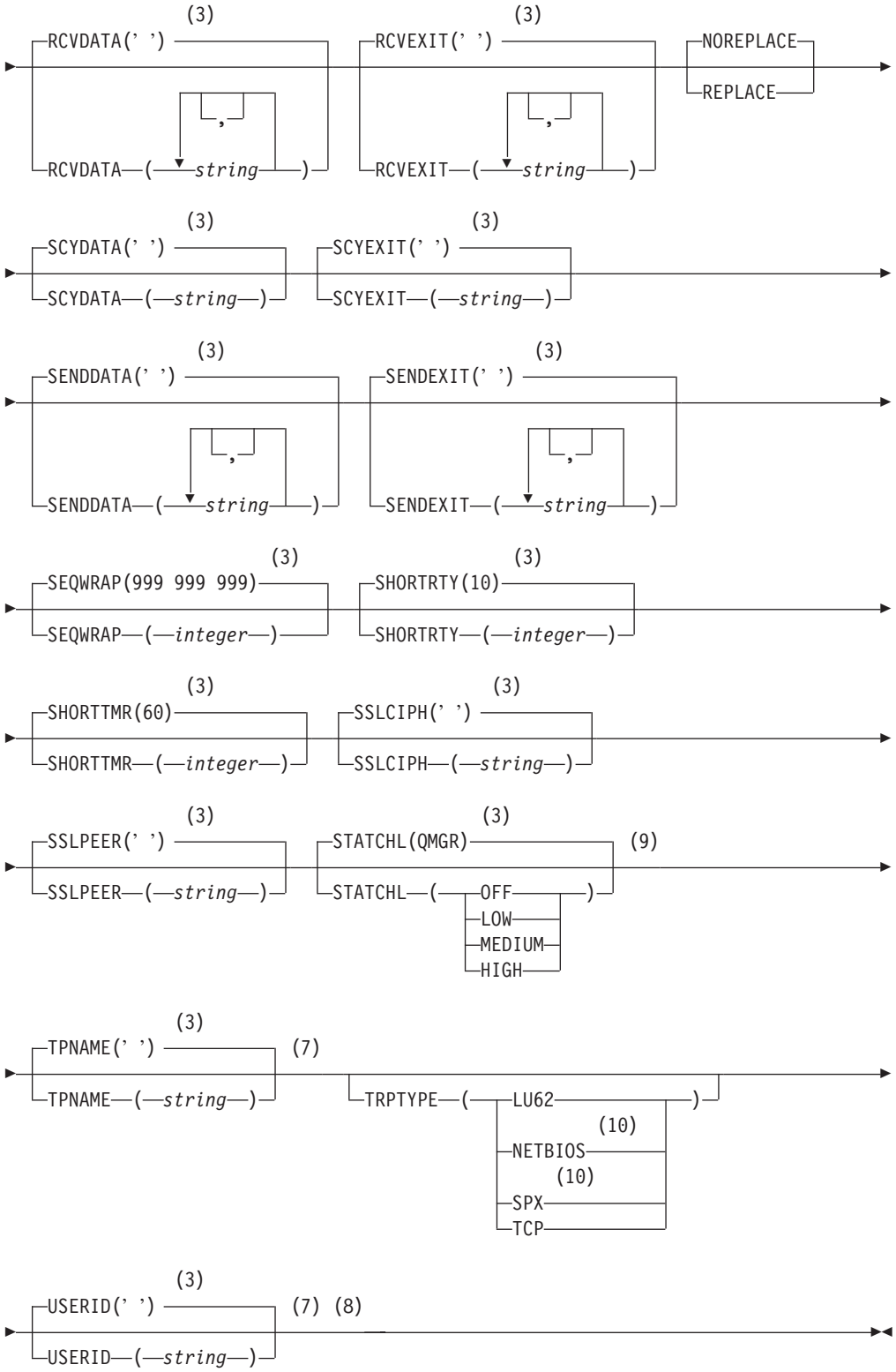
7 This is the default supplied with WebSphere MQ, but your installation might have changed it.

Cluster-sender channel

DEFINE CHANNEL




(3)
HBINT(300)
HBINT—(—integer—)
(3)
KAINT(AUTO)
KAINT—(—integer—)
LIKE—(—channel-name—)
(3)
LOCLADDR(' ')
LOCLADDR—(—string—)
(3)
LONGRTY(999 999 999)
LONGRTY—(—integer—)
(3)
LONGTMR(1200)
LONGTMR—(—integer—)
(3)
MAXMSGL(4 194 304)
MAXMSGL—(—integer—)
MCANAME—(—' '—)
(3)
MCATYPE(THREAD)
MCATYPE(PROCESS)
(6)
(3)
MCAUSER(' ')
MCAUSER—(—string—)
(3)
MODENAME(' ')
MODENAME—(—string—)
(7)
(3)
MONCHL(QMGR)
MONCHL—(—OFF—)
LOW
MEDIUM
HIGH
(3)
MSGDATA(' ')
MSGDATA—(—string—)
(3)
MSGEXIT(' ')
MSGEXIT—(—string—)
(3)
NPMSPEED(FAST)
NPMSPEED(NORMAL)
(3)
PASSWORD(' ')
PASSWORD—(—string—)
(7) (8)
PROPCTL(COMPAT)
PROPCTL—(—ALL—)
NONE
QSGDISP(QMGR)
(5)
(4)
QSGDISP(COPY)
(4)
QSGDISP(GROUP)



Notes:

1 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, 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 on WebSphere MQ for z/OS when the queue manager is a member of a queue-sharing group.

5 Valid only on z/OS.

6 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

7 Valid only if TRPTYPE is LU62.

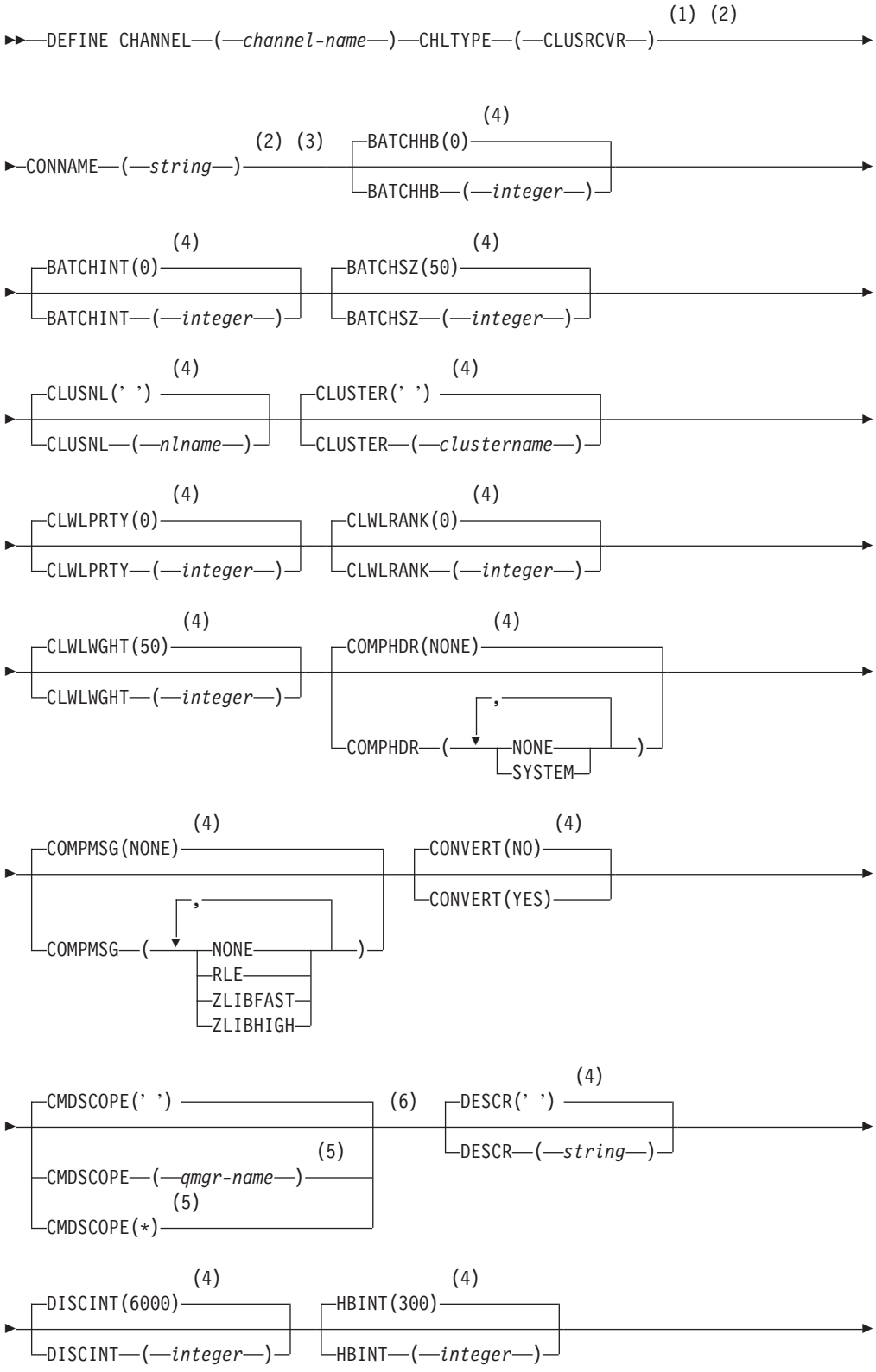
8 Not valid on z/OS.

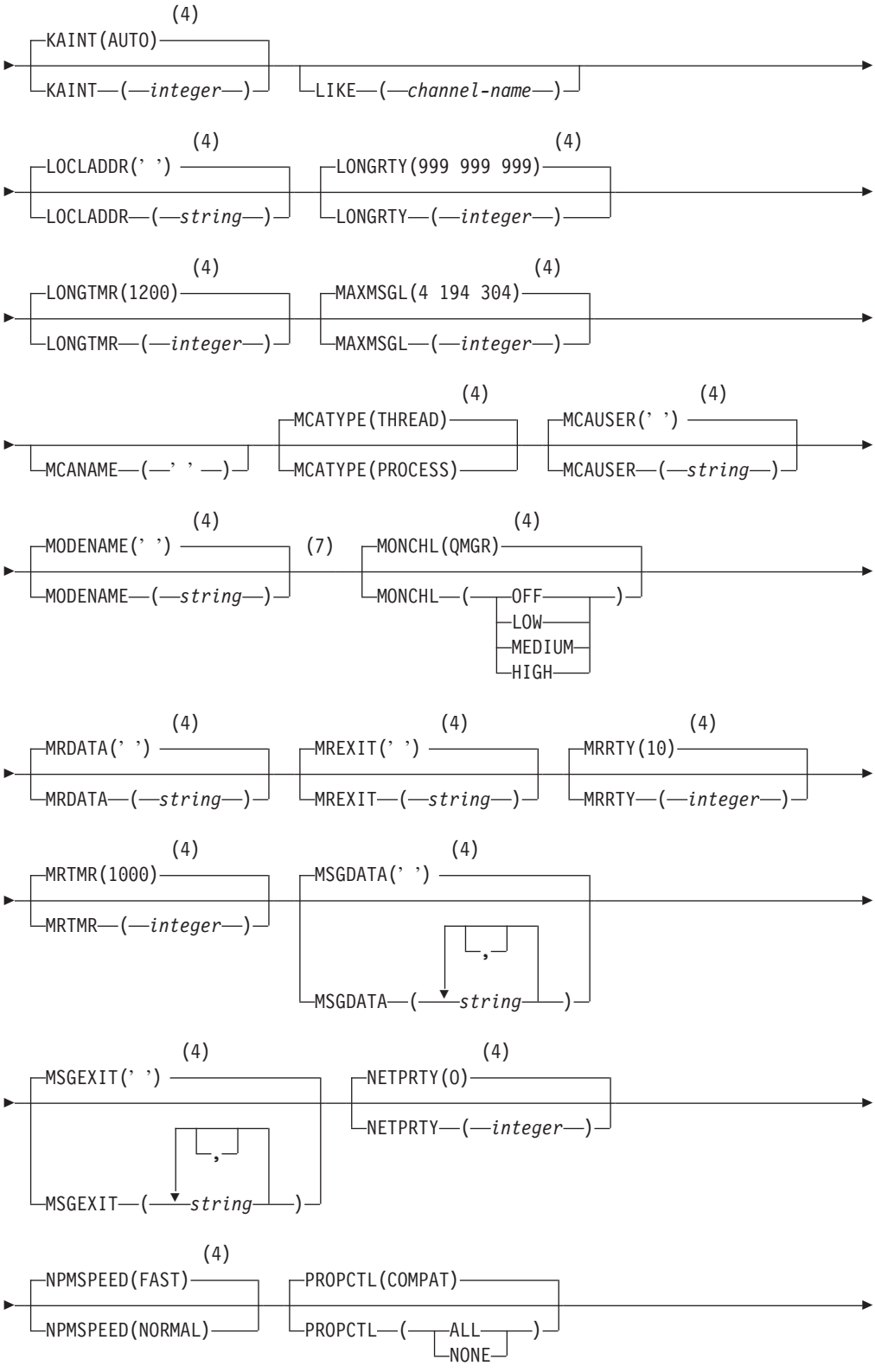
9 Valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

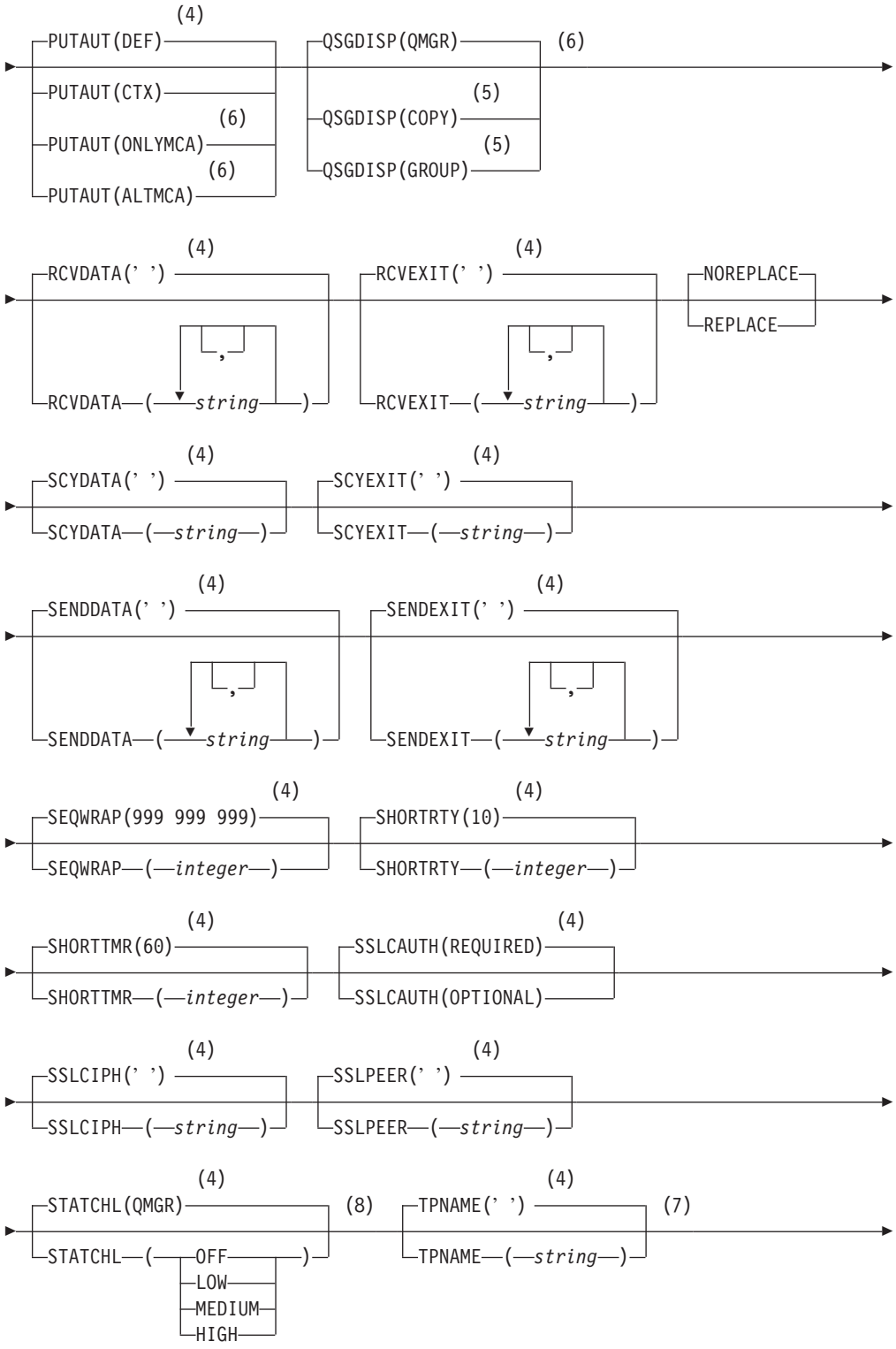
10 Valid only on Windows.

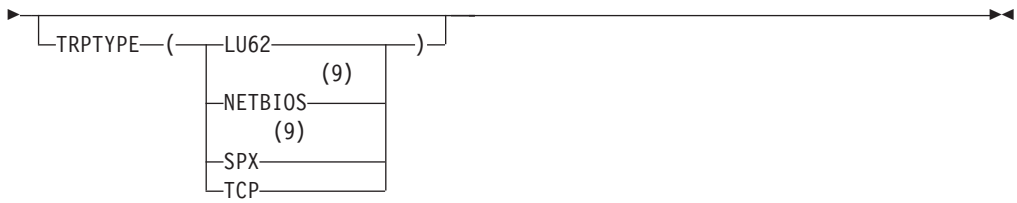
Cluster-receiver channel

DEFINE CHANNEL




(4)
KAINT(AUTO)
KAINT—(—integer—)
LIKE—(—channel-name—)
(4)
LOCLADDR(' ')
LOCLADDR—(—string—)
(4)
LONGRTY(999 999 999)
LONGRTY—(—integer—)
(4)
LONGTMR(1200)
LONGTMR—(—integer—)
(4)
MAXMSGL(4 194 304)
MAXMSGL—(—integer—)
MCANAME—(—' '—)
(4)
MCATYPE(THREAD)
MCATYPE(PROCESS)
(4)
MCAUSER(' ')
MCAUSER—(—string—)
(4)
MODENAME(' ')
MODENAME—(—string—)
(7)
(4)
MONCHL(QMGR)
MONCHL—(—OFF—)
LOW
MEDIUM
HIGH
(4)
MRDATA(' ')
MRDATA—(—string—)
(4)
MREXIT(' ')
MREXIT—(—string—)
(4)
MRRTY(10)
MRRTY—(—integer—)
(4)
MRTMR(1000)
MRTMR—(—integer—)
(4)
MSGDATA(' ')
,
MSGDATA—(—string—)
(4)
MSGEXIT(' ')
,
MSGEXIT—(—string—)
(4)
NETPRTY(0)
NETPRTY—(—integer—)
(4)
NPMSPEED(FAST)
NPMSPEED(NORMAL)
PROPCTL(COMPAT)
PROPCTL—(—ALL—)
NONE


(4)
PUTAUT(DEF)
PUTAUT(CTX)
(6)
PUTAUT(ONLYMCA)
(6)
PUTAUT(ALTMCA)
QSGDISP(QMGR)
(6)
(5)
QSGDISP(COPY)
(5)
QSGDISP(GROUP)
(4)
RCVDATA(' ')
RCVDATA—(—string—)
(4)
RCVEXIT(' ')
RCVEXIT—(—string—)
NOREPLACE
REPLACE
(4)
SCYDATA(' ')
SCYDATA—(—string—)
(4)
SCYEXIT(' ')
SCYEXIT—(—string—)
(4)
SENDDATA(' ')
SENDDATA—(—string—)
(4)
SENDEXIT(' ')
SENDEXIT—(—string—)
(4)
SEQWRAP(999 999 999)
SEQWRAP—(—integer—)
(4)
SHORTRTY(10)
SHORTRTY—(—integer—)
(4)
SHORTTMR(60)
SHORTTMR—(—integer—)
(4)
SSLCAUTH(REQUIRED)
SSLCAUTH(OPTIONAL)
(4)
SSLCIPH(' ')
SSLCIPH—(—string—)
(4)
SSLPEER(' ')
SSLPEER—(—string—)
(4)
STATCHL(QMGR)
(8)
STATCHL—(—OFF—)
LOW
MEDIUM
HIGH
(4)
TPNAME(' ')
(7)
TPNAME—(—string—)



Notes:

1 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, 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 on WebSphere MQ for z/OS when the queue manager is a member of a queue-sharing group.

6 Valid only on z/OS.

7 Valid only if TRPTYPE is LU62.

8 Valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

9 Valid only on Windows.

Parameter descriptions for DEFINE CHANNEL

Parameter descriptions for the ALTER CHANNEL and DEFINE CHANNEL commands.

The parameter descriptions apply to both commands, 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.

Table 4 shows the parameters that are relevant for each type of channel. There is a description of each parameter after the table. Parameters are optional unless the description states that they are required.

Table 4. DEFINE and ALTER CHANNEL parameters

Parameter	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR
AFFINITY					X			
AUTOSTART		X	X	X		X		
BATCHHB	X	X					X	X
BATCHINT	X	X					X	X
BATCHSZ	X	X	X	X			X	X
channel-name	X	X	X	X	X	X	X	X

Table 4. DEFINE and ALTER CHANNEL parameters (continued)

Parameter	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR
CHLTYPE	X	X	X	X	X	X	X	X
CLNTWGHT					X			
CLUSNL							X	X
CLUSTER							X	X
CLWLPRTY							X	X
CLWLRANK							X	X
CLWLWGHT							X	X
CMDSCOPE	X	X	X	X	X	X	X	X
COMPHDR	X	X	X	X	X	X	X	X
COMPMSG	X	X	X	X	X	X	X	X
CONNAME	X	X		X	X		X	X
CONVERT	X	X					X	X
DEFCDISP	X	X	X	X		X		
DESCR	X	X	X	X	X	X	X	X
DISCINT	X	X				X	X	X
HBINT	X	X	X	X	X	X	X	X
KAINT	X	X	X	X	X	X	X	X
LIKE	X	X	X	X	X	X	X	X
LOCLADDR	X	X		X	X		X	X
LONGRTY	X	X					X	X
LONGTMR	X	X					X	X
MAXINST						X		
MAXINSTC						X		
MAXMSGL	X	X	X	X	X	X	X	X
MCANAME	X	X		X			X	X
MCATYPE	X	X		X			X	X
MCAUSER	X	X	X	X		X	X	X
MODENAME	X	X		X	X		X	X
MONCHL	X	X	X	X		X	X	X
MRDATA			X	X				X
MREXIT			X	X				X
MRRTY			X	X				X
MRTMR			X	X				X
MSGDATA	X	X	X	X			X	X
MSGEXIT	X	X	X	X			X	X
NETPRTY								X
NPMSPEED	X	X	X	X			X	X
PASSWORD	X	X		X	X		X	X
PROPCTL	X	X					X	X

Table 4. DEFINE and ALTER CHANNEL parameters (continued)

Parameter	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR
PUTAUT			X	X		X		X
QMNAME					X			
QSGDISP	X	X	X	X	X	X	X	X
RCVDATA	X	X	X	X	X	X	X	X
RCVEXIT	X	X	X	X	X	X	X	X
REPLACE	X	X	X	X	X	X	X	X
SCYDATA	X	X	X	X	X	X	X	X
SCYEXIT	X	X	X	X	X	X	X	X
SENDDATA	X	X	X	X	X	X	X	X
SENDEXIT	X	X	X	X	X	X	X	X
SEQWRAP	X	X	X	X			X	X
SHARECNV					X	X		
SHORTRTY	X	X					X	X
SHORTTMR	X	X					X	X
SSLCAUTH		X	X	X		X		X
SSLCIPH	X	X	X	X	X	X	X	X
SSLPEER	X	X	X	X	X	X	X	X
STATCHL	X	X	X	X			X	X
TPNAME	X	X		X	X	X	X	X
TRPTYPE	X	X	X	X	X	X	X	X
USERID	X	X		X	X		X	
XMITQ	X	X						

(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, i5/OS, 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 5.

AFFINITY

The channel affinity attribute is used so client applications that connect multiple times using the same queue manager name can choose whether to

use the same client channel definition for each connection. This attribute is intended to be used when multiple applicable channel definitions are available.

PREFERRED
The first connection in a process reading a CCDT creates a list of applicable definitions based on the weighting with any applicable CLNTWGHT(0) definitions first and in alphabetical order. Each connection in the process attempts to connect using the first definition in the list. If a connection is unsuccessful the next definition is used. Unsuccessful non CLNTWGHT(0) definitions are moved to the end of the list. CLNTWGHT(0) definitions remain at the start of the list and are selected first for each connection. For C, C++ and .NET (including fully managed .NET) clients the list is updated if the CCDT has been modified since the list was created. Each client process with the same hostname creates the same list. This is the default option.

NONE
The first connection in a process reading a CCDT creates a list of applicable definitions. All connections in a process select an applicable definition based on the weighting with any applicable CLNTWGHT(0) definitions selected first in alphabetical order. For C, C++ and .NET (including fully managed .NET) clients the list is updated if the CCDT has been modified since the list was created.

For example, suppose we had the following definitions in the CCDT:

```
CHLNAME(A) QMNAME(QM1) CLNTWGHT(3)
CHLNAME(B) QMNAME(QM1) CLNTWGHT(4)
CHLNAME(C) QMNAME(QM1) CLNTWGHT(4)
```

The first connection in a process creates its own ordered list based on the weightings. So it might, for example, create the ordered list CHLNAME(B), CHLNAME(A), CHLNAME(C).

For AFFINITY(PREFFERED) each connection in the process attempts to connect using CHLNAME(B). If a connection is unsuccessful the definition is moved to the end of the list which now becomes CHLNAME(A), CHLNAME(C), CHLNAME(B). Each connection in the process then attempts to connect using CHLNAME(A).

For AFFINITY(NONE) each connection in the process attempts to connect using one of the three definitions selected at random based on the weightings.

When sharing conversations is enabled with a non zero channel weighting and AFFINITY(NONE) multiple connections in a process using the same queue manager name may connect using different applicable definitions rather than sharing an existing channel instance.

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 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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

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

The maximum batch size actually used is the lowest of the following:

- The BATCHSZ of the sending channel
- The BATCHSZ of the receiving channel
- On z/OS, three less than the maximum number of uncommitted messages allowed at the sending queue manager (or one if this value is zero or less). On platforms other than z/OS, the maximum number of uncommitted messages allowed at the sending queue manager (or one if this value is zero or less).
- On z/OS, three less than the maximum number of uncommitted messages allowed at the receiving queue manager (or one if this value is

zero or less). On platforms other than z/OS, 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

RCVR Receiver channel

RQSTR
Requester channel

CLNTCONN
Client-connection channel

SVRCONN
Server-connection channel

CLUSSDR
Cluster-sender channel (valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS)

CLUSRCVR
Cluster-receiver channel (valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS)

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

CLNTWGHT

The client channel weighting attribute is used so client channel definitions can be selected at random based on their weighting when more than one suitable definition is available. Specify a value in the range 0 - 99. The default is the special value 0.

The special value 0 indicates that no random load balancing is performed and applicable definitions are selected in alphabetical order. To enable random load balancing the value should be in the range 1 through 99 where 1 is the lowest weighting and 99 is the highest.

When a client issues an MQCONN with queue manager name "*<name>" and more than one suitable definition is available in the CCDT the choice of definition to use is randomly selected based on the weighting with any applicable CLNTWGHT(0) definitions selected first in alphabetical order. The distribution is not guaranteed.

For example, suppose we had the following two definitions in the CCDT:

```
CHLNAME(TO.QM1) CHLTYPE(CLNTCONN) QMNAME(GRP1) CONNAME(address1) CLNTWGHT(2)
CHLNAME(TO.QM2) CHLTYPE(CLNTCONN) QMNAME(GRP1) CONNAME(address2) CLNTWGHT(4)
```

A client MQCONN with queue manager name ″*GRP1″ would choose one of the two definitions based on the weighting of the channel definition. (A random integer would be generated between 1 and 6. If the integer was in the range 1 through 2 address1 would be used otherwise address2 would be used). If this connect was unsuccessful the client would then use the other definition.

The CCDT may contain applicable definitions with both zero and non zero weighting. In this situation the definitions with zero weightings are chosen first and in alphabetical order. If these connects are unsuccessful the definitions with non zero weighting are chosen based on their weighting.

For example, suppose we had the following four definitions in the CCDT:

```
CHLNAME(TO.QM1) CHLTYPE(CLNTCONN) QMNAME(GRP1) CONNAME(address1) CLNTWGHT(1)
CHLNAME(TO.QM2) CHLTYPE(CLNTCONN) QMNAME(GRP1) CONNAME(address2) CLNTWGHT(2)
CHLNAME(TO.QM3) CHLTYPE(CLNTCONN) QMNAME(GRP1) CONNAME(address3) CLNTWGHT(0)
CHLNAME(TO.QM4) CHLTYPE(CLNTCONN) QMNAME(GRP1) CONNAME(address4) CLNTWGHT(0)
```

A client MQCONN with queue manager name ″*GRP1″ would first choose definition ″TO.QM3″. If this connect was unsuccessful the client would then choose definition ″TO.QM4″. If this connection was also unsuccessful the client would then randomly choose one of the remaining two definitions based on their weighting.

CLNTWGHT support is added for all supported transport protocols.

CLUSNL(*nlname***)**
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, HP OpenVMS, HP-UX, Linux i5/OS, Solaris, Windows, and z/OS.

CLUSTER(*clustername***)**
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 or CLUSRCVR. 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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows and z/OS.

CLWLPRTY(*integer***)**
Specifies the priority of the channel for the purposes of cluster workload distribution. The value must be in the range zero through 9 where zero is the lowest priority and 9 is the highest. The default value is zero.

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

For more information about this attribute, see WebSphere MQ Queue Manager Clusters.

CLWLRANK(*integer***)**
Specifies the rank of the channel for the purposes of cluster workload

distribution. The value must be in the range zero through 9 where zero is the lowest rank and 9 is the highest. The default value is zero.

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

For more information about this attribute, see WebSphere MQ Queue Manager Clusters.

CLWLWGHT(*integer***)**
Specifies the weighting to be applied to the channel for the purposes of cluster workload distribution so that the proportion of messages sent down the channel can be controlled. The value must be in the range 1 through 99 where 1 is the lowest rank and 99 is the highest. The default value is 50.

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

For more information about this attribute, see WebSphere MQ Queue Manager Clusters.

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.

COMPHDR
The list of header data compression techniques supported by the channel. For sender, server, cluster-sender, cluster-receiver, and client-connection channels, the values specified are in order of preference with the first compression technique supported by the remote end of the channel being used.

The channel's mutually supported compression techniques are passed to the sending channel's message exit where the compression technique used can be altered on a per message basis. Compression alters the data passed to send and receive exits.

NONE
No header data compression is performed. This is the default value.

SYSTEM
Header data compression is performed.

COMPMSG

The list of message data compression techniques supported by the channel. For sender, server, cluster-sender, cluster-receiver, and client-connection channels, the values specified are in order of preference with the first compression technique supported by the remote end of the channel being used.

The channel's mutually supported compression techniques are passed to the sending channel's message exit where the compression technique used can be altered on a per message basis. Compression will alter the data passed to send and receive exits.

NONE
No message data compression is performed. This is the default value.

RLE Message data compression is performed using run-length encoding.

ZLIBFAST
Message data compression is performed using ZLIB encoding with speed prioritized.

ZLIBHIGH
Message data compression is performed using ZLIB encoding with compression prioritized.

ANY Any compression technique supported by the queue manager can be used. This is only valid for receiver, requester, and server-connection channels.

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 HP OpenVMS.

LU 6.2

- On HP 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:

Form	Example
luname	IGY12355
luname/TPname	IGY12345/APING
luname/TPname/modename	IGY12345/APINGD/#INTER

  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 HP OpenVMS, i5/OS, 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 WebSphere MQ Intercommunication 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('$PPPP.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('$PPPP.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('$PPPP.#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('$PPPP.#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.

If the CONNAME is a hostname, the hostname is resolved to an IP address.

The IP stack used for communication depends on the value specified for CONNAME **and** the value specified for LOCLADDR. See LOCLADDR for information about how this is resolved.

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 Compaq NonStop Kernel the connection name can take an additional field at the start of the value, specifying the name of a specific Guardian TCP/IP server process to be used for the channel.

On AIX, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS, 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 IPv4 address of the system. If the system does not have an IPv4 address, the current IPv6 address of the system is used.

Note: If you are using clustering between IPv6–only and IPv4–only queue managers, do not specify an IPv6 network address as the CONNAME for CLUSRCVR channels. A queue manager that is capable only of IPv4 communication is unable to start a cluster sender channel definition that specifies the CONNAME in IPv6 hexadecimal form. Consider, instead, using hostnames in a heterogeneous IP environment.

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.

DEFCDISP
Specifies the default channel disposition of the channel.

PRIVATE
The intended disposition of the channel is as a PRIVATE channel.

This is the default value.

FIXSHARED
The intended disposition of the channel is as a FIXSHARED channel.

SHARED
The intended disposition of the channel is as a SHARED channel.

This parameter does not apply to channels with a channel type (CHLTYPE) of CLNTCONN, 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.

This parameter is valid only for channels with a channel type (CHLTYPE) of SVRCONN (on z/OS only), SDR, SVR, CLUSSDR, CLUSRCVR.

For SVRCONN channels on z/OS using the TCP protocol, this is the minimum time in seconds for which the SVRCONN instance remains active without any communication from its partner client. A value of zero disables this disconnect processing. The SVRCONN inactivity interval only applies between MQ API calls from a client, so no client is disconnected during an extended MQGET with wait call. This attribute is ignored for SVRCONN channels using protocols other than TCP.

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.

  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.

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.

This parameter is valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows and z/OS.

KAINT(*integer***)**

The value passed to the communications stack for KeepAlive timing for this channel.

For this attribute to have any effect, TCP/IP keepalive must be enabled both in the queue manager and in TCP/IP. On z/OS, you enable TCP/IP keepalive in the queue manager by issuing the ALTER QMGR TCPKEEP(YES) command; if the TCPKEEP queue manager parameter is NO, the value is ignored and the KeepAlive facility is not used. On other platforms, TCP/IP keepalive is enabled when the KEEPALIVE=YES parameter is specified in the TCP stanza in the distributed queuing configuration file, qm.ini, or through the WebSphere MQ Explorer.

Keepalive must also be switched on within TCP/IP itself. Refer to your TCP/IP documentation for information about configuring keepalive. On AIX, use the 'no' command. On HP-UX, use the 'nettune' command. On Windows, edit the registry. On z/OS, update your TCP/IP PROFILE data set and add or change the INTERVAL parameter in the TCPCONFIG section.

Although this parameter is available on all platforms, its setting is implemented only on z/OS. On platforms other than z/OS, you can access and modify the parameter, but it is only stored and forwarded; there is no functional implementation of the parameter. This is useful in a clustered environment where a value set in a cluster-receiver channel definition on Solaris, for example, flows to (and is implemented by) z/OS queue managers that are in, or join, the cluster.

On platforms other than z/OS, if you need the functionality provided by the KAINT parameter, use the Heartbeat Interval (HBINT) parameter, as described in HBINT.

(*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 INTERVAL 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 INTERVAL statement in the TCP profile configuration data set.

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

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 z/OS, the queue manager searches page set zero for an object with the name you specify and a disposition of QMGR or COPY. The disposition of the LIKE object is not copied to the object and channel type you are defining.

Note:

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 or if you want to force a channel to use an IPv4 or IPv6 stack on a dual–stack system, or use a dual mode stack on a single–stack system.

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 IPv4 dotted decimal, IPv6 hexadecimal notation, or alphanumeric hostname form, and `low-port` and `high-port` are port numbers enclosed in parentheses. All are optional.

Table 5 shows how the LOCLADDR parameter can be used:

Table 5. Examples of how the LOCLADDR parameter can be used

LOCLADDR	Meaning
9.20.4.98	Channel binds to this address locally
9.20.4.98(1000)	Channel binds to this address and port 1000 locally
9.20.4.98(1000,2000)	Channel binds to this address and uses a port in the range 1000 to 2000 locally
(1000)	Channel binds to port 1000 locally
(1000,2000)	Channel binds to port in range 1000 to 2000 locally

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

On cluster-sender channels, the IP address and port to which the outbound channel binds to is a merger of the IP address as defined in the LOCLADDR parameter and the port range from the cluster cache. If there is no port range in the cache, the port range defined in the LOCLADDR parameter is used.

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.

When a channel is started, the values specified for CONNAME and LOCLADDR determine the IP stack to be used for communication. See Table 6 for further details on how the IP stack is determined. For further guidance, see the WebSphere MQ Intercommunication manual.

If the appropriate IP stack for the local address is not installed or configured, the channel fails to start and an error message is generated.

Table 6. How the IP stack to be used for communication is determined

Protocols supported	CONNAME	LOCLADDR	Action of channel
IPv4 only	IPv4 address[1]		Channel binds to IPv4 stack
	IPv6 address[2]		Channel fails to resolve CONNAME
	IPv4 and 6 hostname[3]		Channel binds to IPv4 stack
	IPv4 address	IPv4 address	Channel binds to IPv4 stack
	IPv6 address	IPv4 address	Channel fails to resolve CONNAME
	IPv4 and 6 hostname	IPv4 address	Channel binds to IPv4 stack
	Any address[4]	IPv6 address	Channel fails to resolve LOCLADDR
	IPv4 address	IPv4 and 6 hostname	Channel binds to IPv4 stack
	IPv6 address	IPv4 and 6 hostname	Channel fails to resolve CONNAME
	IPv4 and 6 hostname	IPv4 and 6 hostname	Channel binds to IPv4 stack

Table 6. How the IP stack to be used for communication is determined (continued)

Protocols supported	CONNAME	LOCLADDR	Action of channel
IPv4 and IPv6	IPv4 address		Channel binds to IPv4 stack
	IPv6 address		Channel binds to IPv6 stack
	IPv4 and 6 hostname		Channel binds to stack determined by IPADDRV
	IPv4 address	IPv4 address	Channel binds to IPv4 stack
	IPv6 address	IPv4 address	Channel fails to resolve CONNAME
	IPv4 and 6 hostname	IPv4 address	Channel binds to IPv4 stack
	IPv4 address	IPv6 address	Channel maps CONNAME to IPv6[5]
	IPv6 address	IPv6 address	Channel binds IPv6 stack
	IPv4 and 6 hostname	IPv6 address	Channel binds IPv6 stack
	IPv4 address	IPv4 and 6 hostname	Channel binds to IPv4 stack
	IPv6 address	IPv4 and 6 hostname	Channel binds to IPv6 stack
	IPv4 and 6 hostname	IPv4 and 6 hostname	Channel binds to stack determined by IPADDRV
IPv6 only	IPv4 address		Channel maps CONNAME to IPv6[5]
	IPv6 address		Channel binds to IPv6 stack
	IPv4 and 6 hostname		Channel binds to IPv6 stack
	Any address	IPv4 address	Channel fails to resolve LOCLADDR
	IPv4 address	IPv6 address	Channel maps CONNAME to IPv6[5]
	IPv6 address	IPv6 address	Channel binds to IPv6 stack
	IPv4 and 6 hostname	IPv6 address	Channel binds to IPv6 stack
	IPv4 address	IPv4 and 6 hostname	Channel maps CONNAME to IPv6[5]
	IPv6 address	IPv4 and 6 hostname	Channel binds to IPv6 stack
	IPv4 and 6 hostname	IPv4 and 6 hostname	Channel binds to IPv6 stack

Table 6. How the IP stack to be used for communication is determined (continued)

Protocols supported	CONNAME	LOCLADDR	Action of channel
Notes: 1. IPv4 address. An IPv4 hostname that only resolves to an IPv4 network address or a specific dotted notation IPv4 address, for example `1.2.3.4`. This applies to all occurrences of 'IPv4 address' in this table. 2. IPv6 address. An IPv6 hostname that only resolves to an IPv6 network address or a specific hexadecimal notation IPv6 address, for example `4321:54bc`. This applies to all occurrences of 'IPv6 address' in this table. 3. IPv4 and 6 hostname. A hostname that resolves to both IPv4 and IPv6 network addresses. This applies to all occurrences of 'IPv4 and 6 hostname' in this table. 4. Any address. IPv4 address, IPv6 address, or IPv4 and 6 hostname. This applies to all occurrences of 'Any address' in this table. 5. Maps IPv4 CONNAME to IPv4 mapped IPv6 address. IPv6 stack implementations that do not support IPv4 mapped IPv6 addressing fail to resolve the CONNAME. Mapped addresses may require protocol translators in order to be used. The use of mapped addresses is not recommended.			

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.

MAXINST(*integer***)**
The maximum number of simultaneous instances of an individual server-connection channel that can be started.

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

A value of zero prevents all client access on this channel.

If the value of this parameter is reduced to a number that is less than the number of instances of the server-connection channel that are currently running, then those running instances are not affected. However, new instances cannot start until sufficient existing instances have ceased to run so that the number of currently running instances is less than the value of this parameter.

Note that on z/OS, without the Client Attachment feature installed, only a maximum of five instances are allowed on the channel named SYSTEM.ADMIN.SVRCONN. If MAXINST is set to a larger number than five, it is interpreted as zero without the CAF installed.

This parameter is valid only for channels with a channel type (CHLTYPE) of SVRCONN.

MAXINSTC(*integer***)**
The maximum number of simultaneous individual server-connection channels that can be started from a single client. In this context, connections that originate from the same remote network address are regarded as coming from the same client.

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

A value of zero prevents all client access on this channel.

If the value of this parameter is reduced to a number that is less than the number of instances of the server-connection channel that is currently running from individual clients, then those running instances are not affected. However, new instances from those clients cannot start until sufficient instances have ceased to run that the number of running instances is less than the value of this parameter.

Note that on z/OS, without the Client Attachment feature installed, only a maximum of five instances are allowed on the channel named SYSTEM.ADMIN.SVRCONN.

This parameter is valid only for channels with a channel type (CHLTYPE) of SVRCONN.

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, HP OpenVMS, HP-UX, Linux, i5/OS, 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 (104 857 600 bytes).

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

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 on an outbound MCA channel should run as a thread or a process.

PROCESS
The message channel agent runs as a separate process. This is the default value.

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, HP OpenVMS, HP-UX, Linux, i5/OS, 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.
- 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.

MONCHL
Controls the collection of online monitoring data for channels:

QMGR
Collect monitoring data according to the setting of the queue manager parameter MONCHL. This is the default value.

OFF Monitoring data collection is turned off for this channel.

LOW If the value of the queue manager's MONCHL parameter is not NONE, online monitoring data collection is turned on, with a low rate of data collection, for this channel.

MEDIUM
If the value of the queue manager's MONCHL parameter is not NONE, online monitoring data collection is turned on, with a moderate rate of data collection, for this channel.

HIGH If the value of the queue manager's MONCHL parameter is not NONE, online monitoring data collection is turned on, with a high rate of data collection, for this channel.

Changes to this parameter take effect only on channels started after the change occurs.

For cluster channels, the value of this parameter is not replicated in the repository and, therefore, not used in the auto-definition of cluster-sender channels. For auto-defined cluster-sender channels, the value of this parameter is taken from the queue manager's attribute MONACLS. This value may then be overridden in the channel auto-definition exit.

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.

MREXIT(*string***)**
Channel message-retry exit name.

The format and maximum length of the name is the same as for MSGEXIT, however you can only specify one message-retry exit.

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

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.

The default value is 10, except for z/OS channels that you are migrating from an earlier release which take a value of zero as default.

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

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.

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, HP OpenVMS, HP-UX, Linux, 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 i5/OS, 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.

Note: This parameter is accepted but ignored for server-connection and client-connection channels.

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.

If this name is nonblank, on other platforms, 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, HP OpenVMS, HP-UX, Linux, 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 i5/OS, 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 accepted but ignored, because message exits are not invoked for such channels.

The format and maximum length of the name depends on the environment:

- On HP OpenVMS and UNIX systems, it is of the form:

  ```
  libraryname(functionname)
  ```

  The maximum length of the string is 128 characters.
- On 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 i5/OS, it is of the form:

  ```
  progname 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.

NPMSPEED
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.

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.

PROPCTL
Property control attribute.

This parameter is applicable to Sender, Server, Cluster Sender and Cluster Receiver channels.

This parameter is optional.

Permitted values are:

ALL All properties of the message are included with the message when it is sent to the remote queue manager. The properties, except those in the message descriptor (or extension), are placed in one or more MQRFH2 headers in the message data.

COMPAT
If the message contains a property with a prefix of mcd., jms., usr. or mqext., all message properties are delivered to the application in an MQRFH2 header. Otherwise all properties of the message, except those contained in the message descriptor (or extension), are discarded and are no longer accessible to the application.

This is the default value; it allows applications which expect JMS related properties to be in an MQRFH2 header in the message data to continue to work unmodified.

NONE
All properties of the message, except those in the message

descriptor (or extension), are removed from the message before the message is sent to the remote queue manager.

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, on z/OS only, 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).

QSGDISP	ALTER	DEFINE
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.	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.

QSGDISP	ALTER	DEFINE
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 attempt to refresh local copies on page set zero: `DEFINE CHANNEL(channel-name) CHLTYPE(type) REPLACE QSGDISP(COPY)` The ALTER for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.	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 attempt to make or refresh local copies on page set zero: `DEFINE CHANNEL(channe-name) CHLTYPE(type) REPLACE QSGDISP(COPY)` The DEFINE for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.
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.	Not permitted.
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.

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, HP OpenVMS, HP-UX, Linux, 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 i5/OS, 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.

RCVEXIT(*string***)**

Channel receive exit name.

If this name is nonblank, on platforms other than Compaq NSK, 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, HP OpenVMS, HP-UX, Linux, 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 i5/OS, 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.

If this name is nonblank, on platforms other than Compaq NSK, 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.
- At initialization and termination of the channel.

The format and maximum length of the name is the same as for MSGEXIT but only one name is allowed.

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, HP OpenVMS, HP-UX, Linux, 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 i5/OS, 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.

If this name is nonblank, on platforms other than Compaq NSK, 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, HP OpenVMS, HP-UX, Linux, 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 i5/OS, 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 (CHLTYPE) of SDR, SVR, RCVR, RQSTR, CLUSSDR, or CLUSRCVR.

SHARECNV(*integer***)**
Specifies the maximum number of conversations that can be sharing each TCP/IP channel instance. A SHARECNV value of:

1 Specifies no sharing of conversations over a TCP/IP channel instance. Client heartbeating is available whether in an MQGET call or not. Read ahead and client asynchronous consume are also available, and channel quiescing is more controllable.

0 Specifies no sharing of conversations over a TCP/IP channel

instance. The channel instance runs in a mode prior to that of WebSphere MQ Version 7.0, with regard to:

- Administrator stop-quiesce
- Heartbeating
- Read ahead
- Client asynchronous consume

The value must be in the range zero through 999 999 999 and the default value is 10.

This parameter is valid only for channels with a channel type (CHLTYPE) of CLNTCONN or SVRCONN. If the client-connection SHARECNV value does not match the server-connection SHARECNV value, the lower of the two values is used. This parameter is ignored for channels with a transport type (TRPTYPE) other than TCP.

All of the conversations on a socket are received by the same thread.

High SHARECNV limits have the advantage of reducing queue manager thread usage. However, if a large number of conversations sharing a socket are all busy, there is a possibility of delays as the conversations contend with one another to use the receiving thread. In this situation a lower SHARECNV value is better.

You must optimize the SHARECNV value for your enterprise bearing in mind the preceding points.

The number of shared conversations does not contribute to the MAXINST or MAXINSTC totals.

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 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 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 7.

On WebSphere MQ for z/OS and for i5/OS, you can also specify the 2-digit hexadecimal code of a CipherSpec, whether or not it appears in Table 7.

On i5/OS, installation of AC3 is a prerequisite of the use of SSL.

Table 7. CipherSpecs that can be used with WebSphere MQ SSL support

CipherSpec name	Hash algorithm	Encryption algorithm	Encryption bits	FIPS on Windows and UNIX platforms [1]
NULL_MD5 **Note:** Available on all platforms	MD5	None	0	No
NULL_SHA **Note:** Available on all platforms	SHA-1	None	0	No
RC4_MD5_EXPORT **Note:** Available on all platforms	MD5	RC4	40	No
RC4_MD5_US **Note:** Available on all platforms	MD5	RC4	128	No
RC4_SHA_US **Note:** Available on all platforms	SHA-1	RC4	128	No
RC2_MD5_EXPORT **Note:** Available on all platforms	MD5	RC2	40	No

Table 7. CipherSpecs that can be used with WebSphere MQ SSL support (continued)

CipherSpec name	Hash algorithm	Encryption algorithm	Encryption bits	FIPS on Windows and UNIX platforms [1]
DES_SHA_EXPORT **Note:** Available on all platforms	SHA-1	DES	56	No
RC4_56_SHA_EXPORT1024 **Note:** 1. Not available for z/OS or i5/OS 2. Specifies a 1024–bit handshake key size	SHA-1	RC4	56	No
DES_SHA_EXPORT1024 **Note:** 1. Not available for z/OS or i5/OS 2. Specifies a 1024–bit handshake key size	SHA-1	DES	56	No
TRIPLE_DES_SHA_US **Note:** Not available for i5/OS	SHA-1	3DES	168	No
TLS_RSA_WITH_AES_128_CBC_SHA **Note:** 1. Not available for i5/OS 2. The protocol used is TLS rather than SSL	SHA-1	AES	128	Yes
TLS_RSA_WITH_AES_256_CBC_SHA **Note:** 1. Not available for i5/OS 2. The protocol used is TLS rather than SSL	SHA-1	AES	256	Yes
AES_SHA_US **Note:** Available on i5/OS only	SHA-1	AES	128	No
TLS_RSA_WITH_DES_CBC_SHA **Note:** 1. Not available for z/OS or i5/OS 2. The protocol used is TLS rather than SSL	SHA-1	DES	56	No[2]
TLS_RSA_WITH_3DES_EDE_CBC_SHA **Note:** 1. Not available for z/OS or i5/OS 2. The protocol used is TLS rather than SSL	SHA-1	3DES	168	Yes
FIPS_WITH_DES_CBC_SHA **Note:** Available only on Windows and UNIX platforms	SHA-1	DES	56	No[3]
FIPS_WITH_3DES_EDE_CBC_SHA **Note:** Available only on Windows and UNIX platforms	SHA-1	3DES	168	Yes

Note:

1. Is the CipherSpec FIPS-certified on a FIPS-certified platform? See "CipherSuites and CipherSpecs" in the *WebSphere MQ: Security* manual for an explanation of FIPS.
2. This CipherSpec was FIPS 140-2 certified prior to 19th May 2007.
3. This CipherSpec was FIPS 140-2 certified prior to 19th May 2007. The name FIPS_WITH_DES_CBC_SHA is historical and reflects the fact that this CipherSpec was previously FIPS-compliant.

When you request a personal certificate, you specify a key size for the public and private key pair. The key size that is used during the SSL handshake can depend on the size stored in the certificate and on the CipherSpec:

- On UNIX systems, Windows systems, and z/OS, when a CipherSpec name includes _EXPORT, the maximum handshake key size is 512 bits. If either of the certificates exchanged during the SSL handshake has a key size greater than 512 bits, a temporary 512-bit key is generated for use during the handshake.
- On UNIX and Windows systems, when a CipherSpec name includes _EXPORT1024, the handshake key size is 1024 bits.
- Otherwise the handshake key size is the size stored in the certificate.

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 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:

CN	common name
T	title
OU	organizational unit name
O	organization name
L	locality name
S, ST, or SP	state or province name
C	country

WebSphere MQ will only accept upper case letters for the attribute types.

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 be defined in descending hierarchical order. For example, if the flowed certificate's Distinguished Name contains the OUs `OU=Large Unit, OU=Medium Unit, OU=Small Unit`, specifying the following SSLPEER values will work:

```
('OU=Large Unit,OU=Medium Unit')
('OU=*,OU=Medium Unit,OU=Small Unit')
('OU=*,OU=Medium Unit')
```

but specifying the following SSLPEER values will fail:

```
('OU=Medium Unit,OU=Small Unit')
('OU=Large Unit,OU=Small Unit')
('OU=Medium Unit')
('OU=Small Unit, Medium Unit, Large Unit')
```

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, i5/OS, and UNIX platforms, and 256 bytes on z/OS.

STATCHL

Controls the collection of statistics data for channels:

QMGR
The value of the queue manager's STATCHL parameter is inherited by the channel. This is the default value.

OFF Statistics data collection is turned off for this channel.

LOW If the value of the queue manager's STATCHL parameter is not NONE, statistics data collection is turned on, with a low rate of data collection, for this channel.

MEDIUM
If the value of the queue manager's STATCHL parameter is not NONE, statistics data collection is turned on, with a moderate rate of data collection, for this channel.

HIGH If the value of the queue manager's STATCHL parameter is not NONE, statistics data collection is turned on, with a high rate of data collection, for this channel.

Changes to this parameter take effect only on channels started after the change occurs.

For cluster channels, the value of this parameter is not replicated in the repository and used in the auto-definition of cluster-sender channels. For auto-defined cluster-sender channels, the value of this parameter is taken from the queue manager's attribute STATACLS. This value may then be overridden in the channel auto-definition exit.

This parameter is valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

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.

This should be set to the SNA transaction program name, on other platforms, 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, i5/OS, 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 HP OpenVMS)

LU62 SNA LU 6.2

NETBIOS
NetBIOS (supported only on 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 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

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.

On the receiving end, if passwords are kept in encrypted format and the LU 6.2 software is using a different encryption method, an attempt to start the channel fails with invalid security details. You can avoid this by modifying the receiving SNA configuration to either:

- Turn off password substitution, or
- Define a security user ID and password.

XMITQ(*string***)**

Transmission queue name.

The name of the queue from which messages are retrieved. See "Rules for naming WebSphere MQ objects" on page 5.

This parameter is valid only for channels with a channel type (CHLTYPE) of SDR or SVR. For these channel types this parameter is required.

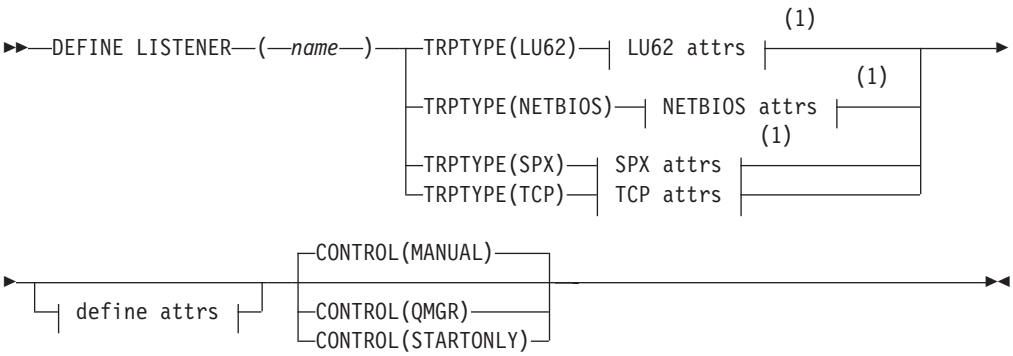
DEFINE LISTENER

Use the MQSC command DEFINE LISTENER to define a new WebSphere MQ listener definition, and set its parameters.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	

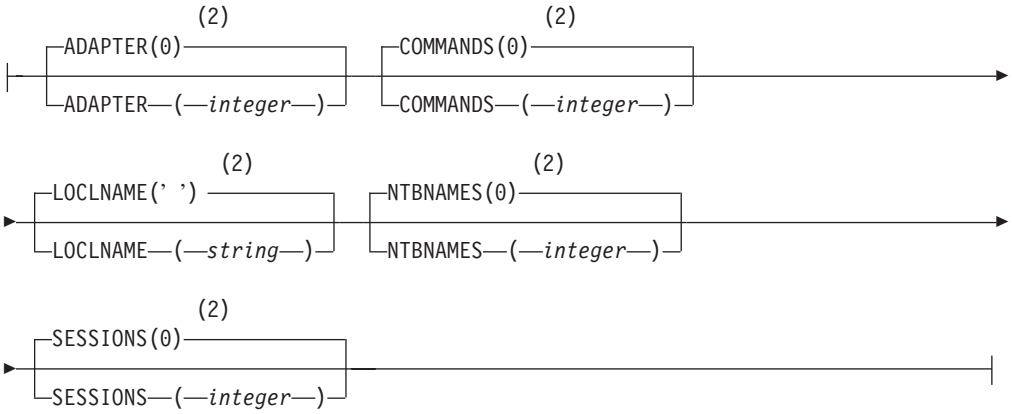
Synonym: DEF LSTR

DEFINE LISTENER

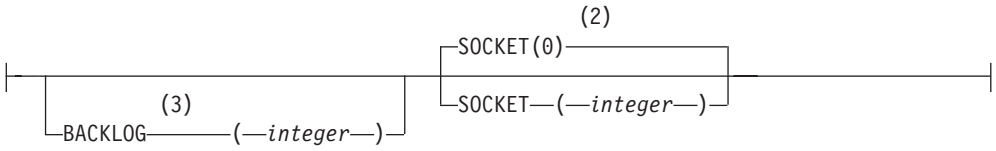


LU62 attrs:

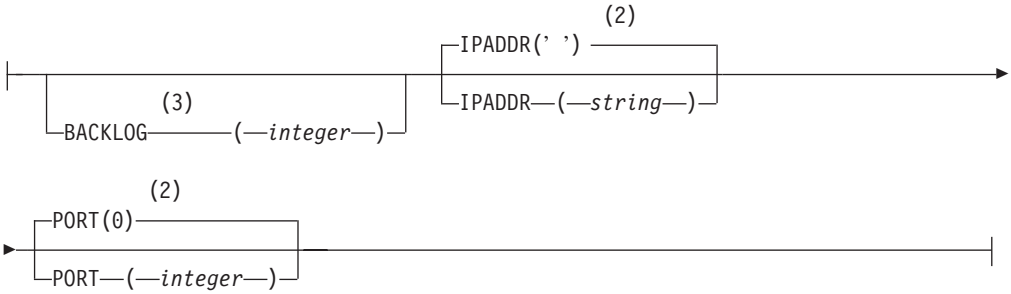
NETBIOS attrs:



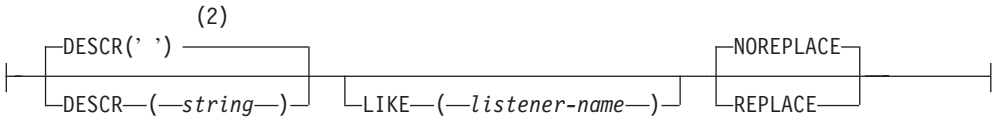
SPX attrs:



TCP attrs:



Define attrs:



Notes:

1 Valid only on Windows.

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

3 When the BACKLOG attribute is left unchanged or when it is explicitly set to zero, the attribute is stored as zero by default in the listener object created by the DEFINE LISTENER command. However, when the listener is started, the default backlog value takes effect. For information about the default value of the BACKLOG attribute, see WebSphere MQ Intercommunication.

Parameter descriptions for DEFINE LISTENER

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

- The **LIKE** parameter applies only to the DEFINE LISTENER command.
- The **NOREPLACE** and **REPLACE** parameter applies only to the DEFINE LISTENER command.

(listener-name)

Name of the WebSphere MQ listener definition (see "Rules for naming WebSphere MQ objects" on page 5). This is required.

The name must not be the same as any other listener definition currently defined on this queue manager (unless REPLACE is specified).

ADAPTER(*integer***)**
The adapter number on which NetBIOS listens. By default, the listener uses adapter 0. This parameter is valid only on Windows when TRPTYPE is NETBIOS.

BACKLOG(*integer***)**
The number of concurrent connection requests that the listener supports. If you do not specify a value, the default value defined by the protocol is used.

COMMANDS(*integer***)**
The number of commands that the listener can use. If you do not specify a value, the default value defined by the protocol is used. This parameter is valid only on Windows when TRPTYPE is NETBIOS.

CONTROL(*string***)**
Specifies how the listener is to be started and stopped.:

MANUAL
The listener is not to be started automatically or stopped automatically. It is to be controlled by use of the START LISTENER and STOP LISTENER commands. This is the default value.

QMGR
The listener being defined is to be started and stopped at the same time as the queue manager is started and stopped.

STARTONLY
The listener is to be started at the same time as the queue manager is started, but is not requested to stop when the queue manager is stopped.

DESCR(*string***)**
Plain-text comment. It provides descriptive information about the listener when an operator issues the DISPLAY LISTENER command (see "DISPLAY LISTENER" on page 390).

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.

IPADDR(*string***)**
IP address for the listener specified in IPv4 dotted decimal, IPv6 hexadecimal notation, or alphanumeric hostname form. If you do not specify a value for this parameter, the listener listens on all configured IPv4 and IPv6 stacks.

LIKE(*listener-name***)**
The name of a listener, whose parameters are used to model this definition.

This parameter applies only to the DEFINE LISTENER 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 listeners on this queue manager. This is equivalent to specifying:

```
LIKE(SYSTEM.DEFAULT.LISTENER)
```

A default listener is provided but it can be altered by the installation of the default values required. See “Rules for naming WebSphere MQ objects” on page 5.

LOCLNAME(*string***)**
The NetBIOS local name that the listener uses. This parameter is valid only on Windows when TRPTYPE is NETBIOS.

NTBNAMES(*integer***)**
The number of names that the listener can use. If you do not specify a value, the default value defined by the protocol is used. This parameter is valid only on Windows when TRPTYPE is NETBIOS.

PORT(*integer***)**
The port number for TCP/IP. This is valid only when TRPTYPE is TCP.

REPLACE and NOREPLACE
Whether the existing definition is to be replaced with this one. This is optional. The default is NOREPLACE.

This parameter applies only to the DEFINE LISTENER 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.

SESSIONS(*integer***)**
The number of sessions that the listener can use. If you do not specify a value, the default value defined for the protocol is used. This parameter is valid only on Windows when TRPTYPE is NETBIOS.

SOCKET(*integer***)**
The SPX socket on which to listen. This is valid only if TRPTYPE is SPX.

TPNAME(*string***)**
The LU 6.2 transaction program name (maximum length 64 characters). This parameter is valid only on Windows when TRPTYPE is LU62.

TRPTYPE(*string***)**
The transmission protocol to be used:

LU62
SNA LU 6.2. This is valid only on Windows.

NETBIOS
NetBIOS. This is valid only on Windows.

SPX
Sequenced packet exchange. This is valid only on Windows.

TCP
TCP/IP.

DEFINE LOG

Use the MQSC command DEFINE LOG to add a new active log data set in the ring of active logs.

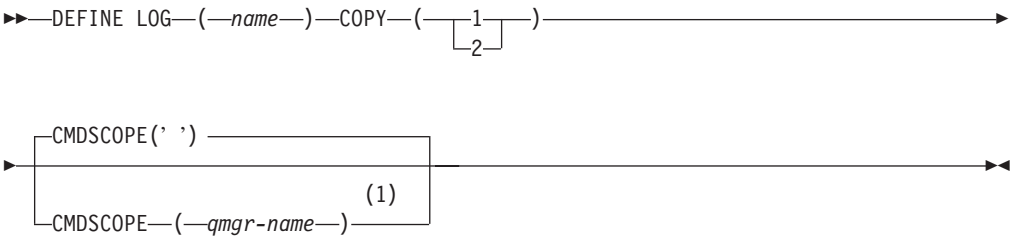
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

The named data set is dynamically allocated to the running queue manager, added to either the COPY1 or COPY2 active log and the BSDS updated with the information so it is retained over a queue manager restart. The data set is added to the active log ring in a position such that it will be the next active log used when the current active log fills and an active log switch occurs.

Synonym: DEF LOG

DEFINE LOG



Notes:

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

Parameter descriptions for DEFINE LOG

(name) The name of the new log data set. This is required and is the name of a VSAM linear data set which will have already been defined by Access Method Services (and, optionally, formatted by utility CSQJUFMT). This is allocated dynamically to the queue manager.

The maximum length of the string is 44 characters. The string must conform to z/OS data set naming conventions.

COPY

Specifies the number of an active log ring to which to add the new log data set. This is either 1 or 2 and 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 it was entered, only if you are using a shared queue environment and if the command server is enabled.

Usage note for DEFINE LOG

- If a log data set has to be added because there is no more log space and the queue manager is waiting, you must issue the command from the z/OS console, and not through the command server.

DEFINE MAXSMSGS

Use the MQSC command DEFINE MAXSMSGS to define the maximum number of messages that a task can get or put within a single unit of recovery.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

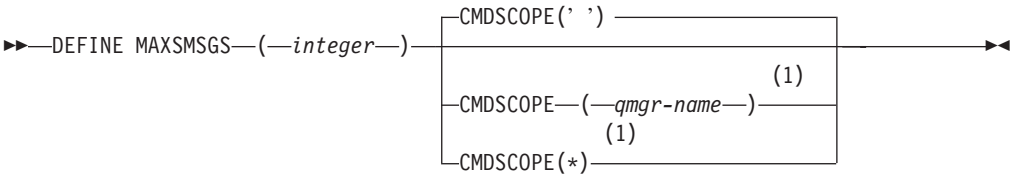
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

1. This command is valid only on z/OS and is retained for compatibility with earlier releases, 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



Notes:

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

Parameter descriptions for DEFINE MAXSMSGS

(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.

' '
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 NAMELIST

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

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

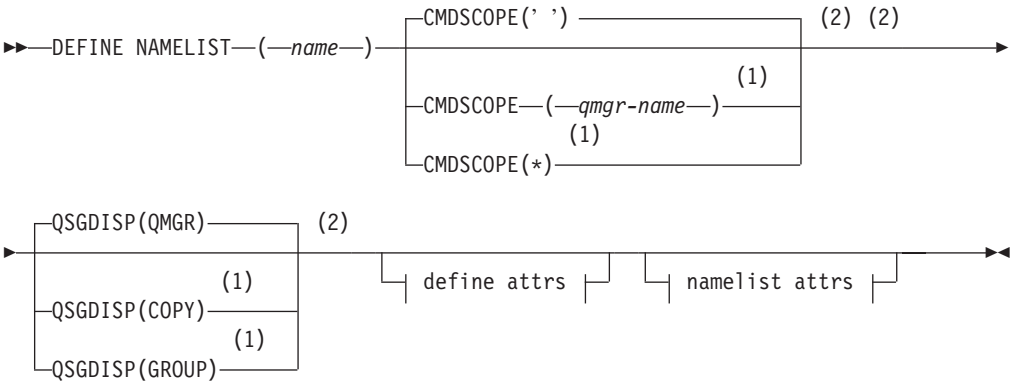
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

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

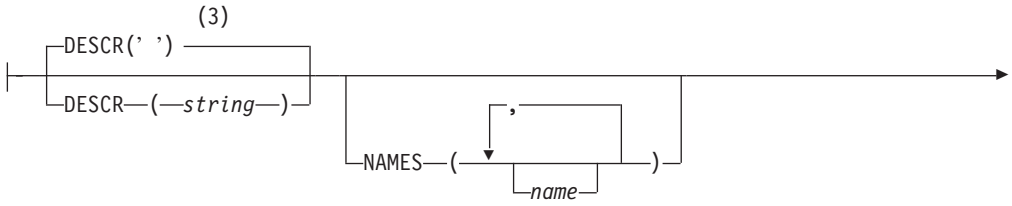
Synonym: DEF NL

DEFINE NAMELIST



Define attrs:

Namelist attrs:



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

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

Parameter descriptions for DEFINE NAMELIST

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.

(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 5.

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 401).

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.

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 5.

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

Note:

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).

QSGDISP	ALTER	DEFINE
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.	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.

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 attempt to refresh local copies on page set zero:
	`DEFINE NAMELIST(name)`
	`REPLACE QSGDISP(COPY)`
	
	The ALTER for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.
		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 attempt to make or refresh local copies on page set zero:
		`DEFINE NAMELIST(name)`
		`REPLACE QSGDISP(COPY)`
		
		The DEFINE for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.

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.	Not permitted.
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.

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.

DEFINE PROCESS

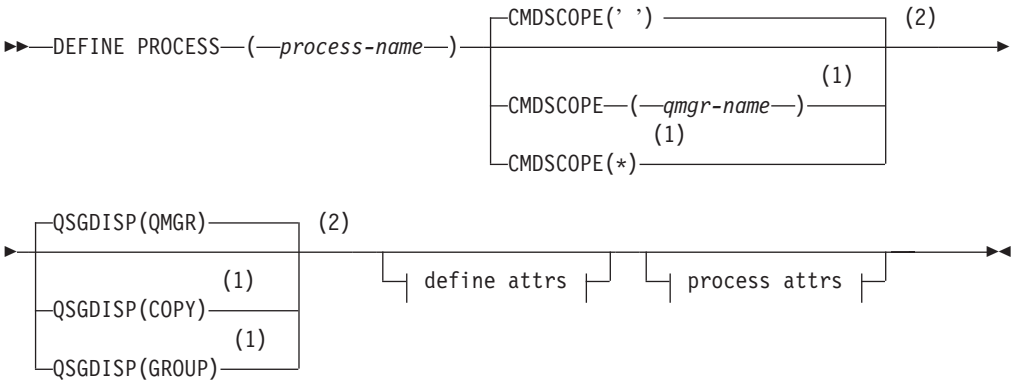
Use the MQSC command DEFINE PROCESS to define a new WebSphere MQ process definition, and set its parameters.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

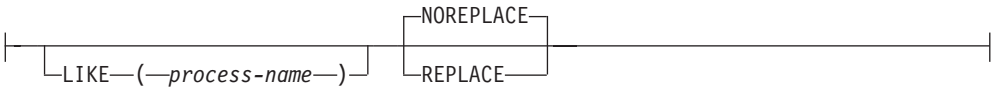
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DEF PRO

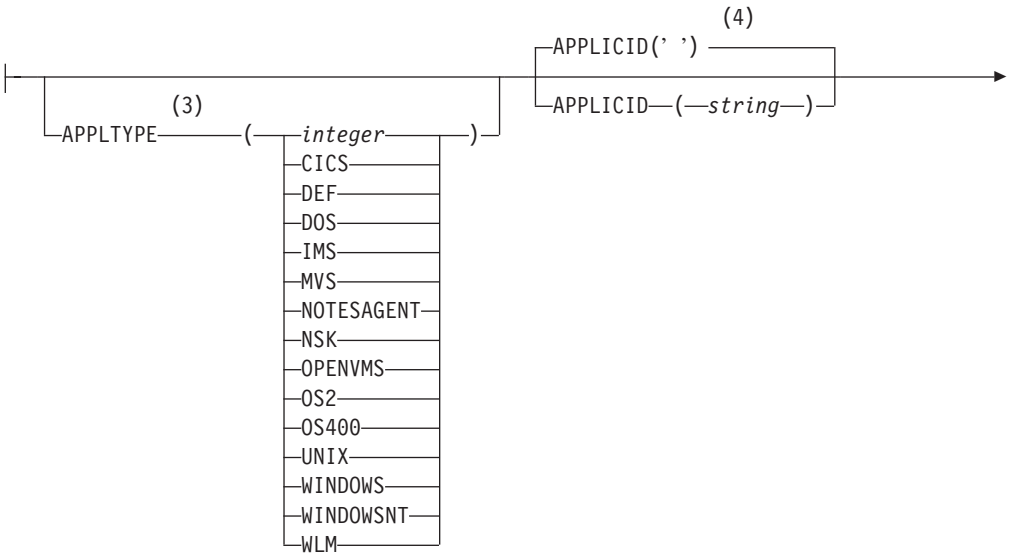
DEFINE PROCESS

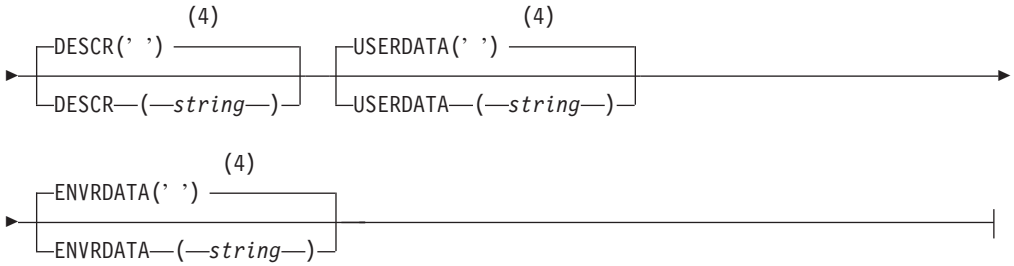


Define attrs:



Process attrs:





Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

3 The default depends on the platform, and can be changed by your installation.

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

Parameter descriptions for DEFINE PROCESS

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.

(process-name)
Name of the WebSphere MQ process definition (see "Rules for naming WebSphere MQ objects" on page 5). 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, 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).

NOTESAGENT
Represents a Lotus Notes agent.

NSK Represents a Compaq NSK application.

OPENVMS
Represents a HP OpenVMS application.

OS2 Represents an OS/2® Warp application.

OS400 Represents an i5/OS application.

UNIX Represents a UNIX application.

WINDOWS
Represents a Windows application.

WINDOWSNT
Represents a Windows NT®, Windows 2000, or Windows XP application.

WLM Represents a z/OS workload manager application.

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 HP OpenVMS, OPENVMS is supported
- On z/OS, CICS (default), DOS, IMS, MVS, OS2, UNIX, WINDOWS, WINDOWSNT, WLM, and DEF are supported
- On i5/OS, OS400 (default), 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.

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.

Note:

1. On z/OS, ENVRDATA is not used by the trigger-monitor applications provided by WebSphere MQ.
2. On z/OS, if APPLTYPE is WLM, the default values for the ServiceName and ServiceStep fields in the work information header (MQWIH) can be supplied in ENVRDATA. The format must be:

   ```
   SERVICENAME=servname,SERVICESTEP=stepname
   ```

   where:

   SERVICENAME=
   is the first 12 characters of ENVRDATA.

   servname
   is a 32-character service name. It can contain embedded blanks or any other data, and have trailing blanks. It is copied to the MQWIH as is.

   SERVICESTEP=
   is the next 13 characters of ENVRDATA.

   stepname
   is a 1- to 8-character service step name. It is copied as is to the MQWIH, and padded to 8 characters with blanks.

   If the format is incorrect, the fields in the MQWIH are set to blanks.
3. 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.

This parameter applies only to the DEFINE PROCESS 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 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 5.

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

Note:

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).

QSGDISP	ALTER	DEFINE
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.	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.

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 attempt to refresh local copies on page set zero:
	`DEFINE PROCESS(process-name)`
	`REPLACE QSGDISP(COPY)`
	
	The ALTER for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.
		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 attempt to make or refresh local copies on page set zero:
		`DEFINE PROCESS(process-name)`
		`REPLACE QSGDISP(COPY)`
		
		The DEFINE for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.

QSGDISP	ALTER	DEFINE
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.	Not permitted.
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.

This parameter applies only to the DEFINE PROCESS 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.

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.

DEFINE PSID

Use the MQSC command DEFINE PSID to define a page set and associated buffer pool.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					1CR

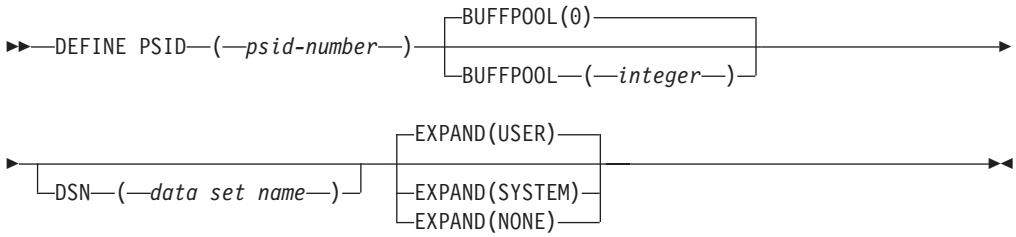
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

You can use ALTER PSID to dynamically change the expansion method.

You can use the DISPLAY USAGE TYPE(PAGESET) command to display information about page sets (see "DISPLAY USAGE" on page 507).

Synonym: DEF PSID

DEFINE PSID



Parameter descriptions for DEFINE PSID

(psid-number)
Identifier of the page set. This is required.

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 LDS 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.

If the buffer pool has not already been created by a DEFINE BUFFPOOL command, the buffer pool is created with 1000 buffers.

DSN(*data set name***)**
The name of a catalogued VSAM LDS data set. This is optional. There is no default.

EXPAND
Controls how the queue manager should expand a page set when it becomes nearly full, and further pages are required in a page set.

USER The secondary extent size that was specified when the page set was defined is used. If no secondary extent size was specified, or it was specified as zero, no dynamic page set expansion can take place.

See the WebSphere MQ for z/OS Concepts and Planning Guide for more information about the EXPAND parameter.

At restart, if a previously used page set has been replaced with a data set that is smaller, it is expanded until it reaches the size of the previously used data set. Only one extent is required to reach this size.

SYSTEM

A secondary extent size that is approximately 10 per cent of the current size of the page set is used. It can be rounded up depending on the characteristics of the DASD.

NONE

No further page set expansion is to take place.

Usage notes for DEFINE PSID

The command can be used in two ways:

1. **At restart, from the CSQINP1 initialization input data set, to specify your standard page sets:**
   - These definitions are not retained so you must define them at each queue manager start using a data set referenced from CSQINP1.
   - You cannot specify the DSN keyword if issuing the command from CSQINP1.
   - If more than one DEFINE PSID command is issued for the same page set, only the last one is processed.
2. **While the queue manager is running, to dynamically add a page set:**
   - The command must specify the DSN keyword and can be issued from either of the following:
     - The z/OS console.
     - The command server and command queue by means of CSQUTIL, CSQINPX, or applications.
   - The page set identifier (that is the PSID number) may have previously been used by a queue manager. It should therefore be freshly formatted by a FORMAT(RECOVER) statement in CSQUTIL, or formatted by with a FORMAT(REPLACE) in CSQUTIL.
   - You cannot dynamically add page set zero.
   - The BUFFPOOL parameter can specify a currently unused buffer pool. If the buffer pool was defined in CSQINP1 but not used by any PSID, then the number of buffers specified there is created if the required virtual storage is available. If this is not available, or if the buffer pool was not defined in CSQINP1, the queue manager attempts to allocate 1000 buffers. If this is not possible, 100 buffers are allocated.
   - You should update your queue manager started task procedure JCL and your CSQINP1 initialization input data set to include the new page set.

DEFINE QUEUE

This section contains the following commands:

- "DEFINE QALIAS" on page 230
- "DEFINE QLOCAL" on page 232
- "DEFINE QMODEL" on page 236
- "DEFINE QREMOTE" on page 240

These commands are supported on the following platforms:

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

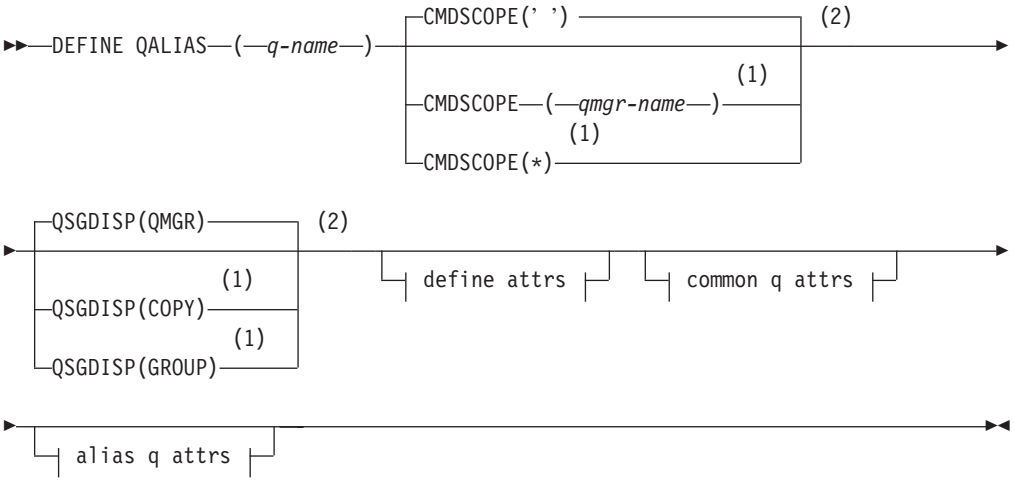
DEFINE QALIAS

Use DEFINE QALIAS to define a new alias queue, and set its parameters.

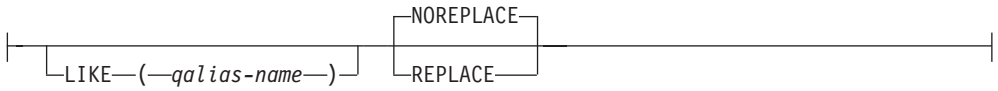
Note: An alias queue provides a level of indirection to another queue or a topic object. If the alias refers to a queue, it must be another local or remote queue, defined at this queue manager. It cannot be another alias queue. If the alias refers to a topic, it must be a topic object defined at this queue manager.

Synonym: DEF QA

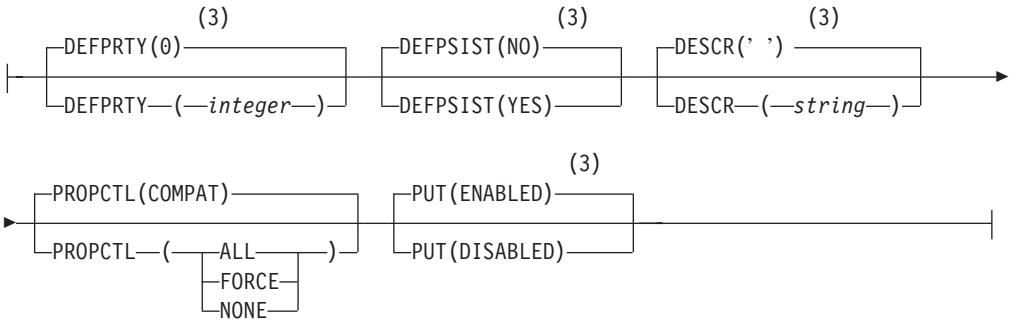
DEFINE QALIAS



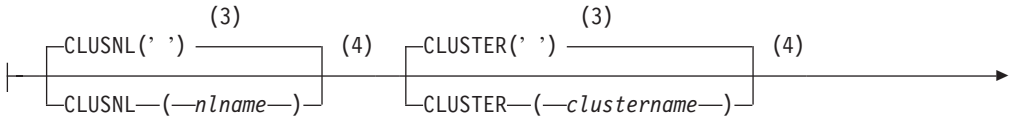
Define attrs:

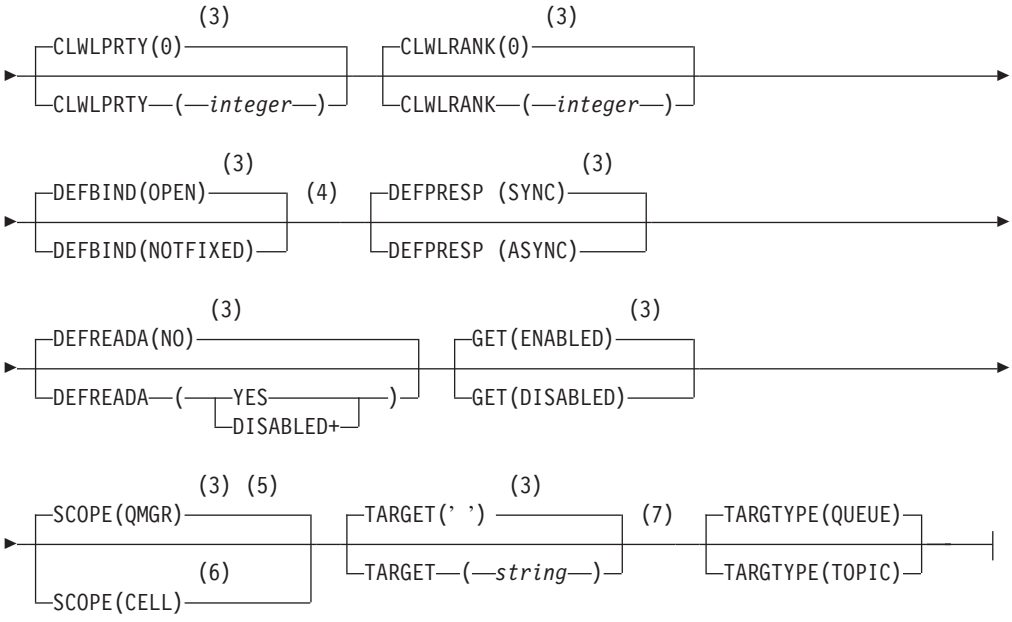


Common q attrs:



Alias q attrs:





Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

5 Valid only on HP OpenVMS, i5/OS, UNIX systems, and Windows.

6 Valid only on HP OpenVMS, UNIX systems, and Windows.

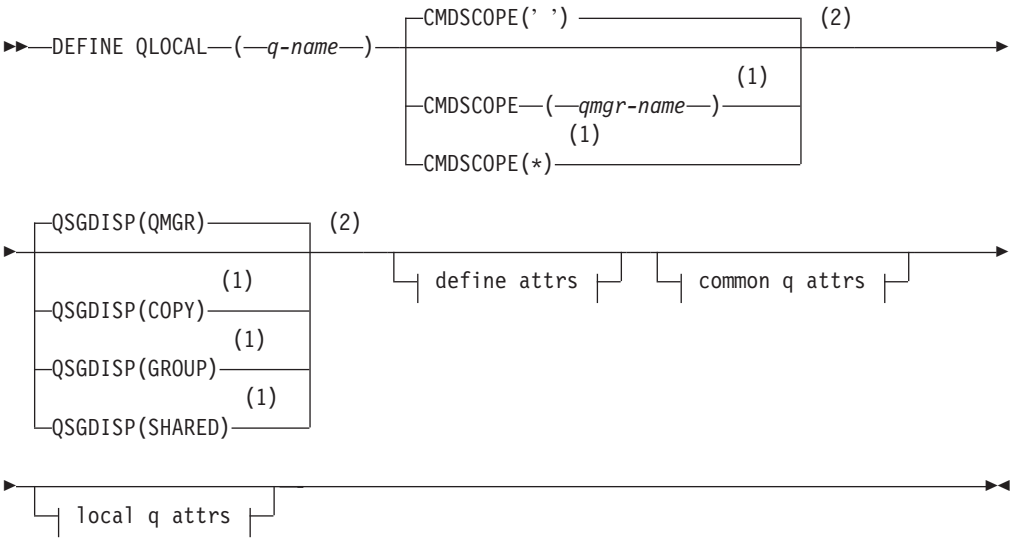
7 The TARGQ parameter is available for compatibility with previous releases. It is a synonym of TARGET; you cannot specify both parameters.

DEFINE QLOCAL

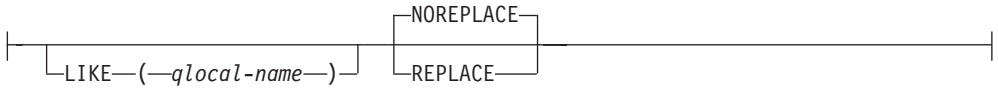
Use DEFINE QLOCAL to define a new local queue, and set its parameters.

Synonym: DEF QL

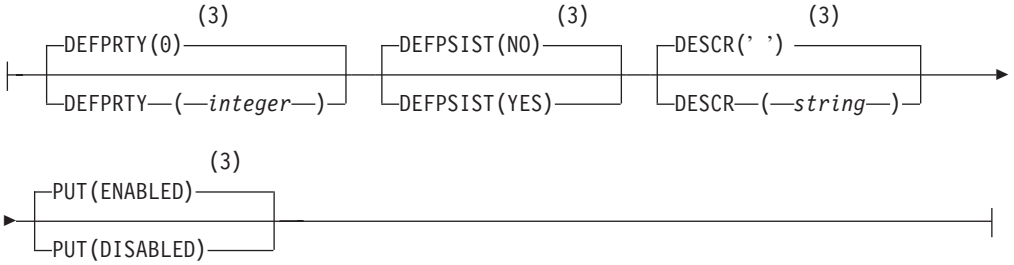
DEFINE QLOCAL



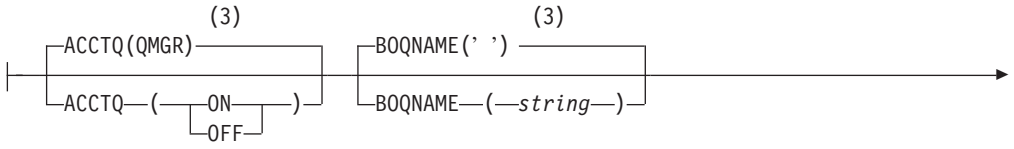
Define attrs:

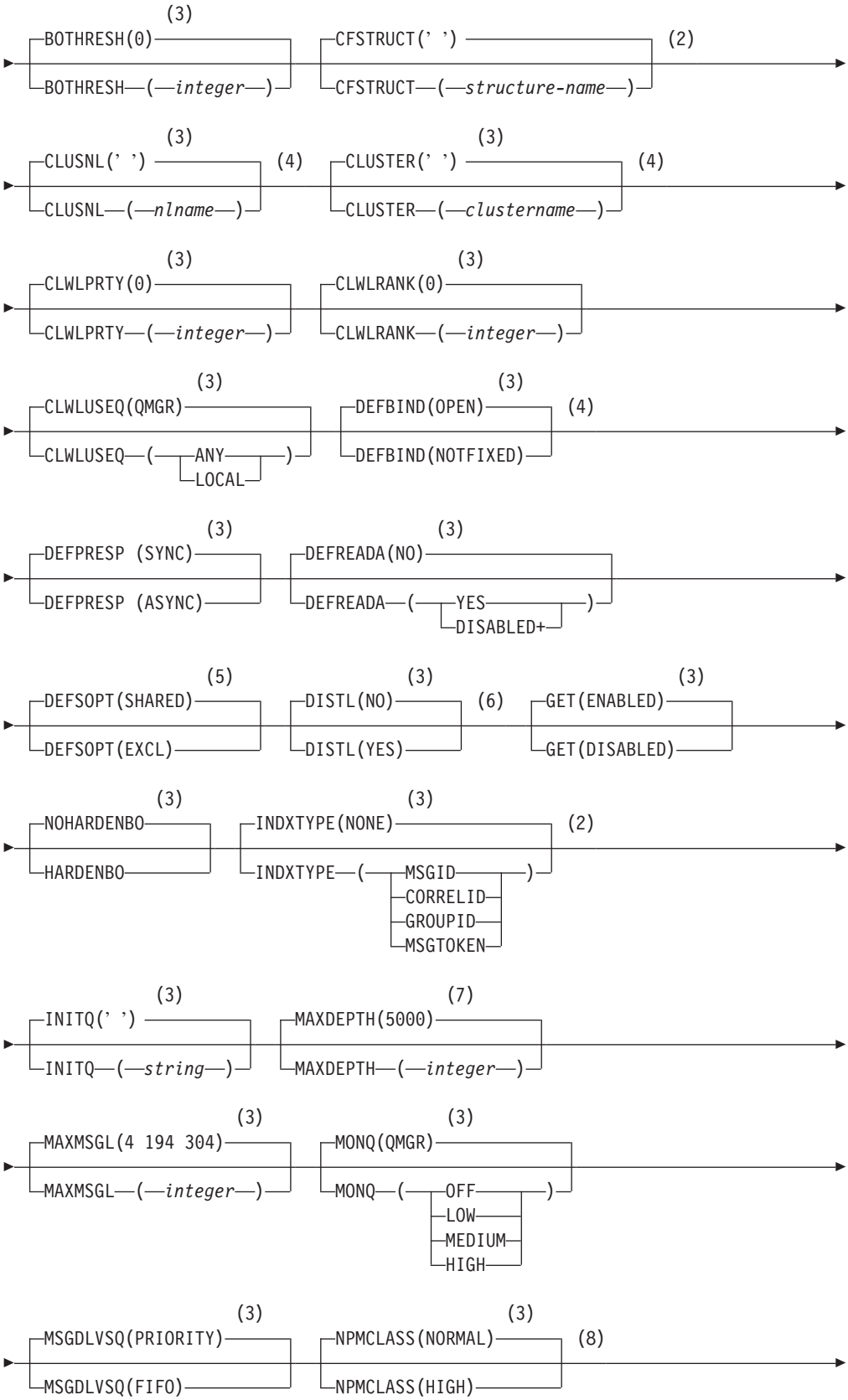


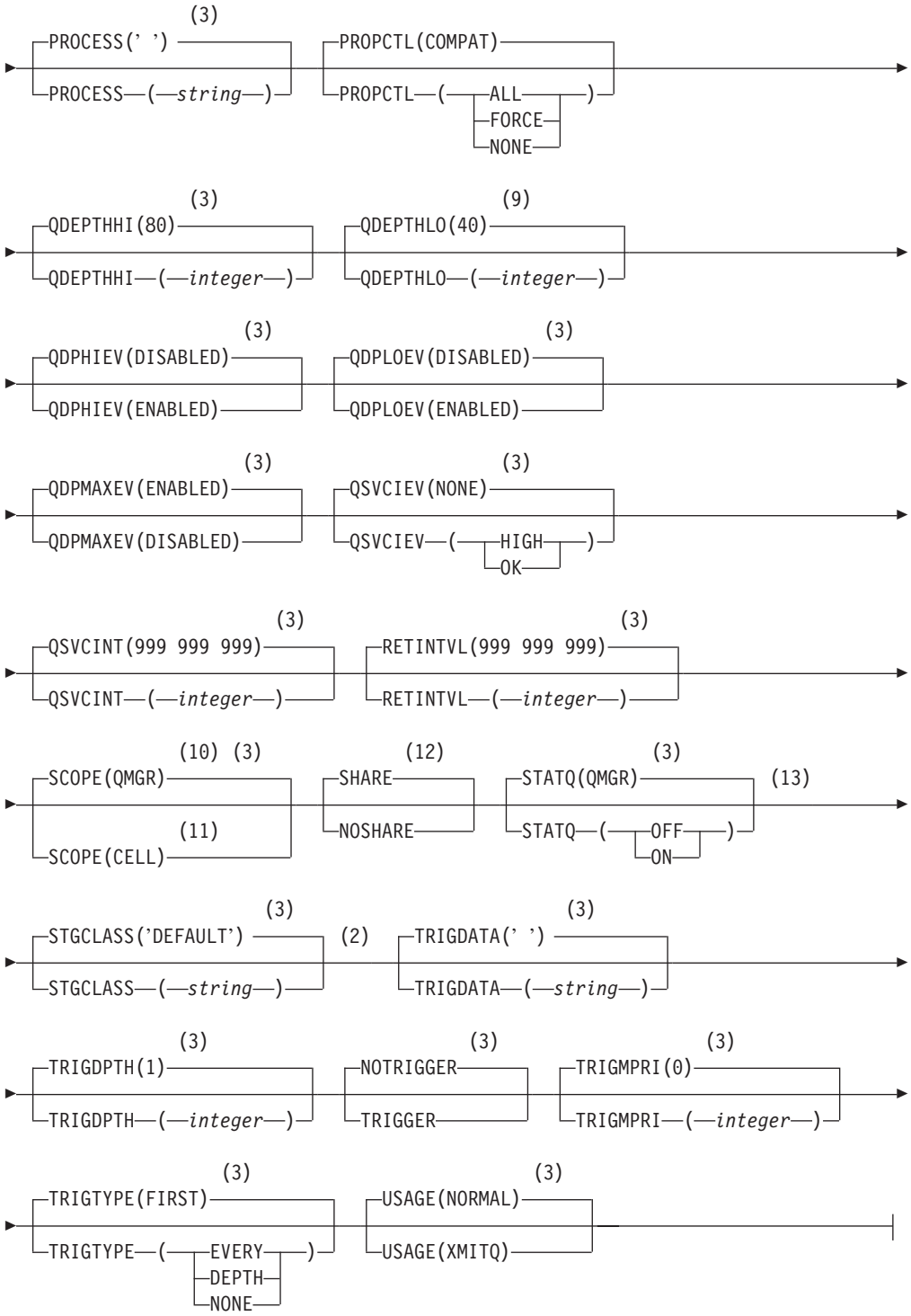
Common q attrs:



Local q attrs:




(3)
BOTHRESH(0)
BOTHRESH—(—integer—)
CFSTRUCT(' ')
CFSTRUCT—(—structure-name—)
(2)
(3)
CLUSNL(' ')
CLUSNL—(—nlname—)
(4)
(3)
CLUSTER(' ')
CLUSTER—(—clustername—)
(4)
(3)
CLWLPRTY(0)
CLWLPRTY—(—integer—)
(3)
CLWLRANK(0)
CLWLRANK—(—integer—)
(3)
CLWLUSEQ(QMGR)
CLWLUSEQ—(—ANY—)
LOCAL
(3)
DEFBIND(OPEN)
DEFBIND(NOTFIXED)
(4)
(3)
DEFPRESP (SYNC)
DEFPRESP (ASYNC)
(3)
DEFREADA(NO)
DEFREADA—(—YES—)
DISABLED+
(5)
DEFSOPT(SHARED)
DEFSOPT(EXCL)
(3)
DISTL(NO)
DISTL(YES)
(6)
(3)
GET(ENABLED)
GET(DISABLED)
(3)
NOHARDENBO
HARDENBO
(3)
INDXTYPE(NONE)
INDXTYPE—(—MSGID—)
CORRELID
GROUPID
MSGTOKEN
(2)
(3)
INITQ(' ')
INITQ—(—string—)
(7)
MAXDEPTH(5000)
MAXDEPTH—(—integer—)
(3)
MAXMSGL(4 194 304)
MAXMSGL—(—integer—)
(3)
MONQ(QMGR)
MONQ—(—OFF—)
LOW
MEDIUM
HIGH
(3)
MSGDLVSQ(PRIORITY)
MSGDLVSQ(FIFO)
(3)
NPMCLASS(NORMAL)
NPMCLASS(HIGH)
(8)



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

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, HP OpenVMS, HP-UX, Linux, i5/OS, 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, HP OpenVMS, HP-UX, Linux, i5/OS, 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 Not valid on z/OS.

9 This is the default supplied with WebSphere MQ (except on platforms other than z/OS where it is 20), but your installation might have changed it.

10 Valid only on HP OpenVMS, i5/OS, UNIX systems, and Windows.

11 Valid only on HP OpenVMS, UNIX systems, and Windows.

12 This is the default supplied with WebSphere MQ (except on z/OS, where it is NOSHARE), but your installation might have changed it.

13 Valid only on i5/OS, UNIX systems, and Windows.

DEFINE QMODEL

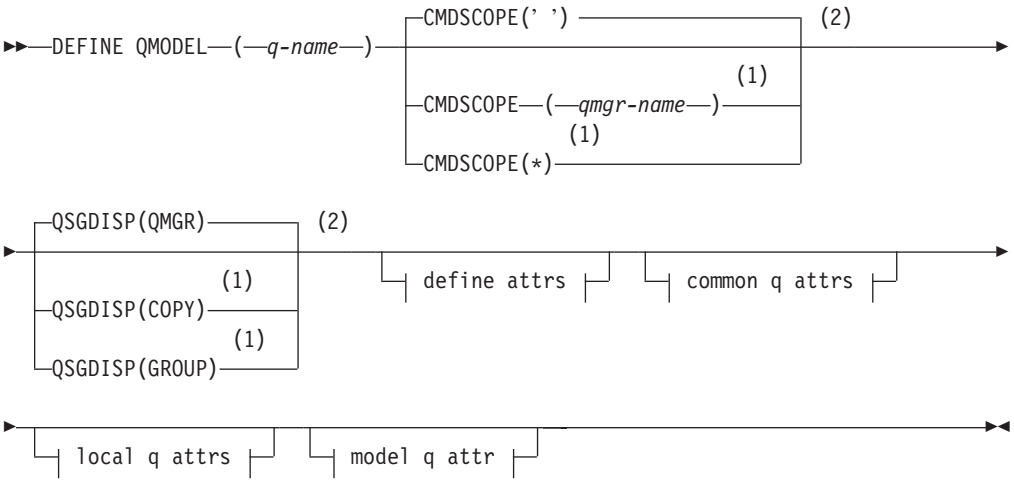
Use DEFINE QMODEL to define a new model queue, and set its parameters.

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.

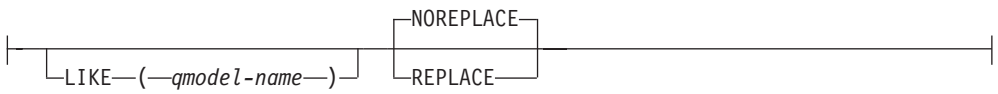
When it has been defined, a model queue (like any other queue) has a complete set of applicable attributes, even if some of these are defaults.

Synonym: DEF QM

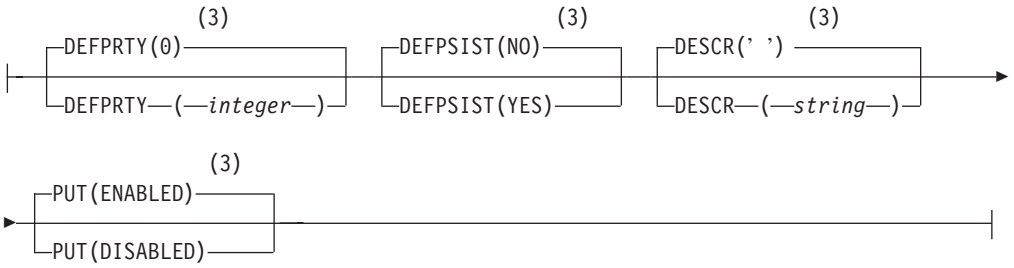
DEFINE QMODEL



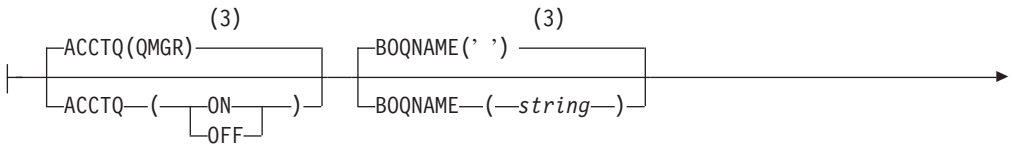
Define attrs:

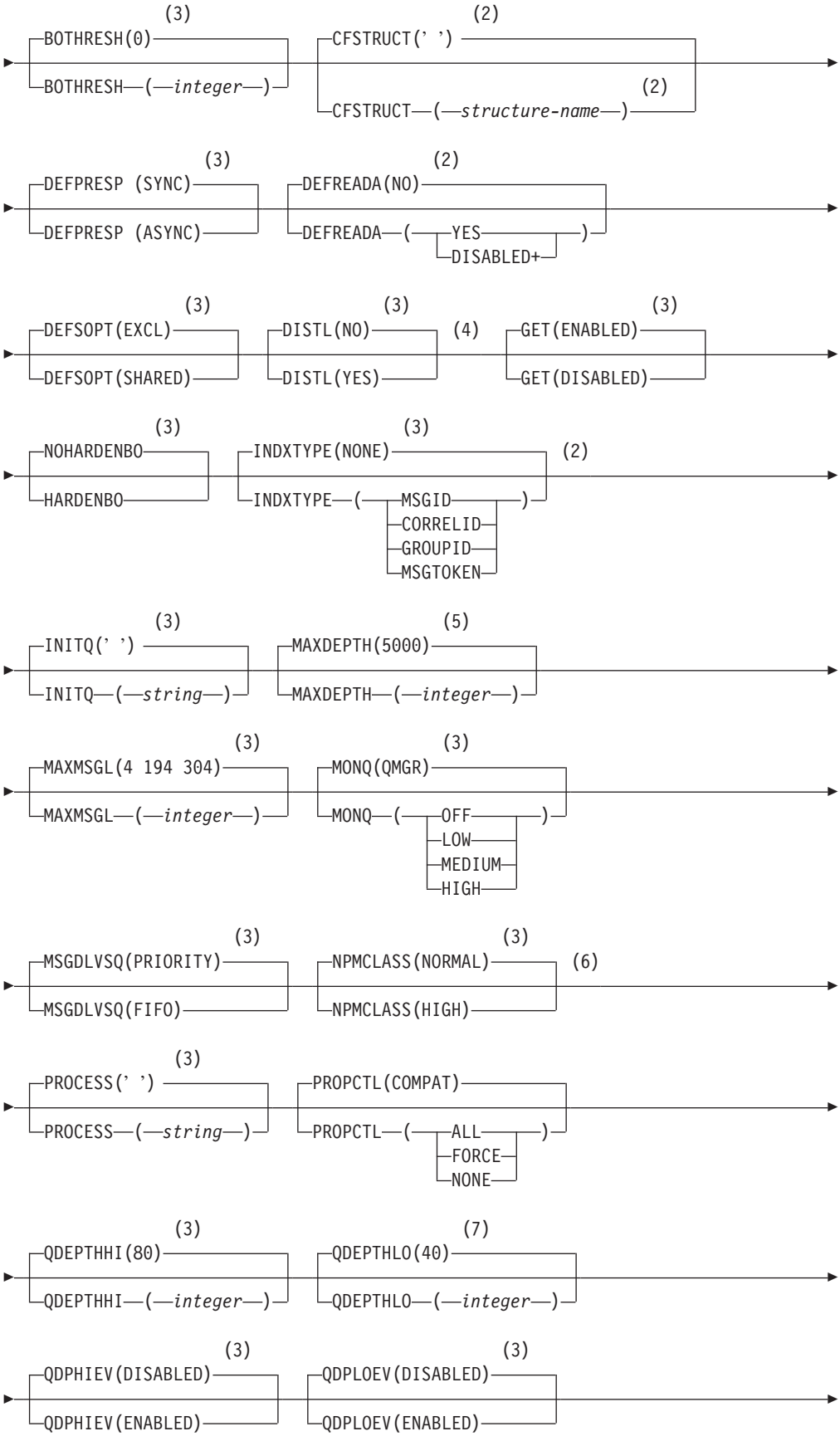


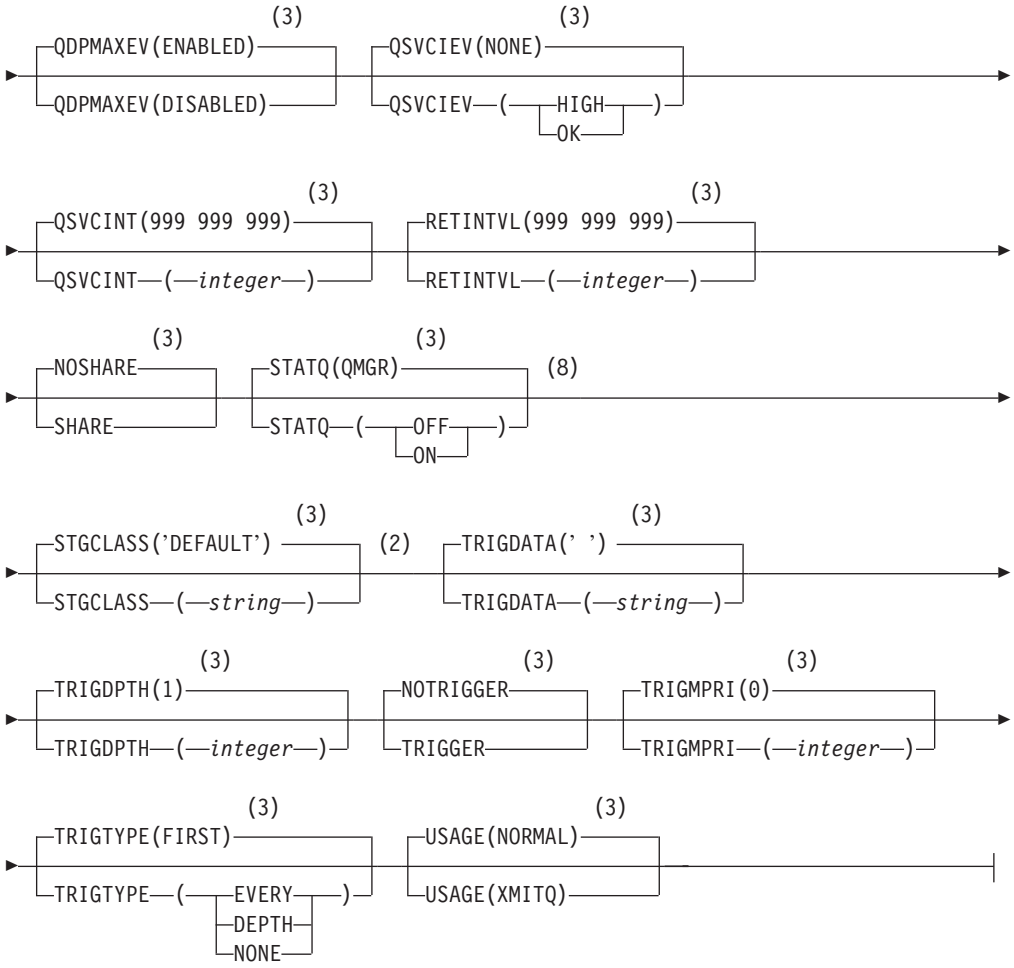
Common q attrs:



Local q attrs:




(3)
BOTHRESH(0)
BOTHRESH—(—integer—)
(2)
CFSTRUCT(' ')
CFSTRUCT—(—structure-name—)
(2)
(3)
DEFPRESP (SYNC)
DEFPRESP (ASYNC)
(2)
DEFREADA(NO)
DEFREADA—(—YES—)
DISABLED+
(3)
DEFSOPT(EXCL)
DEFSOPT(SHARED)
(3)
DISTL(NO)
DISTL(YES)
(4)
(3)
GET(ENABLED)
GET(DISABLED)
(3)
NOHARDENBO
HARDENBO
(3)
INDXTYPE(NONE)
(2)
INDXTYPE—(—MSGID—)
CORRELID
GROUPID
MSGTOKEN
(3)
INITQ(' ')
INITQ—(—string—)
(5)
MAXDEPTH(5000)
MAXDEPTH—(—integer—)
(3)
MAXMSGL(4 194 304)
MAXMSGL—(—integer—)
(3)
MONQ(QMGR)
MONQ—(—OFF—)
LOW
MEDIUM
HIGH
(3)
MSGDLVSQ(PRIORITY)
MSGDLVSQ(FIFO)
(3)
NPMCLASS(NORMAL)
NPMCLASS(HIGH)
(6)
(3)
PROCESS(' ')
PROCESS—(—string—)
PROPCTL(COMPAT)
PROPCTL—(—ALL—)
FORCE
NONE
(3)
QDEPTHHI(80)
QDEPTHHI—(—integer—)
(7)
QDEPTHLO(40)
QDEPTHLO—(—integer—)
(3)
QDPHIEV(DISABLED)
QDPHIEV(ENABLED)
(3)
QDPLOEV(DISABLED)
QDPLOEV(ENABLED)



Model q attr:

Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

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, HP OpenVMS, HP-UX, Linux, i5/OS, 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.

6 Not valid on z/OS.

7 This is the default supplied with WebSphere MQ (except on platforms other than z/OS where it is 20), but your installation might have changed it.

8 Valid only on i5/OS, UNIX systems, and Windows.

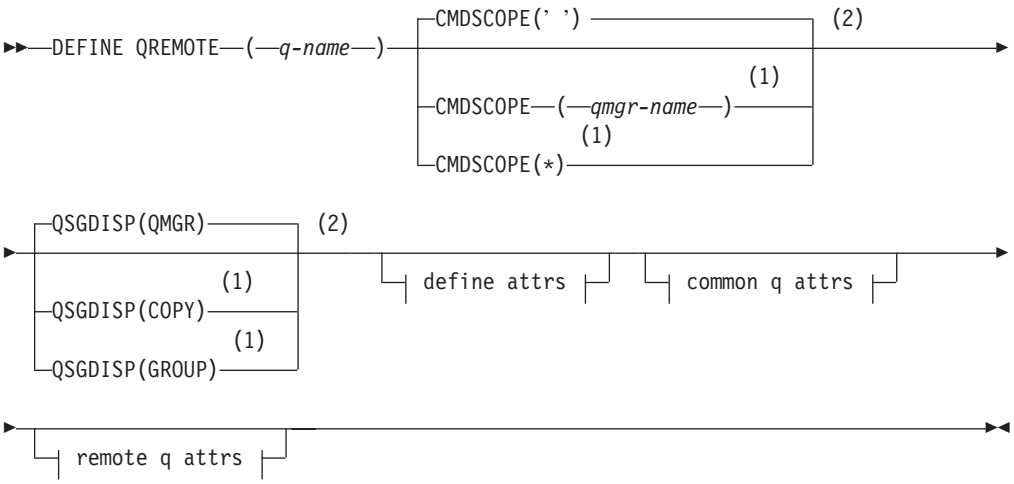
DEFINE QREMOTE

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.

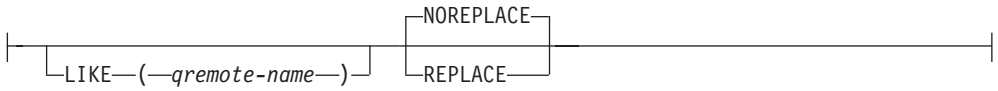
A remote queue is one that is owned by another queue manager that application processes connected to this queue manager need to access.

Synonym: DEF QR

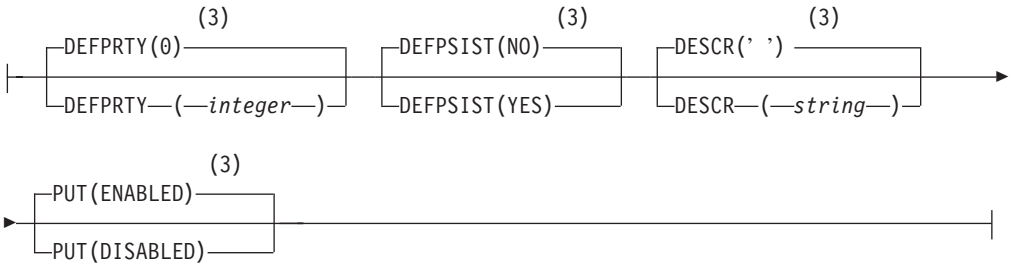
DEFINE QREMOTE



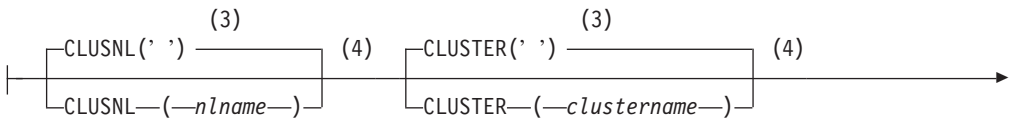
Define attrs:

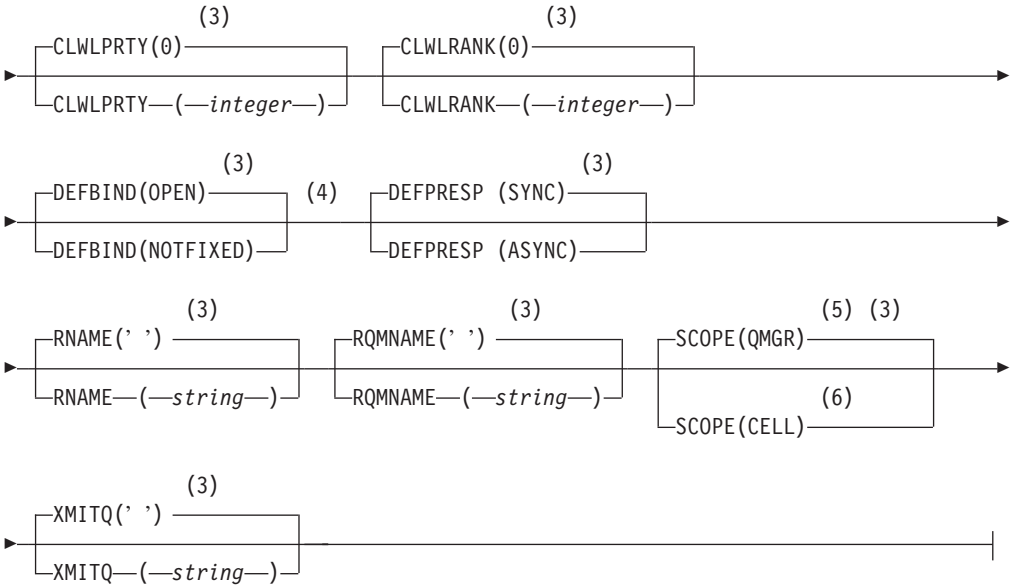


Common q attrs:



Remote q attrs:





Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

5 Valid only on HP OpenVMS, i5/OS, UNIX systems, and Windows.

6 Valid only on HP OpenVMS, UNIX systems, and Windows.

Parameter descriptions for DEFINE QUEUE and ALTER QUEUE.

These parameter descriptions apply to the DEFINE QUEUE and 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** parameters apply only to the DEFINE QUEUE commands.

Table 8 shows the parameters that are relevant for each type of queue. There is a description of each parameter after the table.

Table 8. DEFINE and ALTER QUEUE parameters

	Local queue	Model queue	Alias queue	Remote queue
ACCTQ	X	X		
BOQNAME	X	X		
BOTHRESH	X	X		
CFSTRUCT	X	X		

Table 8. DEFINE and ALTER QUEUE parameters (continued)

	Local queue	Model queue	Alias queue	Remote queue
CLUSNL	X		X	X
CLUSTER	X		X	X
CLWLPRTY	X		X	X
CLWLRANK	X		X	X
CLWLUSEQ	X			
CMDSCOPE	X	X	X	X
DEFBIND	X		X	X
DEFPRESP	X	X	X	X
DEFPRTY	X	X	X	X
DEFPSIST	X	X	X	X
DEFREADA	X	X	X	
DEFSOPT	X	X		
DEFTYPE	X	X		
DESCR	X	X	X	X
DISTL	X	X		
FORCE	X		X	X
GET	X	X	X	
HARDENBO or NOHARDENBO	X	X		
INDXTYPE	X	X		
INITQ	X	X		
LIKE	X	X	X	X
MAXDEPTH	X	X		
MAXMSGL	X	X		
MONQ	X	X		
MSGDLVSQ	X	X		
NOREPLACE	X	X	X	X
NPMCLASS	X	X		
PROCESS	X	X		
PROPCTL	X	X	X	
PUT	X	X	X	X
q-name	X	X	X	X
QDEPTHHI	X	X		
QDEPTHLO	X	X		
QDPHIEV	X	X		
QDPLOEV	X	X		
QDPMAXEV	X	X		
QSGDISP	X	X	X	X
QSVCIEV	X	X		
QSVCINT	X	X		

Table 8. DEFINE and ALTER QUEUE parameters (continued)

	Local queue	Model queue	Alias queue	Remote queue
REPLACE	X	X	X	X
RETINTVL	X	X		
RNAME				X
RQMNAME				X
SCOPE	X		X	X
SHARE or NOSHARE	X	X		
STATQ	X	X		
STGCLASS	X	X		
TARGET			X	
TARGQ			X	
TARGTYPE			X	
TRIGDATA	X	X		
TRIGDPTH	X	X		
TRIGGER or NOTRIGGER	X	X		
TRIGMPRI	X	X		
TRIGTYPE	X	X		
USAGE	X	X		
XMITQ				X

(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 5.

ACCTQ

Specifies whether accounting data collection is to be enabled for the queue. On z/OS, the data collected is class 3 accounting data (thread-level and queue-level accounting). In order for accounting data to be collected for this queue, accounting data for this connection must also be enabled by either the ACCTQ queue manager attribute or the options field in the MQCNO structure on the MQCONNX call.

QMGR
The collection of accounting data is based on the setting of the ACCTQ parameter on the queue manager definition.

ON Accounting data collection is enabled for the queue unless the ACCTQ queue manager parameter has a value of NONE. On z/OS systems, you need to have switched on class 3 accounting by the START TRACE command.

OFF Accounting data collection is disabled for the queue.

BOQNAME(*string***)**
The excessive backout requeue name.

This parameter is supported only on local and model queues.

Use this parameter to set or change the value of the *BackoutRequeueQName* attribute of a local or model queue. Apart from allowing its value to be queried, the queue manager takes no action based on the value of this attribute. However, the Application Server Facilities (ASF) of WebSphere MQ classes for JMS use the attribute to determine where to transfer a message that has already been backed out the maximum number of times as specified by the *BackoutThreshold* attribute.

BOTHRESH(*integer***)**
The backout threshold.

This parameter is supported only on local and model queues.

Use this parameter to set or change the value of the *BackoutThreshold* attribute of a local or model queue. Apart from allowing its value to be queried, the queue manager takes no action based on the value of this attribute. However, the Application Server Facilities (ASF) of WebSphere MQ classes for JMS use the attribute to determine how many times to allow a message to be backed out before transferring the message to the queue specified by the *BackoutRequeueQName* attribute.

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.

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, HP OpenVMS, HP-UX, Linux, 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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

CLWLPRTY(*integer***)**
Specifies the priority of the queue for the purposes of cluster workload distribution. This parameter is valid only for local, remote, and alias queues. The value must be in the range zero through 9 where zero is the lowest priority and 9 is the highest. For more information about this attribute, see WebSphere MQ Queue Manager Clusters.

CLWLRANK(*integer***)**
Specifies the rank of the queue for the purposes of cluster workload distribution. This parameter is valid only for local, remote, and alias

queues. The value must be in the range zero through 9 where zero is the lowest rank and 9 is the highest. For more information about this attribute, see WebSphere MQ Queue Manager Clusters.

CLWLUSEQ
Specifies the behavior of an MQPUT operation when the target queue has a local instance and at least one remote cluster instance (except where the MQPUT originates from a cluster channel). This parameter is valid only for local queues.

QMGR
The behavior will be as specified by the CLWLUSEQ parameter of the queue manager definition.

ANY The queue manager is to treat the local queue as another instance of the cluster queue for the purposes of workload distribution.

LOCAL
The local queue is the only target of the MQPUT operation.

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 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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

DEFPRESP
Specifies the behavior to be used by applications when the put response type, within the MQPMO options, is set to MQPMO_RESPONSE_AS_Q_DEF.

SYNC Ensures that the put operations to the queue specifying MQPMO_RESPONSE_AS_Q_DEF are issued as if MQPMO_SYNC_RESPONSE had been specified instead. This is the default supplied with WebSphere MQ, but your installation might have changed it.

ASYNC
Ensures that the put operations to the queue specifying MQPMO_RESPONSE_AS_Q_DEF are issued as if MQPMO_ASYNC_RESPONSE had been specified instead.

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.)

DEFPSIST
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**.

DEFREADA
Specifies the default read ahead behavior for non-persistent messages delivered to the client. Enabling read ahead can improve the performance of client applications consuming non-persistent messages.

NO Non-persistent messages are not read ahead unless the client application is configured to request read ahead.

YES Non-persistent messages are sent to the client before an application requests them. Non-persistent messages can be lost if the client ends abnormally or if the client does not consume all the messages it is sent.

DISABLED
Read ahead of non-persistent messages in not enabled for this queue. Messages are not sent ahead to the client regardless of whether read ahead is requested by the client application.

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 DEFPSIST parameter of YES.

If you specify this option, do not specify INDXTYPE(MSGTOKEN).

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, HP OpenVMS, HP-UX, Linux, i5/OS, 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
- 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).

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.

Note: This parameter affects only WebSphere MQ for z/OS. It can be set on other platforms but has no effect.

INDXTYPE
The type of index maintained by the queue manager to expedite **MQGET** operations on the queue. For shared queues, the type of index determines the type of **MQGET** operations that can be used.

This parameter is supported only on local and model queues.

Messages can be retrieved using a selection criterion only if an appropriate index type is maintained, as the following table shows:

Retrieval selection criterion	Index type required	
	Shared queue	Other queue
None (sequential retrieval)	Any	Any
Message identifier	MSGID or NONE	Any
Correlation identifier	CORRELID	Any
Message and correlation identifiers	MSGID or CORRELID	Any
Group identifier	GROUPID	Any
Grouping	GROUPID	GROUPID
Message token	Not allowed	MSGTOKEN

where the value of INDXTYPE parameter has the following values:

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.

Note:

1. You cannot set INDXTYPE to GROUPID if the queue is a transmission queue.
2. 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)

For queues that are not shared and do not use grouping or message tokens, the index type does not restrict the type of retrieval selection. However, the index is used to expedite GET operations on the queue, so choose the type that corresponds to the most common retrieval selection.

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:

Queue type		**NON-SHARED**			**SHARED**	
Queue state		**Uncommitted activity**	**No uncommitted activity, messages present**	**No uncommitted activity, and empty**	**Open or messages present**	**Not open, and empty**
Change INDXTYPE from:	**To:**	**Change allowed?**				
NONE	MSGID	No	Yes	Yes	No	Yes
NONE	CORRELID	No	Yes	Yes	No	Yes
NONE	MSGTOKEN	No	No	Yes	-	-
NONE	GROUPID	No	No	Yes	No	Yes
MSGID	NONE	No	Yes	Yes	No	Yes
MSGID	CORRELID	No	Yes	Yes	No	Yes
MSGID	MSGTOKEN	No	No	Yes	-	-
MSGID	GROUPID	No	No	Yes	No	Yes
CORRELID	NONE	No	Yes	Yes	No	Yes
CORRELID	MSGID	No	Yes	Yes	No	Yes
CORRELID	MSGTOKEN	No	No	Yes	-	-
CORRELID	GROUPID	No	No	Yes	No	Yes
MSGTOKEN	NONE	No	Yes	Yes	-	-
MSGTOKEN	MSGID	No	Yes	Yes	-	-
MSGTOKEN	CORRELID	No	Yes	Yes	-	-
MSGTOKEN	GROUPID	No	No	Yes	-	-
GROUPID	NONE	No	No	Yes	No	Yes
GROUPID	MSGID	No	No	Yes	No	Yes
GROUPID	CORRELID	No	No	Yes	No	Yes
GROUPID	MSGTOKEN	No	No	Yes	-	-

This parameter is supported only on z/OS. On other platforms, all queues are automatically indexed.

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 5.

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:

SYSTEM.DEFAULT.ALIAS.QUEUE	Alias queue
SYSTEM.DEFAULT.LOCAL.QUEUE	Local queue
SYSTEM.DEFAULT.MODEL.QUEUE	Model queue
SYSTEM.DEFAULT.REMOTE.QUEUE	Remote queue

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 z/OS, the queue manager searches for an object with the name and queue type you specify with a disposition of QMGR, COPY, or SHARED. The disposition of the LIKE object is not copied to the object you are defining.

Note:

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, HP OpenVMS, HP-UX, Linux, i5/OS, 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, HP OpenVMS, HP-UX, Linux, i5/OS, 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 (104 857 600 bytes).

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

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.

MONQ

Controls the collection of online monitoring data for queues.

This parameter is supported only on local and model queues.

QMGR
Collect monitoring data according to the setting of the queue manager parameter MONQ. This is the default value.

OFF Online monitoring data collection is turned off for this queue.

LOW If the value of the queue manager's MONQ parameter is not NONE, online monitoring data collection is turned on for this queue.

MEDIUM
If the value of the queue manager's MONQ parameter is not NONE, online monitoring data collection is turned on for this queue.

HIGH If the value of the queue manager's MONQ parameter is not NONE, online monitoring data collection is turned on for this queue.

Note that there is no distinction between the values LOW, MEDIUM and HIGH. These values all turn data collection on, but do not affect the rate of collection.

When this parameter is used in an ALTER queue command, the change is effective only when the queue is next opened.

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.

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.

NPMCLASS
The level of reliability to be assigned to non-persistent messages that are put to the queue:

NORMAL
Non-persistent messages are lost after a failure, or queue manager shutdown. These messages are discarded on a queue manager restart.

HIGH The queue manager attempts to retain non-persistent messages on this queue over a queue manager restart.

You cannot set this parameter on z/OS.

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 5.

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 definition contains the name of the channel to be started. This parameter is optional for transmission queues on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS; if you do not specify it, the channel name is taken from the value specified for the TRIGDATA parameter.

PROPCTL
Property control attribute.

This parameter is applicable to Local, Alias and Model queues.

This parameter is optional.

Permissable values are:

ALL All properties of the message are included with the message when it is sent to the remote queue manager. The properties, except those in the message descriptor (or extension), are placed in one or more MQRFH2 headers in the message data.

COMPAT
If the message contains a property with a prefix of mcd., jms., usr. or mqext., all message properties are delivered to the application in an MQRFH2 header. Otherwise all properties of the message, except those contained in the message descriptor (or extension), are discarded and are no longer accessible to the application.

This is the default value; it allows applications which expect JMS related properties to be in an MQRFH2 header in the message data to continue to work unmodified.

FORCE
Properties are always returned in the message data in an MQRFH2 header regardless of whether the application specifies a message handle.

A valid message handle supplied in the MsgHandle field of the MQGMO structure on the MQGET call is ignored. Properties of the message are not accessible via the message handle.

NONE
All properties of the message, except those in the message descriptor (or extension), are removed from the message before the message is sent to the remote queue manager.

Note, that for a transmission queue, which is a local queue with USAGE(XMITQ), the PROPCTL attribute of the queue is irrelevant and it is the PROPCTL attribute on the corresponding channel object that controls the message property behavior.

PUT Whether messages can be put on the queue.

ENABLED
Messages can be added to the queue (by suitably authorized applications).

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 Monitoring WebSphere MQ 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 Monitoring WebSphere MQ 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 Monitoring WebSphere MQ 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 Monitoring WebSphere MQ 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 Monitoring WebSphere MQ 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.

QSGDISP	ALTER	DEFINE
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.	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.

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), 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 attempt to refresh local copies on page set zero:
	`DEFINE QUEUE(q-name)`
	`REPLACE QSGDISP(COPY)`
	
	The ALTER for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.
		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 attempt to make or refresh local copies on page set zero:
		`DEFINE QUEUE(q-name)`
		`REPLACE QSGDISP(COPY)`
		
		The DEFINE for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.

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.	Not permitted.
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. For local queues, messages are stored on the page sets of each queue manager and are available only through that queue manager.

QSGDISP	ALTER	DEFINE
SHARED	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.	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: • CFSTRUCT is nonblank • INDXTYPE is not MSGTOKEN • The queue is not one of the following: – SYSTEM.CHANNEL.INITQ – SYSTEM.COMMAND.INPUT 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.

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 Monitoring WebSphere MQ 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 QSVCIEV parameter.

The value is in units of milliseconds, and must bein 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.

This parameter applies only to the DEFINE Queue command.

(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 HP OpenVMS, UNIX systems, 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.

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***)**
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.

In a queue manager cluster, this definition applies only to the queue manager that made it. To advertise the alias to the whole cluster, add the CLUSTER attribute to the remote queue definition.

- 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 5).

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 5).

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 has been configured (note that the DCE name service is no longer supported).

This parameter is valid only on HP OpenVMS, 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

STATQ
Whether statistics data collection is enabled:

QMGR
Statistics data collection is based on the setting of the queue manager's STATQ parameter.

ON If the value of the queue manager's STATQ parameter is not NONE, statistics data collection for the queue is enabled.

OFF Statistics data collection for the queue is disabled.

When this parameter is used in an ALTER queue command, the change is effective only only for connections to the queue manager made after the change to the parameter.

This parameter is valid only on i5/OS, UNIX systems, and Windows.

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.

If you specify QSGDISP(SHARED) or DEFTYPE(SHAREDYN), this parameter is ignored.

TARGET(*string***)**
The name of the queue or topic object being aliased. (See "Rules for naming WebSphere MQ objects" on page 5.) The object can be a queue or a topic as defined by TARGTYPE. The maximum length is 48 characters.

This parameter is supported only on alias queues.

This object needs to be defined only when an application process opens the alias queue.

This is a synonym of the parameter TARGQ which has been retained for compatibility purposes. If you specify TARGET, you cannot also specify TARGQ.

TARGTYPE(*string***)**
The type of object to which the alias resolves.

QUEUE
The alias resolves to a queue.

TOPIC
The alias resolves to a topic.

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 NonStop Kernel, HP OpenVMS, HP-UX, Linux, i5/OS, 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 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 414 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 for DEFINE queues

1. For local queues
   a. If you define a local queue with QSGDISP(SHARED) and another queue manager in the queue-sharing group already has a local version of a queue with the same name (it can be of type QLOCAL, QREMOTE, or QALIAS and will have QSGDISP(QMGR)), the definition will complete normally. However, when you try to access the locally-defined queue, it will fail with reason code 2343 (MQRC_OBJECT_NOT_UNIQUE).

      To resolve this, you must delete one of the queues using the DELETE command. If the queue you want to delete contains messages, you can use the PURGE option or remove the messages first using the MOVE command.

      For example, if you want to delete the QSGDISP(LOCAL) version, which contains messages, and you want to copy those messages to the QSGDISP(SHARED) version, then issue commands like the following:

```
MOVE QLOCAL(QUEUE.1) QSGDISP(PRIVATE) TOQLOCAL(QUEUE.1) TYPE(ADD)
DELETE QLOCAL(QUEUE.1) QSGDISP(QMGR)
```

2. For alias queues:
   a. `DEFINE QALIAS(aliasqueue) TARGET(otherqname) CLUSTER(c)` has the effect of advertising queue *otherqname* by the name *aliasqueue*.
   b. `DEFINE QALIAS(aliasqueue) TARGET(otherqname)` has the effect of allowing a queue advertised by the name *otherqname* to be used on this queue manager by the name *aliasqueue*.
   c. TARGTYPE and TARGET are not cluster attributes, that is, they are not shared in a cluster environment.
3. 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).

DEFINE SERVICE

Use the MQSC command DEFINE SERVICE to define a new WebSphere MQ service definition, and set its parameters.

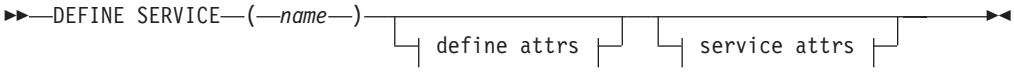
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	

A service is used to define the user programs that are to be started and stopped when the queue manager is started and stopped. You can also start and stop these programs by issuing the START SERVICE and STOP SERVICE commands.

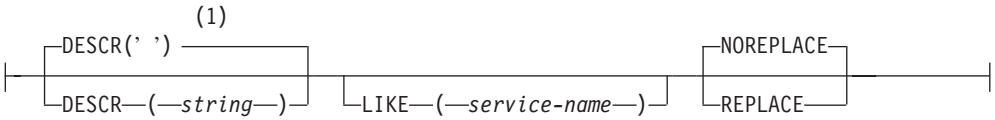
For more information about services, see the WebSphere MQ System Administration Guide.

Synonym:

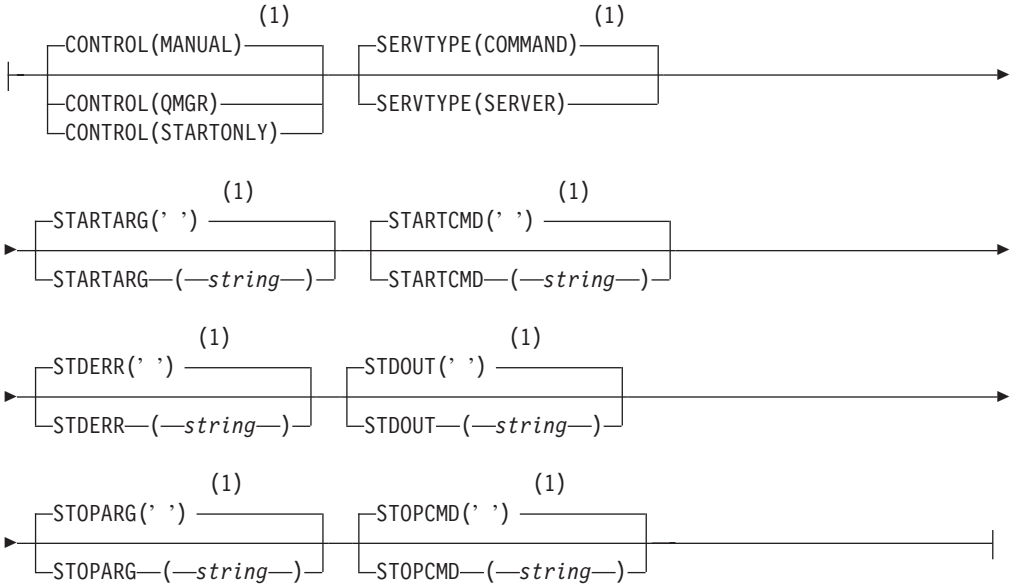
DEFINE SERVICE



Define attrs:



Service attrs:



Notes:

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

Parameter descriptions for DEFINE SERVICE

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

- The **LIKE** parameter applies only to the DEFINE SERVICE command.
- The **NOREPLACE** and **REPLACE** parameter applies only to the DEFINE SERVICE command.

(service-name)
Name of the WebSphere MQ service definition (see "Rules for naming WebSphere MQ objects" on page 5). This is required.

The name must not be the same as any other service definition currently defined on this queue manager (unless REPLACE is specified).

CONTROL(*string***)**
Specifies how the service is to be started and stopped:

MANUAL
The service is not to be started automatically or stopped automatically. It is to be controlled by use of the START SERVICE and STOP SERVICE commands. This is the default value.

QMGR
The service being defined is to be started and stopped at the same time as the queue manager is started and stopped.

STARTONLY
The service is to be started at the same time as the queue manager is started, but is not requested to stop when the queue manager is stopped.

DESCR(*string***)**
Plain-text comment. It provides descriptive information about the service when an operator issues the DISPLAY SERVICE command (see "DISPLAY SERVICE" on page 468).

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(*service-name***)**
The name of a service the parameters of which are used to model this definition.

This parameter applies only to the DEFINE SERVICE 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 services on this queue manager. This is equivalent to specifying:

```
LIKE(SYSTEM.DEFAULT.SERVICE)
```

A default service is provided but it can be altered by the installation of the default values required. See “Rules for naming WebSphere MQ objects” on page 5.

REPLACE and NOREPLACE
Whether the existing definition is to be replaced with this one. This is optional. The default is NOREPLACE.

This parameter applies only to the DEFINE SERVICE 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.

SERVTYPE
Specifies the mode in which the service is to run:

COMMAND
A command service object. Multiple instances of a command service object can be executed concurrently. You cannot monitor the status of command service objects.

SERVER
A server service object. Only one instance of a server service object can be executed at a time. The status of server service objects can be monitored using the DISPLAY SVSTATUS command.

STARTARG(*string***)**
Specifies the arguments to be passed to the user program at queue manager startup.

STARTCMD(*string***)**
Specifies the name of the program which is to run. You must specify a fully qualified path name to the executable program.

STDERR(*string***)**
Specifies the path to a file to which the standard error (stderr) of the service program should be redirected. If the file does not exist when the service program is started, the file is created. If this value is blank then any data written to stderr by the service program is discarded.

STDOUT(*string***)**
Specifies the path to a file to which the standard output (stdout) of the service program should be redirected. If the file does not exist when the service program is started, the file is created. If this value is blank then any data written to stdout by the service program is discarded.

STOPARG(*string***)**
Specifies the arguments to be passed to the stop program when instructed to stop the service.

STOPCMD(*string***)**
Specifies the name of the executable program to run when the service is requested to stop. You must specify a fully qualified path name to the executable program.

DEFINE STGCLASS

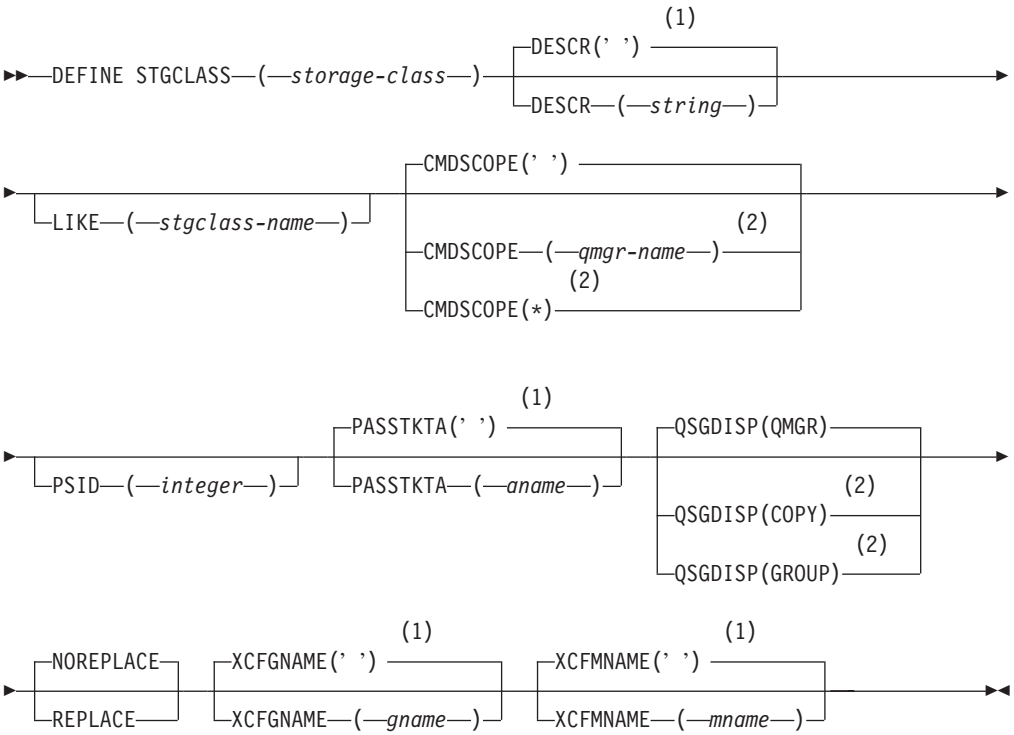
Use the MQSC command DEFINE STGCLASS to define a storage class to page set mapping.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DEF STC

DEFINE STGCLASS



Notes:

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

2 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for DEFINE STGCLASS

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.

(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.

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.

This parameter applies only to the DEFINE STGCLASS 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 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 for an object with the name you specify and a disposition of QMGR or COPY. The disposition of the LIKE object is not copied to the object you are defining.

Note:

1. QSGDISP (GROUP) objects are not searched.
2. LIKE is ignored if QSGDISP(COPY) is specified.

PASSTKTA(*application name***)**
The application name that is passed to RACF when authenticating the passticket specified in the MQIIH header.

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 227.

QSGDISP
Specifies the disposition of the object in the group.

QSGDISP	ALTER	DEFINE
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.	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.

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 attempt to refresh local copies on page set zero:
	`DEFINE STGCLASS(storage-class)`
	`REPLACE QSGDISP(COPY)`
	
	The ALTER for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.
		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 attempt to make or refresh local copies on page set zero:
		`DEFINE STGCLASS(storage-class)`
		`REPLACE QSGDISP(COPY)`
		
		The DEFINE for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.

QSGDISP	ALTER	DEFINE
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.	Not permitted.
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 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 STGCLASS command.

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 for DEFINE STGCLASS

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 "DISPLAY QUEUE" on page 446 for more information about active association of a queue to a page set.

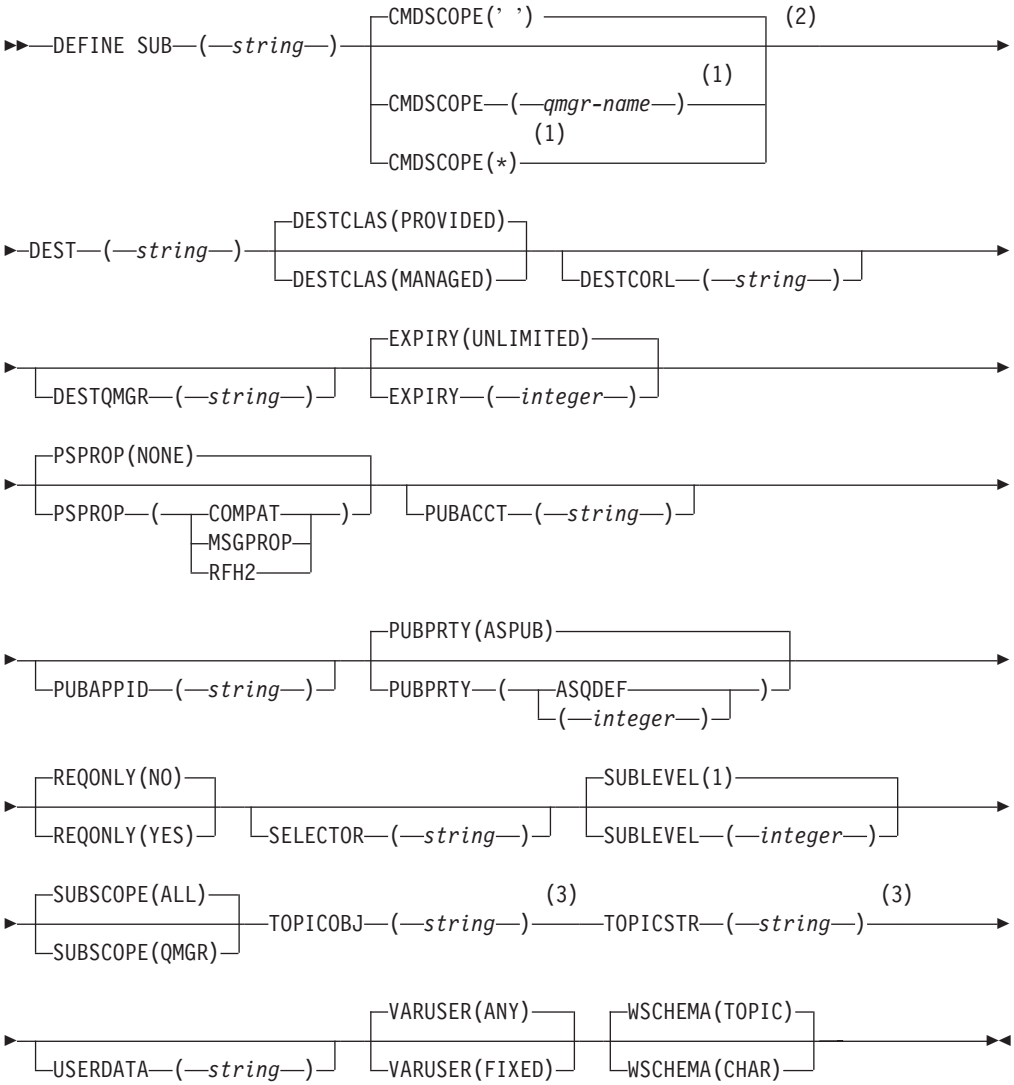
DEFINE SUB

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

Use DEFINE SUB to allow a legacy application to participate in a pub/sub application by allowing the administrative creation of a subscription.

Synonym: DEF SUB

DEFINE SUB



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

3 At least one of TOPICSTR and TOPICOBJ must be present on DEFINE.

Parameter descriptions for DEFINE SUB

Describes the DEFINE SUB (DEF SUB) parameters.

The following standard **DEFINE** command modifiers are available on **DEF SUB**:

- **REPLACE / NOREPLACE**
- **LIKE**
- On z/OS only, **CMDSCOPE** with usual options for a **DEFINE** command.

(string)
A mandatory parameter. Specifies the unique name for this subscription, see **SUBNAME** property.

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

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

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

qmgr-name
The command is processed 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 processed 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.

You cannot use CMDSCOPE as a filter keyword.

DEST(*string***)**
The destination for messages published to this subscription; this is the name of a queue.

DESTCLAS
System managed destination.

PROVIDED
The destination is a queue.

MANAGED
The destination is managed.

DESTCORL(*string***)**
The *CorrelId* used for messages published to this subscription.

DESTQMGR(*string***)**
The destination queue manager for messages published to this subscription.

DURABLE
A durable subscription is not deleted when the creating application closes its subscription handle.

NO The subscription is removed when the application that created it, is closed or disconnected from the queue manager.

YES The subscription persists even when the creating application is no longer running or has been disconnected. The subscription is reinstated when the queue manager restarts.

EXPIRY
The time to expiry from the creation date and time.

(*integer***)**
The time to expiry, in tenths of a second, from the creation date and time.

UNLIMITED
There is no expiry time.

PSPROP
The manner in which publish subscribe related message properties are added to messages sent to this subscription.

NONE
Do not add publish subscribe properties to the message. This is the default value supplied with the product.

COMPAT
Publish subscribe properties are added as PCF attributes.

MSGPROP
Publish subscribe properties are added within an MQRFH version 1 header.

RFH2 Publish subscribe properties are added within an MQRFH version 2 header.

PUBACCT(*string***)**
Accounting token passed by the subscriber, for propagation into messages published to this subscription in the *AccountingToken* field of the MQMD.

PUBAPPID(*string***)**
Identity data passed by the subscriber, for propagation into messages published to this subscription in the *ApplIdentityData* field of the MQMD.

PUBPRTY
The priority of the message sent to this subscription.

ASPUB
Priority of the message sent to this subscription is taken from that supplied in the published message.

ASQDEF
Priority of the message sent to this subscription is taken from the default priority of the queue defined as a destination.

(integer)
An integer providing an explicit priority for messages published to this subscription.

REQONLY
Indicates whether the subscriber polls for updates using the MQSUBRQ API call, or whether all publications are delivered to this subscription.

NO All publications on the topic are delivered to this subscription.

YES Publications are only delivered to this subscription in response to an MQSUBRQ API call.

SELECTOR(*string***)**
A selector applied to messages published to the topic.

SUBLEVEL(*integer***)**
The level within the subscription hierarchy at which this subscription is made. The range is zero through 9. The default value is 1.

SUBSCOPE
Determines whether this subscription should be forwarded to other brokers, so that the subscriber receives messages published at those other brokers.

ALL The subscription is forwarded to all queue managers directly connected through a publish subscribe collective or hierarchy.

QMGR
The subscription forwards messages published on the topic only within this queue manager.

TOPICSTR(*string***)**
Specifies a fully qualified topic name, or a topic set using wildcard characters for the subscription.

TOPICOBJ(*string***)**
The name of a topic object used by this subscription.

USERDATA(*string***)**
Specifies the user data associated with the subscription. The string is a variable length value that can be retrieved by the application on an MQSUB API call and passed in a message sent to this subscription as a message property.

VARUSER
Specifies whether a user other than the subscription creator can connect to and take over ownership of the subscription.

ANY Any user can connect to and takeover ownership of the subscription.

FIXED
Takeover by another **USERID** is not permitted.

WSCHEMA
The schema to be used when interpreting any wildcard characters in the topic string.

CHAR
Wildcard characters represent portions of strings.

TOPIC
Wildcard characters represent portions of the topic hierarchy. This is the default value supplied with the product.

Usage notes for DEFINE SUB

1. You must provide the following when you define a subscription:
   - The SUBNAME
   - A destination for messages
   - The topic to which the subscription applies
2. You can provide the topic name in the following ways:

   TOPICSTR
   The topic is fully specified as the TOPICSTR attribute.

   TOPICOBJ
   The topic is obtained from the TOPICSTR attribute of the named topic object. The named topic object is retained as the TOPICOBJ attribute of the new subscription. This method is provided to help you enter long topic strings through an object definition.

   TOPICSTR and TOPICOBJ
   The topic is obtained by the concatenation of the TOPICSTR attribute of the named topic object and the value of TOPICSTR (see the MQSUB API specification for concatenation rules). The named topic object is retained as the TOPICOBJ attribute of the new subscription.
3. If you specify TOPICOBJ, the parameter must name a WebSphere MQ topic object. The existence of the named topic object is checked at the time the command processes.
4. You can explicitly specify the destination for messages through the use of the DEST and DESTQMGR keywords.

   Note that you must provide the DEST keyword for the default option of DESTCLAS(PROVIDED); a managed destination is created if you specify DESTCLAS(MANAGED).
5. On z/OS only, at the time the DEF SUB command processes, no check is performed that the named DEST or DESTQMGR exists.

   These names are used at publishing time as the *ObjectName* and *ObjectQMgrName* for an MQOPEN call. These names are resolved according to the WebSphere MQ name resolution rules.

DEFINE TOPIC

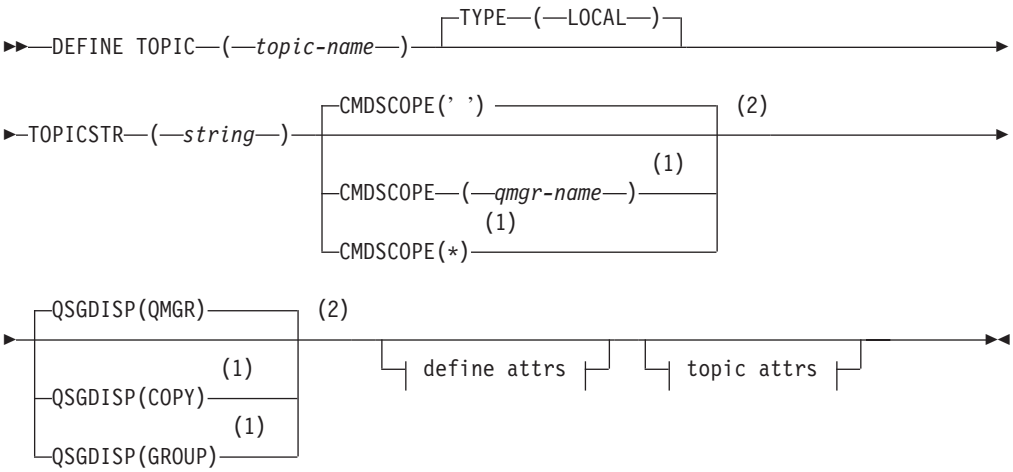
Use DEFINE TOPIC to define a new WebSphere MQ administrative topic node in a topic tree, and set its parameters.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

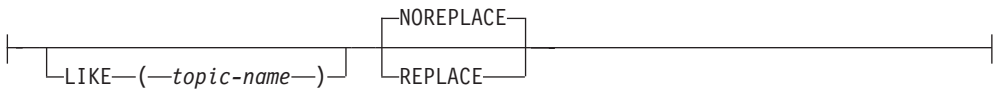
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DEF TOPIC

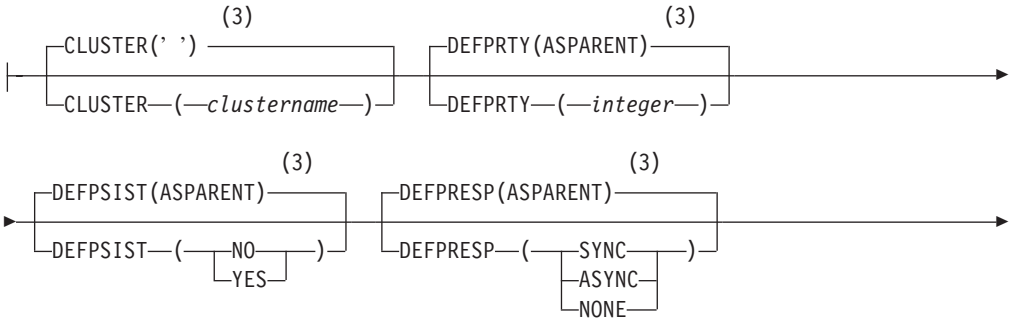
DEFINE TOPIC

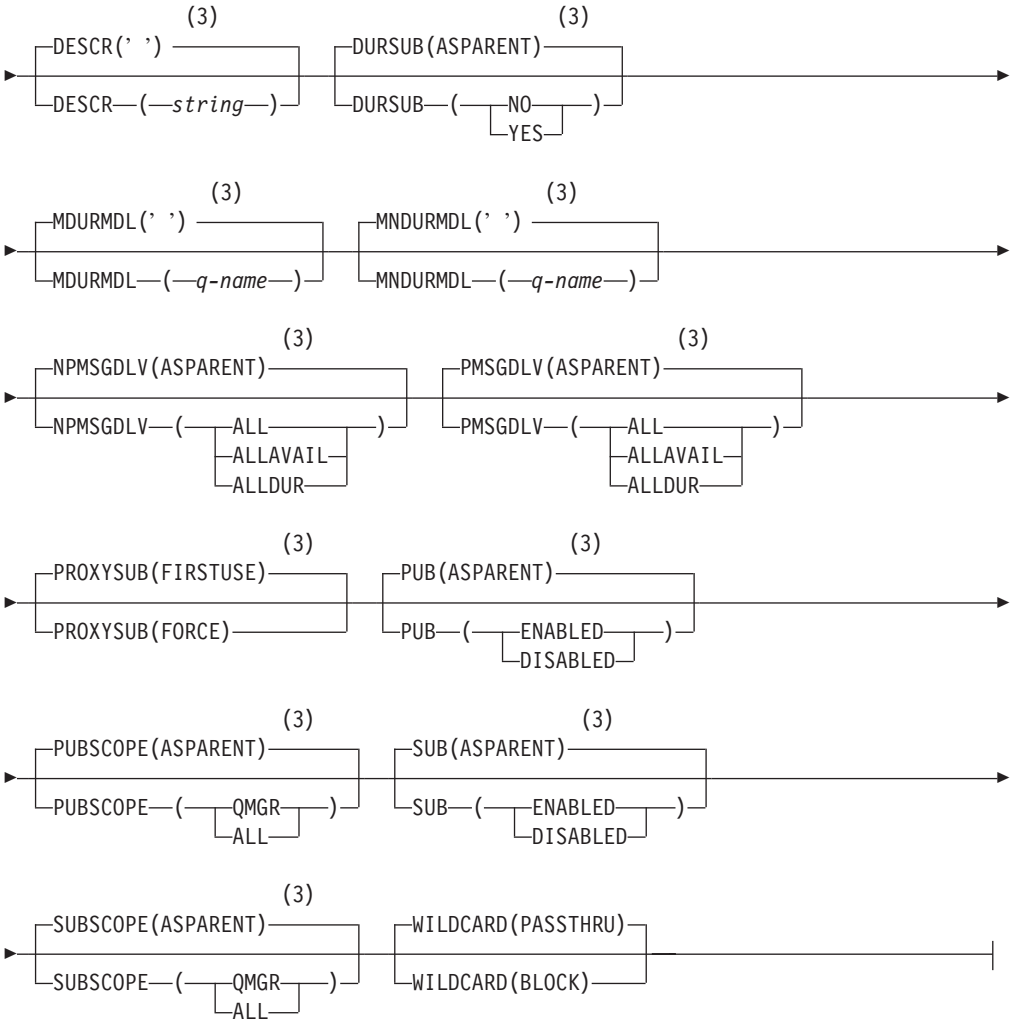


Define attrs:



Topic attrs:





Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

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

Parameter descriptions for DEFINE TOPIC

This describes the parameters for the MQSC command DEFINE TOPIC.

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

- The **LIKE** parameter applies only to the DEFINE TOPIC command.
- The **REPLACE** and **NOREPLACE** parameter applies only to the DEFINE TOPIC command.
- The **TOPICSTR** parameter applies only to the **DEFINE TOPIC** command.

(topic-name)

Name of the WebSphere MQ topic definition (see "Rules for naming WebSphere MQ objects" on page 5). The maximum length is 48 characters. This is required.

The name must not be the same as any other topic definition currently defined on this queue manager (unless REPLACE is specified).

CLUSTER

The name of the cluster to which this topic belongs.

' ' This topic does not belong to a cluster. Publications and subscriptions for this topic are not propagated to publish/subscribe cluster-connected queue managers.

string The topic belongs to this cluster.

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.

DEFPRTY(*integer***)**

The default priority of messages published to the topic.

(*integer***)**

The value must be in the range zero (the lowest priority), through to the MAXPRTY queue manager parameter (MAXPRTY is 9).

ASPARENT

The default priority is based on the setting of the closest parent administrative topic object in the topic tree. This is the default supplied with WebSphere MQ, but your installation might have changed it.

DEFPSIST

Specifies the message persistence to be used when applications specify the MQPER_PERSISTENCE_AS_TOPIC_DEF option.

ASPARENT

The default persistence is based on the setting of the closest parent administrative topic object in the topic tree. This is the default supplied with WebSphere MQ, but your installation might have changed it.

NO Messages on this queue are lost during a restart of the queue manager.

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.

DEFPRESP
Specifies the put response to be used when applications specify the MQPMO_RESPONSE_AS_DEF option.

ASPARENT
The default put response is based on the setting of the closest parent administrative topic object in the topic tree. This is the default supplied with WebSphere MQ, but your installation might have changed it.

SYNC Put operations to the queue that specify MQPMO_RESPONSE_AS_Q_DEF are issued as if MQPMO_SYNC_RESPONSE had been specified instead. Fields in the MQMD and MQPMO are returned by the queue manager to the application.

ASYNC
Put operations to the queue that specify MQPMO_RESPONSE_AS_Q_DEF are always issued as if MQPMO_ASYNC_RESPONSE had been specified instead. Some fields in the MQMD and MQPMO are not returned by the queue manager to the application; but an improvement in performance might be seen for messages put in a transaction and any non-persistent messages

DESCR(*string***)**
Plain-text comment. It provides descriptive information about the object when an operator issues the DISPLAY TOPIC 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.

DURSUB
Specifies whether applications are permitted to make durable subscriptions on this topic.

ASPARENT
Whether durable subscriptions can be made on this topic is based on the setting of the closest parent administrative topic object in the topic tree. This is the default supplied with WebSphere MQ, but your installation might have changed it.

NO Durable subscriptions cannot be made on this topic.

YES Durable subscriptions can be made on this topic.

LIKE(*topic-name***)**
The name of a topic, whose parameters will be used to model this definition.

This parameter applies only to the DEFINE TOPIC 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 topics on this queue manager.

This is equivalent to specifying:

```
LIKE(SYSTEM.DEFAULT.TOPIC)
```

A default topic 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 5.

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

Note:

1. QSGDISP (GROUP) objects are not searched.
2. LIKE is ignored if QSGDISP(COPY) is specified.

MDURMDL(*string***)**
The name of the model queue to be used for durable subscriptions that request that the queue manager manages the destination of its publications (see "Rules for naming WebSphere MQ objects" on page 5). The maximum length is 48 characters.

If MDURMDL is blank, it operates in the same way as ASPARENT values on other attributes. The name of the model queue to be used will be based on the closest parent administrative topic object in the topic tree with a value set for MDURMDL.

The dynamic queue created from this model will have a prefix of SYSTEM.MANAGED.DURABLE

MNDURMDL(*string***)**
The name of the model queue to be used for non-durable subscriptions that request that the queue manager manages the destination of its publications (see "Rules for naming WebSphere MQ objects" on page 5). The maximum length is 48 characters.

If MNDURMDL is blank, it operates in the same way as ASPARENT values on other attributes. The name of the model queue to be used will be based on the closest parent administrative topic object in the topic tree with a value set for MNDURMDL.

The dynamic queue created from this model will have a prefix of SYSTEM.MANAGED.NDURABLE.

NPMSGDLV
The delivery mechanism for persistent messages published to this topic:

ASPARENT
The delivery mechanism used is based on the setting of the first parent administrative node found in the topic tree relating to this topic. This is the default supplied with WebSphere MQ, but your installation might have changed it.

ALL Non-persistent messages must be delivered to all subscribers, irrespective of durability for the MQPUT call to report success. If a delivery failure to any subscriber occurs, no other subscribers receive the message and the MQPUT call fails.

ALLAVAIL
Non-persistent messages are delivered to all subscribers that can accept the message. Failure to deliver the message to any subscriber does not prevent other subscribers from receiving the message.

ALLDUR
Non-persistent messages must be delivered to all durable subscribers. Failure to deliver a non-persistent message to any non-durable subscribers does not return an error to the MQPUT call. If a delivery failure to a durable subscriber occurs, no subscribers receive the message and the MQPUT calls fails.

PMSGDLV
The delivery mechanism for persistent messages published to this topic:

ASPARENT
The delivery mechanism used is based on the setting of the first parent administrative node found in the topic tree relating to this topic. This is the default supplied with WebSphere MQ, but your installation might have changed it.

ALL Persistent messages must be delivered to all subscribers, irrespective of durability for the MQPUT call to report success. If a delivery failure to any subscriber occurs, no other subscribers receive the message and the MQPUT call fails.

ALLAVAIL
Persistent messages are delivered to all subscribers that can accept the message. Failure to deliver the message to any subscriber does not prevent other subscribers from receiving the message.

ALLDUR
Persistent messages must be delivered to all durable subscribers. Failure to deliver a persistent message to any non-durable subscribers does not return an error to the MQPUT call. If a delivery failure to a durable subscriber occurs, no subscribers receive the message and the MQPUT calls fails.

PROXYSUB
Controls whether a proxy subscription can be sent for this topic to directly connected queue managers, even if no local subscriptions exist.

FIRSTUSE
A proxy subscription is sent for this topic only when a local subscription exists or a proxy subscription is received that should be propagated to further directly-connected queue managers.

FORCE
A proxy subscription is sent to connected queue managers even if no local subscriptions exist.

Note: The proxy subscription is sent when this value is set on DEFINE or ALTER.

PUB Controls whether messages can be published to this topic.

ASPARENT
Whether messages can be published to the topic is based on the setting of the closest parent administrative topic object in the topic

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

ENABLED
Messages can be published to the topic (by suitably authorized applications).

DISABLED
Messages cannot be published to the topic.

PUBSCOPE
Determines whether this queue manager propagates publications to queue managers as part of a hierarchy or as part of a publish/subscribe cluster.

Note: You can restrict the behavior on a publication-by-publication basis, using MQPMO_SCOPE_QMGR on the Put Message options.

ASPARENT
Whether this queue manager propagates publications to queue managers as part of a hierarchy or as part of a publish/subscribe cluster is based on the setting of the first parent administrative node found in the topic tree that relates to this topic.

QMGR
Publications for this topic are not propagated to connected queue managers.

ALL Publications for this topic are propagated to hierarchically connected queue managers and to publish/subscribe cluster-connected queue managers.

QSGDISP
This parameter applies to z/OS only.

Specifies the disposition of the object within the group.

QSGDISP	ALTER	DEFINE
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.	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.

QSGDISP	ALTER	DEFINE

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 attempt to refresh local copies on page set zero:
	`DEFINE TOPIC(name)`
	`REPLACE QSGDISP(COPY)`
	
	The ALTER for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.
		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 attempt to make or refresh local copies on page set zero:
		`DEFINE TOPIC(name)`
		`REPLACE QSGDISP(COPY)`
		
		The DEFINE for the group object takes effect regardless of whether the generated command with QSGDISP(COPY) fails.

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.	Not permitted.
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. 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.

This parameter applies to only the DEFINE TOPIC command.

(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.

NOREPLACE
The definition should not replace any existing definition of the object.

SUB Controls whether applications are to be permitted to subscribe to this topic.

ASPARENT
Whether applications can subscribe to the topic is based on the setting of the closest parent administrative topic object in the topic tree. This is the default supplied with WebSphere MQ, but your installation might have changed it.

ENABLED
Subscriptions can be made to the topic (by suitably authorized applications).

DISABLED
Applications cannot subscribe to the topic.

SUBSCOPE
Determines whether this queue manager subscribes to publications in this queue manager or in the network of connected queue managers. If subscribing to all queue managers, the queue manager will propagate subscriptions to them as part of a hierarchy or as part of a publish/subscribe cluster.

Note: You can restrict the behavior on a subscription-by-subscription basis, using MQPMO_SCOPE_QMGR on the Subscription Descriptor or SUBSCOPE(QMGR) on DEFINE SUB.

ASPARENT
Whether this queue manager will subscribe to publications in the same way as the setting of the first parent administrative node found in the topic tree relating to this topic.

QMGR
Only publications that are published on this queue manager will reach the subscriber.

ALL A publication made on this queue manager or on another queue manager will reach the subscriber. Subscriptions for this topic are propagated to hierarchically connected queue managers and to publish/subscribe cluster-connected queue managers.

TOPICSTR(*string***)**
The topic string represented by this topic object definition. This parameter is required and cannot contain the empty string. The forward slash (/) character within this string has special meaning. It delimits the elements in the topic tree. A topic string can start with the "/" character but is not required to. A string starting with a "/" is not the same as a string that starts without a "/" character.

The topic string must not be the same as any other topic string already represented by a topic object definition.

The maximum length of the string is 10 240 characters.

TYPE (topic-type)
This is optional. If used it must follow immediately after the *topic-name* parameter on all platforms except z/OS.

LOCAL
A local topic object.

WILDCARD
The behavior of wildcard subscriptions with respect to this topic.

PASSTHRU
Subscriptions made to a wildcarded topic less specific than the topic string at this topic object will receive publications made to this topic and to topic strings more specific than this topic. This is the default value.

BLOCK
Subscriptions made to a wildcarded topic less specific than the topic string at this topic object will not receive publications made to this topic or to topic strings more specific than this topic.

This value of this attribute is used when subscriptions are defined. If you alter this attribute, the set of topics covered by existing subscriptions is not affected by the modification. This applies also, if the topology is changed when topic objects are created or deleted; the set of topics matching subscriptions created following the modification of the WILDCARD attribute is created using the modified topology. If you want to force the matching set of topics to be reevaluated for existing subscriptions, you must restart the queue manager.

Usage notes for DEFINE TOPIC

These are the usage notes for the MQSC command DEFINE TOPIC.

- When an attribute has the value ASPARENT, the value is taken from the setting of the first parent administrative node that is found in the topic tree. Administered nodes are based on either locally defined topic objects or remotely defined cluster topics when participating in a publish/subscribe cluster. If the first parent topic object also has the value ASPARENT, the next object is looked for. If every object that is found, when looking up the tree, uses ASPARENT, the values are taken from the SYSTEM.BASE.TOPIC, if it exists. If SYSTEM.BASE.TOPIC does not exist, the values are the same as those supplied with WebSphere MQ in the definition of the SYSTEM.BASE.TOPIC.
- The ASPARENT attribute is applied at each queue manager in the cluster collective by inspecting the set of local definitions and cluster definitions that is visible in the queue manager at the time.
- When a publication is sent to multiple subscribers, the attributes used from the topic object are used consistently for all subscribers that receive the publication. For example, inhibiting publication on a topic will apply for the next application MQPUT to the topic. A publication that is in progress to multiple subscribers will complete to all subscribers and will not take note of a change that has happened, part of the way through, to any attribute on the topic.

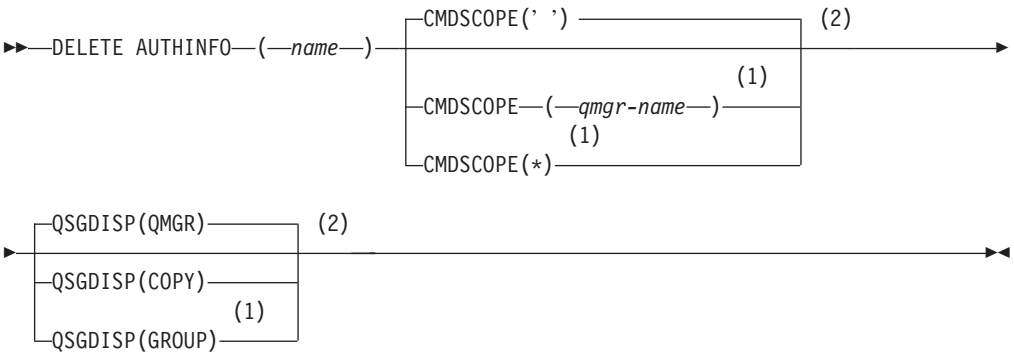
DELETE AUTHINFO

Use MQSC command DELETE AUTHINFO to delete an authentication information object.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

Synonym: None

DELETE 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.

Parameter descriptions for DELETE AUTHINFO

(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.

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 AUTHINFO(name) QSGDISP(COPY)
```

The deletion of the group object takes effect even if the generated command with QSGDISP(COPY) fails.

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 BUFFPOOL

Use the MQSC command DELETE BUFFPOOL to delete a buffer pool that is used for holding messages in main storage.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DEL BP

DELETE BUFFPOOL

Parameter descriptions for DELETE BUFFPOOL

(integer)

This is the number of the buffer pool to be deleted. The value is a number in the range 0 – 15.

Usage note for DELETE BUFFPOOL

- Ensure there are no current page set definitions using the named buffer pool, otherwise the command will fail.

DELETE CFSTRUCT

Use the MQSC command DELETE CFSTRUCT to delete a CF application structure definition.

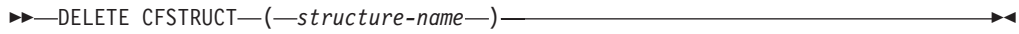
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

This command is valid only z/OS when the queue manager is a member of a queue-sharing group.

Synonym: None

DELETE CFSTRUCT



Keyword and parameter descriptions for DELETE CFSTRUCT

(structure-name)

The name of the CF structure definition to be deleted. The name must be defined within the queue sharing group.

Usage notes for DELETE CFSTRUCT

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.

DELETE CHANNEL

Use the MQSC command DELETE CHANNEL to delete a channel definition.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	CR

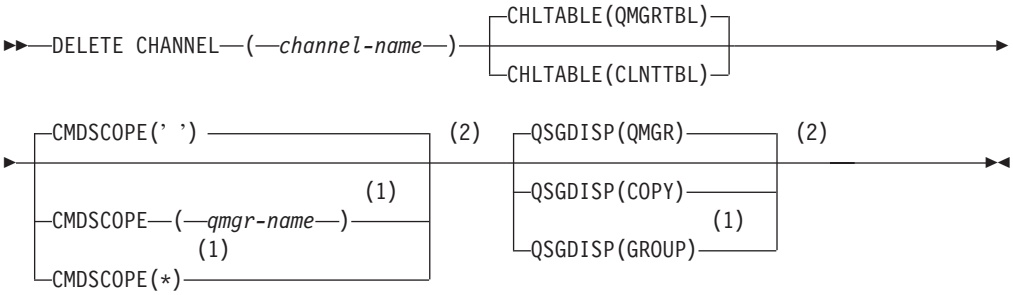
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Notes for z/OS users:

1. 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.
2. You can only delete cluster-sender channels that have been created manually.

Synonym: DELETE CHL

DELETE CHANNEL



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for DELETE CHANNEL

(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 HP OpenVMS, i5/OS, 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)
```

The deletion of the group object takes effect even if the generated command with QSGDISP(COPY) fails.

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 LISTENER

Use the MQSC command DELETE LISTENER to delete a listener definition.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	

Synonym: DELETE LSTR

DELETE LISTENER

Keyword and parameter descriptions for DELETE LISTENER

(listener-name)

The name of the listener definition to be deleted. This is required. The name must be that of an existing listener defined on the local queue manager.

Usage notes for DELETE LISTENER

1. The command fails if an application has the specified listener object open, or if the listener is currently running.

DELETE NAMELIST

Use the MQSC command DELETE NAMELIST to delete a namelist definition.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

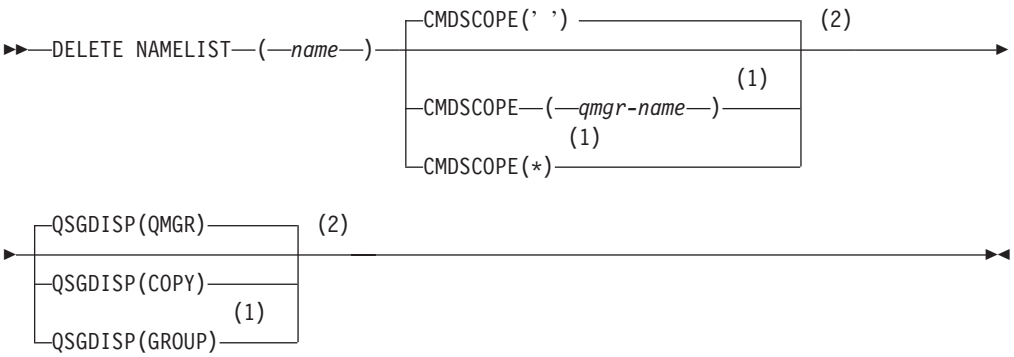
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

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

Synonym: DELETE NL

DELETE NAMELIST



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for 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. 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)
```

The deletion of the group object takes effect even if the generated command with QSGDISP(COPY) fails.

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 PROCESS

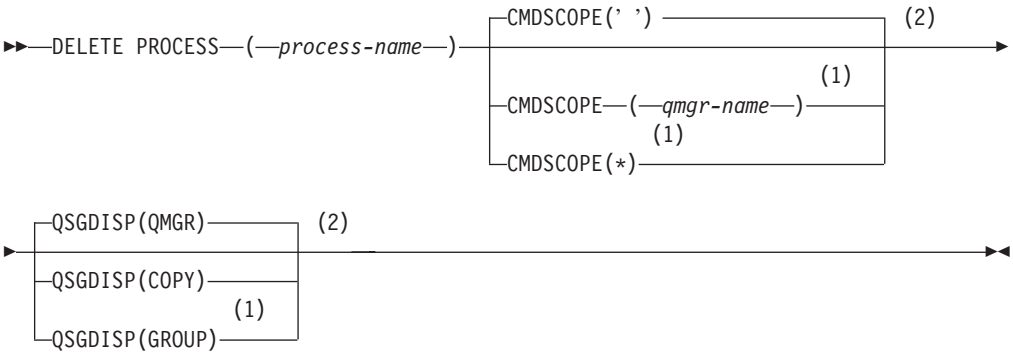
Use the MQSC command DELETE PROCESS to delete a process definition.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DELETE PRO

DELETE PROCESS



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for DELETE PROCESS

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.

***** 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)
```

The deletion of the group object takes effect even if the generated command with QSGDISP(COPY) fails.

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 PSID

Use the MQSC command DELETE PSID to delete a page set. This command closes the page set and de-allocates it from the queue manager.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DEL PSID

DELETE PSID

Parameter descriptions for DELETE PSID

(psid-number)
Identifier of the page set. This is required. You cannot delete page set 0.

Usage notes for DELETE PSID

1. The identified page set must have no storage class (STGCLASS) referencing it.
2. If the page set still has buffers in the buffer pool when you issue this command, the command fails and an error message is issued. You cannot delete the page set until 3 checkpoints have been completed since the page set was emptied.
3. If the page set is not to be used again by the queue manager, update the queue manager started task procedure JCL, and remove the corresponding DEFINE PSID command from the CSQINP1 initialization data set. If the page set had a dedicated buffer pool, remove its definitions also from CSQINP1.
4. If you wish to reuse the data set again as a page set, format it before doing so.

DELETE queues

This section contains the following commands:

- "DELETE QALIAS"
- "DELETE QLOCAL"
- "DELETE QMODEL" on page 302
- "DELETE QREMOTE" on page 302

These commands are supported on the following platforms:

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

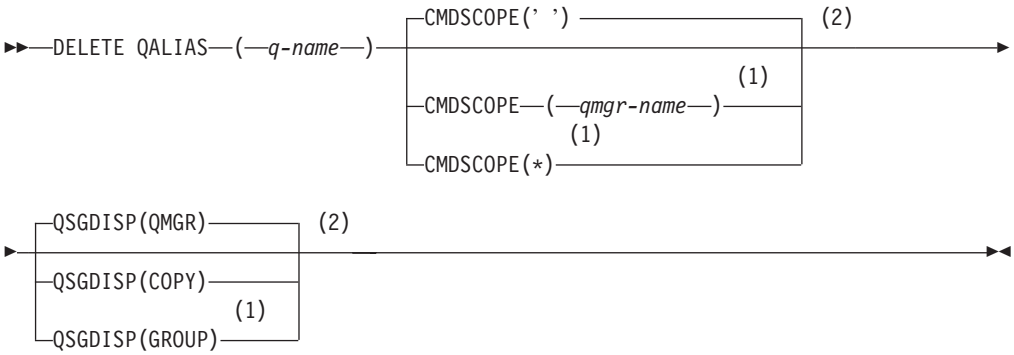
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

DELETE QALIAS

Use DELETE QALIAS to delete an alias queue definition.

Synonym: DELETE QA

DELETE QALIAS



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

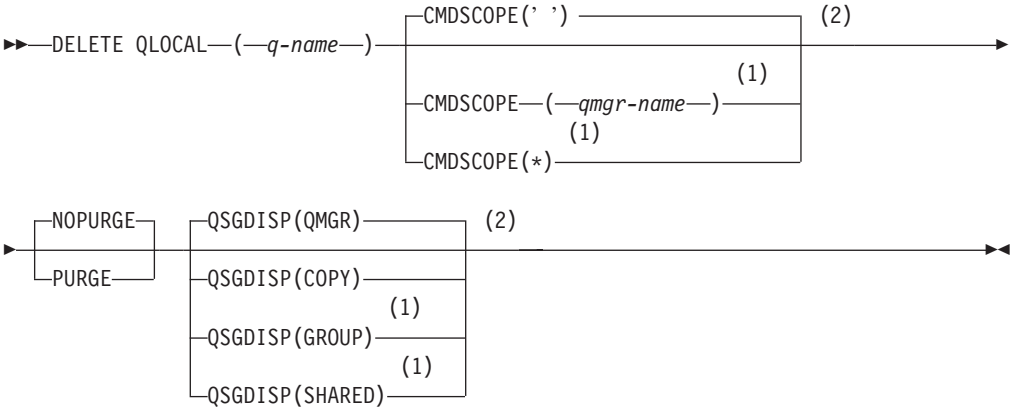
2 Valid only on z/OS.

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



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

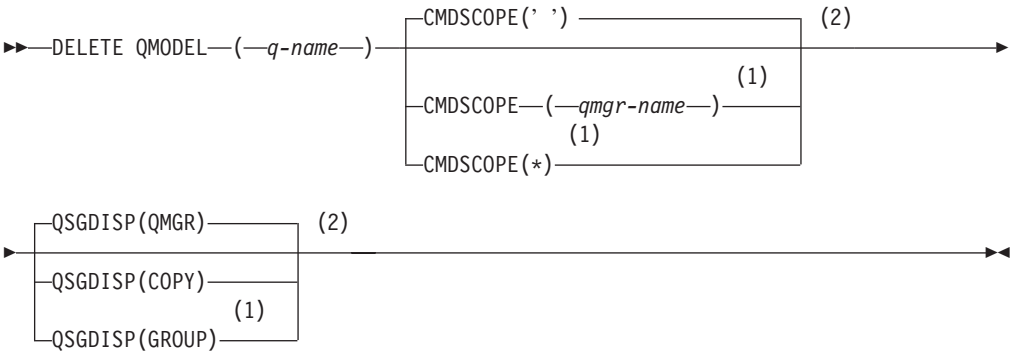
2 Valid only on z/OS.

DELETE QMODEL

Use DELETE QMODEL to delete a model queue definition.

Synonym: DELETE QM

DELETE QMODEL



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

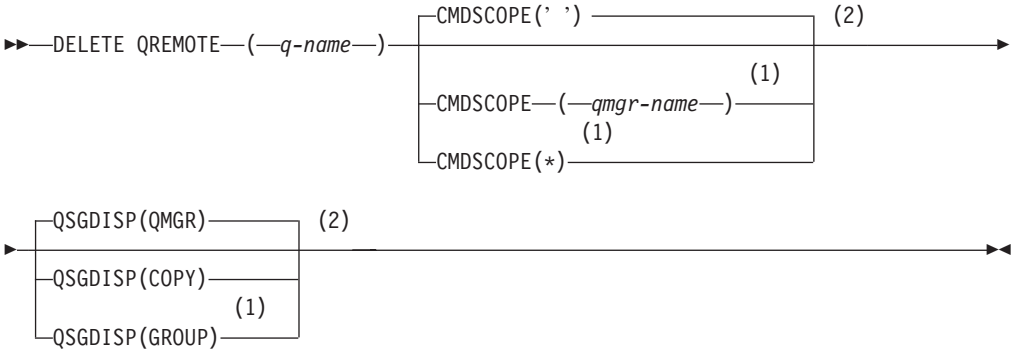
2 Valid only on z/OS.

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 QR

DELETE QREMOTE



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for DELETE queues

(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.

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)
```

The deletion of the group object takes effect even if the generated command with QSGDISP(COPY) fails.

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.

DELETE SERVICE

Use the MQSC command DELETE SERVICE to delete a service definition.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	

Synonym:

DELETE SERVICE

Keyword and parameter descriptions for DELETE SERVICE

(service-name)

The name of the service definition to be deleted. This is required. The name must be that of an existing service defined on the local queue manager.

Usage notes for DELETE SERVICE

1. The command fails if an application has the specified service object open, or if the service is currently running.

DELETE SUB

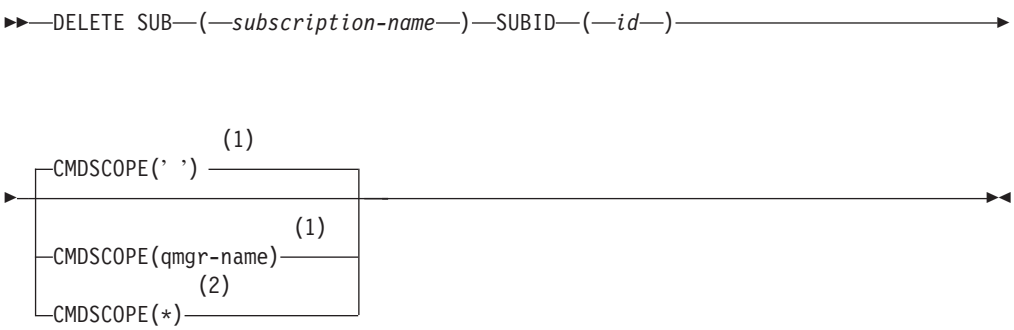
Use the MQSC command DELETE SUB to remove a durable subscription from the system. For a managed destination, any unprocessed messages left on the destination are removed.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DEL SUB

DELETE SUB



Notes:

1 Valid only on z/OS.

2 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for DELETE SUB

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 processed on the queue manager on which it was entered. This is the default value.

qmgr-name
The command is processed 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 processed 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.

You cannot use CMDSCOPE as a filter keyword.

SUBID(*string***)**
The internal, unique key identifying a subscription.

DELETE STGCLASS

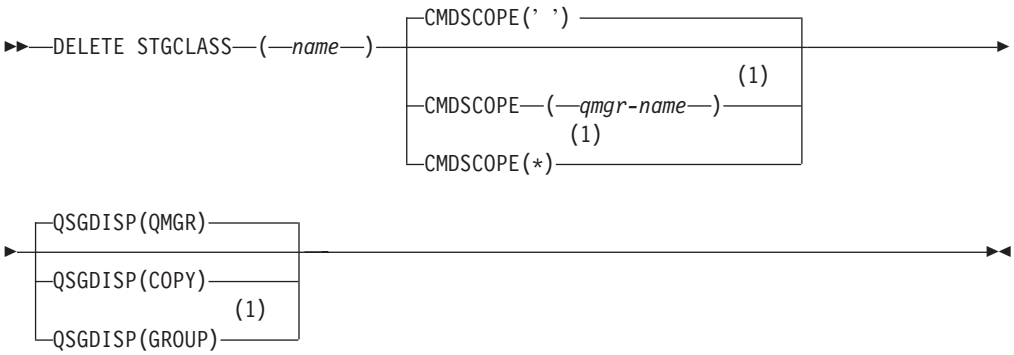
Use the MQSC command DELETE STGCLASS to delete a storage class definition.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DELETE STC

DELETE STGCLASS



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for DELETE STGCLASS

You must specify which storage class definition you want to delete.

All queues that use the storage class must be empty and closed.

(name) The name of the storage class definition to be deleted. The name must be defined to the local queue manager.

The command fails unless all queues referencing the storage class are empty and closed.

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 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 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)
```

The deletion of the group object takes effect even if the generated command with QSGDISP(COPY) fails.

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 TOPIC

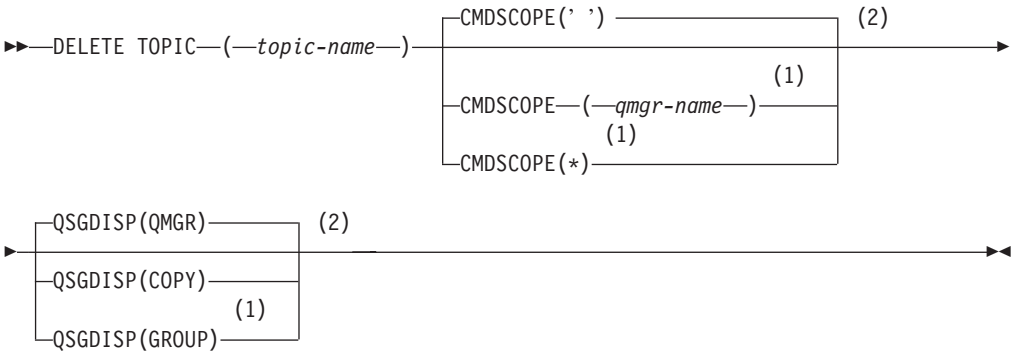
Use DELETE TOPIC to delete a WebSphere MQ administrative topic node.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: None

DELETE TOPIC



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for DELETE TOPIC

(topic-name)
The name of the administrative topic object to be deleted. This parameter is required.

The name must be that of an existing administrative topic object.

CMDSCOPE
This parameter applies to only z/OS 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 make, or delete, local copies on page set zero:

```
DELETE TOPIC(topic-name) QSGDISP(COPY)
```

The deletion of the group object takes effect even if the generated command with QSGDISP(COPY) fails.

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.

Usage note

- Deleting an administrative topic node does not remove the non-administrative topic associated with it (represented by the TOPICSTR parameter). The non-administrative topic is removed by the queue manager when:
  - there are no applications publishing to or subscribing to that topic, and
  - the interval specified by the TREELIFE parameter on the queue manager object has elapsed.

DISPLAY ARCHIVE

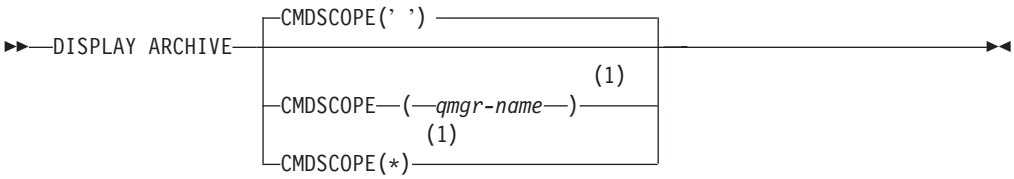
Use the MQSC command DISPLAY ARCHIVE to display archive system parameters and information.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS ARC

DISPLAY ARCHIVE



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for DISPLAY ARCHIVE

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 for DISPLAY ARCHIVE

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).
   - 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.

   For more details of these parameters, see “SET ARCHIVE” on page 550.
2. This command is issued internally by WebSphere MQ at the end of queue manager startup.

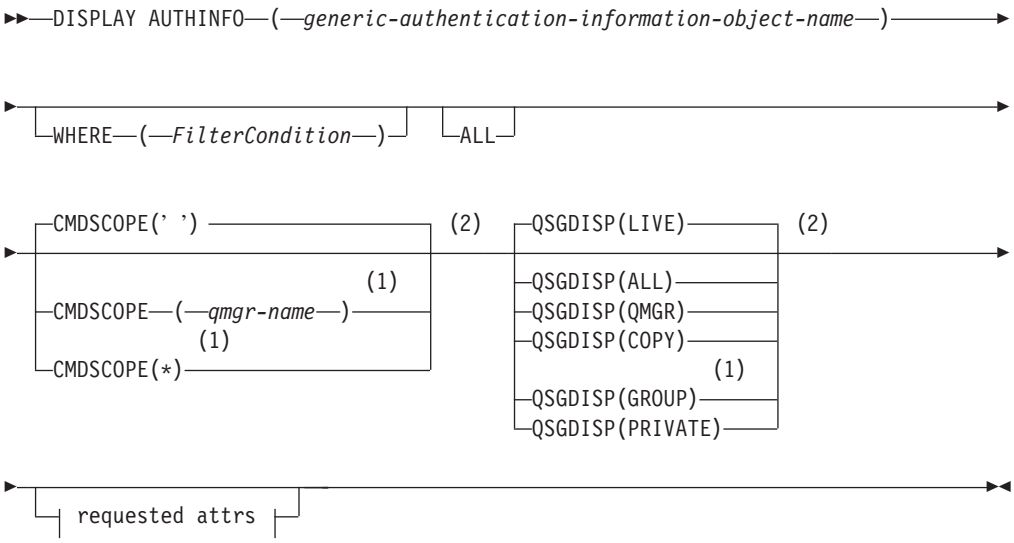
DISPLAY AUTHINFO

Use the MQSC command DISPLAY AUTHINFO to display the attributes of an authentication information object.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

Synonym: DIS AUTHINFO

DISPLAY AUTHINFO



Requested attrs:

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 DISPLAY AUTHINFO

(generic-authentication-information-object-name)
The name of the authentication information object to be displayed (see

"Rules for naming WebSphere MQ objects" on page 5). 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.

WHERE

Specify a filter condition to display only those authentication information objects that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword

Almost any parameter that can be used to display attributes for this DISPLAY command. However, you cannot use the CMDSCOPE or QSGDISP parameters as filter keywords.

operator

This is used to determine whether an authentication information object satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

filter-value

The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.

  You can use any of the operators except LK and NL.
- A generic value. This is a character string (such as the character string you supply for the DESCR parameter) with an asterisk at the end, for example ABC*. The characters must be valid for the attribute you are testing. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed. You cannot use a generic filter-value with numeric values.

  You can only use operators LK or NL for generic values on the DISPLAY AUTHINFO command.

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.

On z/OS this is also the default if you specify a filter condition using the WHERE parameter, but on other platforms only requested attributes are displayed.

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.

You cannot use CMDSCOPE as a filter keyword.

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:

QMGR
The object was defined with QSGDISP(QMGR).

GROUP
The object was defined with QSGDISP(GROUP).

COPY The object was defined with QSGDISP(COPY).

You cannot use QSGDISP as a filter keyword.

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, and, on z/OS, their QSGDISPs, 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 hostname, IPv4 dotted decimal address, or IPv6 hexadecimal notation 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. If nonblank, this is displayed as asterisks (on all platforms except z/OS).

LDAPUSER
Distinguished Name of the user on the LDAP server

See "DEFINE AUTHINFO" on page 134 for more information about individual parameters.

DISPLAY CFSTATUS

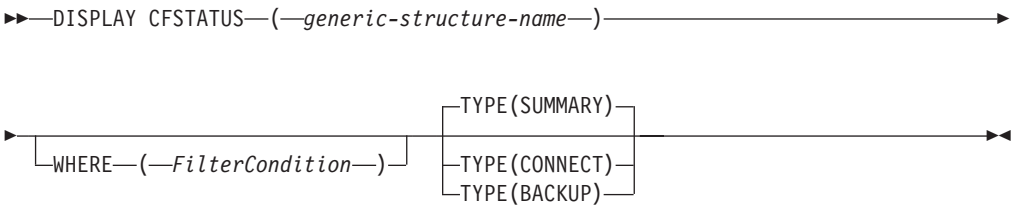
Use the MQSC command DISPLAY CFSTATUS to display the status of one or more CF application structures. This command is valid only on WebSphere MQ for z/OS when the queue manager is a member of a queue-sharing group.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS CFSTATUS

DISPLAY CFSTATUS



Keyword and parameter descriptions for DISPLAY CFSTATUS

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.

WHERE
Specify a filter condition to display status information for those CF

application structures that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Almost any parameter that is returned by this DISPLAY command. However, you cannot use the TYPE parameter as a filter keyword.

operator
This is used to determine whether a CF application structure satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

CT Contains a specified item. If the *filter-keyword* is a list, you can use this to display objects the attributes of which contain the specified item.

EX Does not contain a specified item. If the *filter-keyword* is a list, you can use this to display objects the attributes of which do not contain the specified item.

CTG Contains an item which matches a generic string that you provide as a *filter-value*. If the *filter-keyword* is a list, you can use this to display objects the attributes of which match the generic string.

EXG Does not contain any item which matches a generic string that you provide as a *filter-value*. If the *filter-keyword* is a list, you can use this to display objects the attributes of which do not match the generic string.

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.

  You can use operators LT, GT, EQ, NE, LE, GE, only. However, if the value is one from a possible set of values returnable on a parameter (for example, the value ACTIVE on the STATUS parameter), you can only use EQ or NE.
- A generic value. This is a character string (such as the character string in the QMNAME parameter) with an asterisk at the end, for example ABC*. The characters must be valid for the attribute you are testing. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.

- An item in a list of values. The value can be explicit or, if it is a character value, it can be explicit or generic. If it is explicit, use CT or EX as the operator. For example, if the value DEF is specified with the operator CT, all items where one of the attribute values is DEF are listed. If it is generic, use CTG or EXG as the operator. If ABC* is specified with the operator CTG, all items where one of the attribute values begins with ABC are listed.

TYPE Specifies the type of status information required to be displayed. Values are:

SUMMARY
Display summary status information for each application structure. This is the default.

CONNECT
Display connection status information for each application structure for each active queue manager.

BACKUP
Display backup status information for each application structure.

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.

UNKNOWN
The status of the CF structure is not known because, for example, DB2 may be unavailable.

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.

UNKNOWN
The status of the CF structure is not known.

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.

UNKNOWN
The status of the CF structure is not known.

QMNAME*(qmgrname)*
The name of the queue manager that took the last successful backup for this application structure.

BKUPTIME*(time)*
The end time of the last successful backup taken for this application structure. The format of this field is `hh.mm.ss`.

BKUPDATE*(date)*
The date of the last successful backup taken for this application structure. The format of this field is `yyyy-mm-dd`.

BKUPSIZE*(size)*
The size in megabytes of the last successful backup taken for this application structure.

BKUPSRBA*(hexadecimal)*
This is the backup dataset start RBA for the start of the last successful backup taken for this application structure.

BKUPERBA*(hexadecimal)*
This is the backup dataset end RBA for the end of the last successful backup taken for this application structure.

LOGS*(qmgrname-list)*
This is the list of queue managers, the logs of which are required to perform a recovery.

FAILTIME*(time)*

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().

FAILDATE*(date)*

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

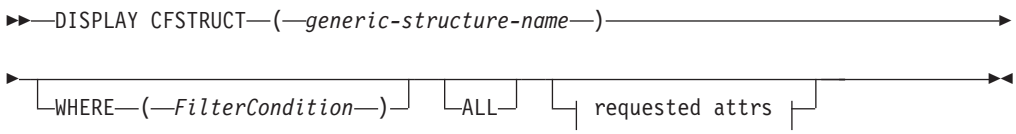
Use the MQSC command DISPLAY CFSTRUCT to display the attributes of one or more CF application structures. This command is valid only on z/OS when the queue manager is a member of a queue-sharing group.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

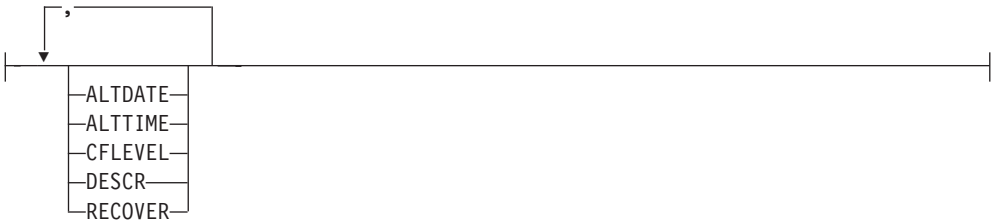
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS CFSTRUCT

DISPLAY CFSTRUCT



Requested attrs:



Keyword and parameter descriptions for DISPLAY CFSTRUCT

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.

WHERE
Specify a filter condition to display only those CF application structures that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Any parameter that can be used to display attributes for this DISPLAY command.

operator
This is used to determine whether a CF application structure satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.

  You can use any of the operators except LK and NL. However, if the value is one from a possible set of values returnable on a parameter (for example, the value YES on the RECOVER parameter), you can only use EQ or NE.
- A generic value. This is a character string (such as the character string you supply for the DESCR parameter) with an asterisk at the end, for example ABC*. The characters must be valid for the attribute you are testing. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed. You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.

  You can only use operators LK or NL for generic values on the DISPLAY CFSTRUCT command.

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`.

ALTTIME
The time at which the definition was last altered, in the form `hh.mm.ss`.

CFLEVEL
Indicates the functional capability level for this CF application structure.

DESCR
Descriptive comment.

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

NO
CF application structure recovery is not supported.

YES
CF application structure recovery is supported.

Usage notes for DISPLAY CFSTRUCT

1. The command cannot specify the CF administration structure (CSQ_ADMIN).

DISPLAY CHANNEL

Use the MQSC command DISPLAY CHANNEL to display a channel definition.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

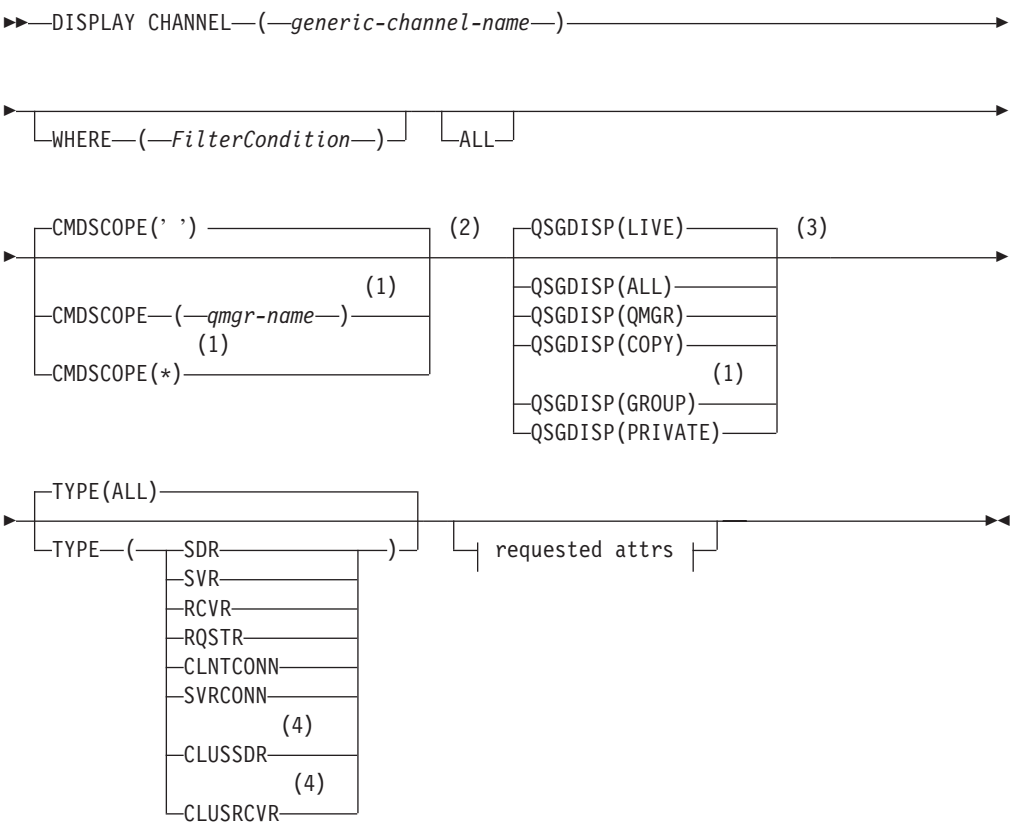
The values shown describe the current definition of the channel. If the channel has been altered since it was started, any currently running instance of the channel object may not have the same values as the current definition.

Note:

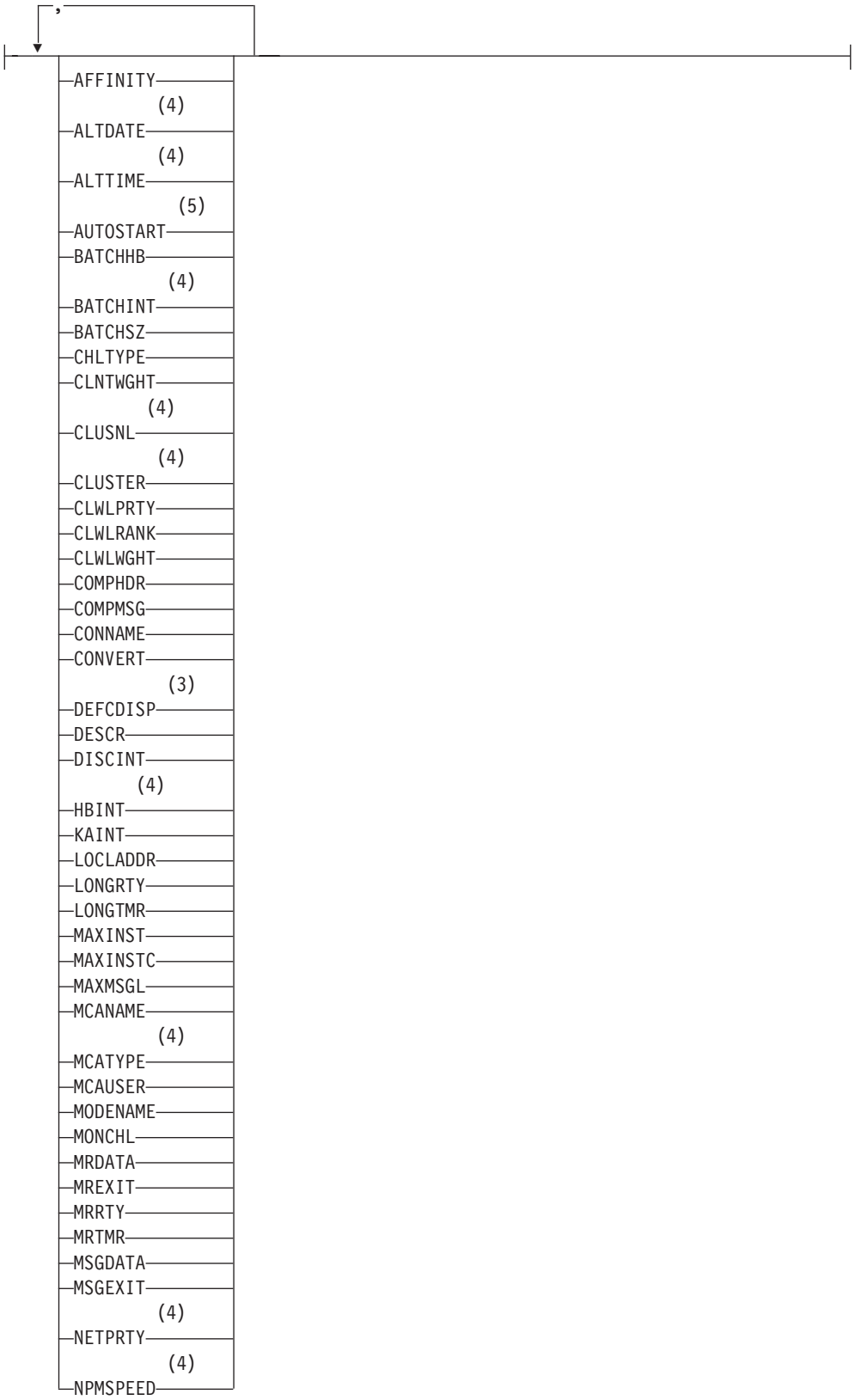
1. You can only display cluster-sender channels if they were created manually.

Synonym: DIS CHL

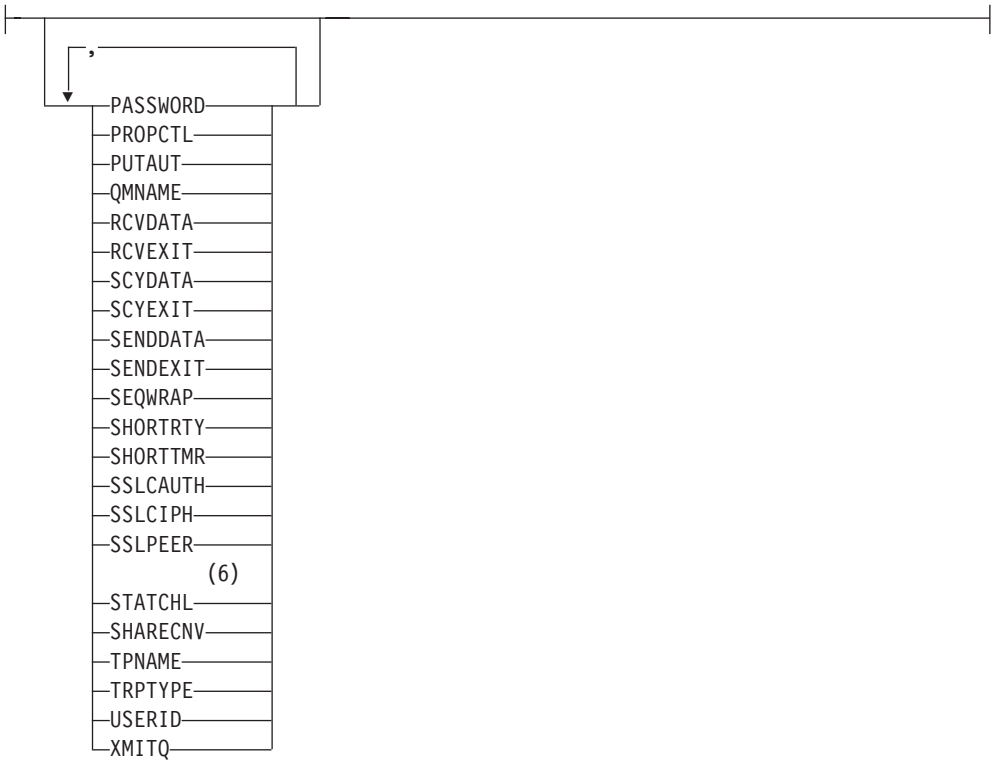
DISPLAY CHANNEL



Requested attrs:


,
AFFINITY
(4)
ALTDATE
(4)
ALTTIME
(5)
AUTOSTART
BATCHHB
(4)
BATCHINT
BATCHSZ
CHLTYPE
CLNTWGHT
(4)
CLUSNL
(4)
CLUSTER
CLWLPRTY
CLWLRANK
CLWLWGHT
COMPHDR
COMPMSG
CONNAME
CONVERT
(3)
DEFCDISP
DESCR
DISCINT
(4)
HBINT
KAINT
LOCLADDR
LONGRTY
LONGTMR
MAXINST
MAXINSTC
MAXMSGL
MCANAME
(4)
MCATYPE
MCAUSER
MODENAME
MONCHL
MRDATA
MREXIT
MRRTY
MRTMR
MSGDATA
MSGEXIT
(4)
NETPRTY
(4)
NPMSPEED

Requested attrs continued:



Notes:

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

2 Not valid for z/OS client-connection channels.

3 Valid only on z/OS.

4 Valid only on AIX, HP OpenVMS, HP-UX, i5/OS, Solaris, Windows, and z/OS.

5 Valid only on Compaq NSK.

6 Valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

Parameter descriptions for DISPLAY CHANNEL

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 5). 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.

WHERE
Specify a filter condition to display only those channels that satisfy the

selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Almost any parameter that can be used to display attributes for this DISPLAY command. However, you cannot use the CMDSCOPE, QSGDISP, or MCANAME parameters as filter keywords. You cannot use TYPE (or CHLTYPE) if it is also used to select channels. Channels of a type for which the filter keyword is not a valid attribute are not displayed.

operator
This is used to determine whether a channel satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

CT Contains a specified item. If the *filter-keyword* is a list, you can use this to display objects the attributes of which contain the specified item.

EX Does not contain a specified item. If the *filter-keyword* is a list, you can use this to display objects the attributes of which do not contain the specified item.

CTG Contains an item which matches a generic string that you provide as a *filter-value*. If the *filter-keyword* is a list, you can use this to display objects the attributes of which match the generic string.

EXG Does not contain any item which matches a generic string that you provide as a *filter-value*. If the*filter-keyword* is a list, you can use this to display objects the attributes of which do not match the generic string.

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.

  You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter (for example, the value SDR on the TYPE parameter), you can only use EQ or NE.
- A generic value. This is a character string (such as the character string you supply for the DESCR parameter) with an asterisk at the end, for example ABC*. The characters must be valid for the attribute you are testing. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are

listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.

- An item in a list of values. The value can be explicit or, if it is a character value, it can be explicit or generic. If it is explicit, use CT or EX as the operator. For example, if the value DEF is specified with the operator CT, all items where one of the attribute values is DEF are listed. If it is generic, use CTG or EXG as the operator. If ABC* is specified with the operator CTG, all items where one of the attribute values begins with ABC are listed.

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, HP OpenVMS, HP-UX, i5/OS, Linux, 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 z/OS this is also the default if you specify a filter condition using the WHERE parameter, but on other platforms only requested attributes are displayed.

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 and QSGDISP values are also displayed.

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.

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:

QMGR
The object was defined with QSGDISP(QMGR).

GROUP
The object was defined with QSGDISP(GROUP).

COPY The object was defined with QSGDISP(COPY).

You cannot use QSGDISP as a filter keyword.

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.

CLNTCONN
Client-connection channels only are displayed.

SVRCONN
Server-connection channels only are displayed.

CLUSSDR
Cluster-sender channels only are displayed (valid on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS only).

CLUSRCVR
Cluster-receiver channels only are displayed (valid on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS only).

On AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows, CHLTYPE(*type*) can be used as a synonym for this parameter.

Requested parameters

Specify one or more DISPLAY CHANNEL 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. Table 9 shows the parameters that are relevant for each type of channel. There is a description of each parameter after the table. Parameters are optional unless the description states that they are required.

Table 9. Parameters that can be returned by the DISPLAY CHANNEL command

Parameter	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR
AFFINITY					X			
ALTDATE	X	X	X	X	X	X	X	X
ALTTIME	X	X	X	X	X	X	X	X
AUTOSTART		X	X	X		X		
BATCHHB	X	X					X	X
BATCHINT	X	X					X	X
BATCHSZ	X	X	X	X			X	X
channel-name	X	X	X	X	X	X	X	X
CHLTYPE	X	X	X	X	X	X	X	X
CLNTWGHT					X			
CLUSNL							X	X
CLUSTER							X	X
CLWLPRTY							X	X
CLWLRANK							X	X
CLWLWGHT							X	X
COMPHDR	X	X	X	X	X	X	X	X
COMPMSG	X	X	X	X	X	X	X	X

Table 9. Parameters that can be returned by the DISPLAY CHANNEL command (continued)

Parameter	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR
CONNAME	X	X		X	X		X	X
CONVERT	X	X					X	X
DEFCDISP	X	X	X	X		X		
DESCR	X	X	X	X	X	X	X	X
DISCINT	X	X				X[1]	X	X
HBINT	X	X	X	X	X	X	X	X
KAINT	X	X	X	X	X	X	X	X
LOCLADDR	X	X		X	X		X	X
LONGRTY	X	X					X	X
LONGTMR	X	X					X	X
MAXINST						X		
MAXINSTC						X		
MAXMSGL	X	X	X	X	X	X	X	X
MCANAME	X	X		X			X	X
MCATYPE	X	X		X			X	X
MCAUSER	X	X	X	X		X	X	X
MODENAME	X	X		X	X		X	X
MONCHL	X	X	X	X		X	X	X
MRDATA			X	X				X
MREXIT			X	X				X
MRRTY			X	X				X
MRTMR			X	X				X
MSGDATA	X	X	X	X			X	X
MSGEXIT	X	X	X	X			X	X
NETPRTY								X
NPMSPEED	X	X	X	X			X	X
PASSWORD	X	X		X	X		X	
PROPCTL	X	X					X	
PUTAUT			X	X		X[2]		X
QMNAME					X			
RCVDATA	X	X	X	X	X	X	X	X
RCVEXIT	X	X	X	X	X	X	X	X
SCYDATA	X	X	X	X	X	X	X	X
SCYEXIT	X	X	X	X	X	X	X	X
SENDDATA	X	X	X	X	X	X	X	X
SENDEXIT	X	X	X	X	X	X	X	X
SEQWRAP	X	X	X	X			X	X
SHARECNV						X		
SHORTRTY	X	X					X	X

Table 9. Parameters that can be returned by the DISPLAY CHANNEL command (continued)

Parameter	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR
SHORTTMR	X	X					X	X
SSLCAUTH		X	X	X		X		X
SSLCIPH	X	X	X	X	X	X	X	X
SSLPEER	X	X	X	X	X	X	X	X
STATCHL	X	X	X	X			X	X
TPNAME	X	X		X	X	X	X	X
TRPTYPE	X	X	X	X	X	X	X	X
USERID	X	X		X	X		X	
XMITQ	X	X						
Note: 1. DISCINT is valid for a channel type of SVRCONN on z/OS only. 2. PUTAUT is valid for a channel type of SVRCONN on z/OS only.								

AFFINITY
The channel affinity attribute .

PREFERRED
Subsequent connections in a process attempt to use the same channel definition as the first connection.

NONE
All connections in a process select an applicable definition based on the weighting with any applicable CLNTWGHT(0) definitions selected first in alphabetical order.

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, HP OpenVMS, HP-UX, Linux, i5/OS, 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.

On AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows, TYPE can be used as a synonym for this parameter.

CLNTWGHT
The client channel weighting.

The special value 0 indicates that no random load balancing is performed and applicable definitions are selected in alphabetical order. If random load balancing is performed the value is in the range 1 - 99 where 1 is the lowest weighting and 99 is the highest.

CLUSTER
The name of the cluster to which the channel belongs.

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

CLWLPRTY
The priority of the channel for the purposes of cluster workload distribution.

CLWLRANK
The rank of the channel for the purposes of cluster workload distribution.

CLWLWGHT
The weighting of the channel for the purposes of cluster workload distribution.

COMPHDR
The list of header data compression techniques supported by the channel. For sender, server, cluster-sender, cluster-receiver and client-connection channels the values specified are in order of preference.

COMPMSG
The list of message data compression techniques supported by the channel. For sender, server, cluster-sender, cluster-receiver and client-connection channels the values specified are in order of preference.

CONNAME
Connection name.

CONVERT
Whether sender should convert application message data.

DEFCDISP
Specifies the default channel disposition of the channels for which information is to be returned. If this keyword is not present, channels of all default channel dispositions are eligible.

ALL Channels of all default channel dispositions are displayed.

This is the default setting.

PRIVATE
Only channels where the default channel disposition is PRIVATE are displayed.

SHARED
Only channels where the default channel disposition is FIXSHARED or SHARED are displayed.

Note: This does not apply to client-connection channel types on z/OS.

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.

LONGTMR
Long retry timer.

MAXINST(*integer***)**
The maximum number of instances of a server-connection channel that are permitted to run simultaneously.

MAXINSTC(*integer***)**
The maximum number of instances of a server-connection channel, started from a single client, that are permitted to run simultaneously.

Note: In this context, connections originating from the same remote network address are regarded as coming from the same client.

MAXMSGL
Maximum message length for channel.

MCANAME
Message channel agent name.

You cannot use MCANAME as a filter keyword.

MCATYPE
Whether message channel agent runs as a separate process or a separate thread.

MCAUSER
Message channel agent user identifier.

MODENAME
LU 6.2 mode name.

MONCHL
Online monitoring data collection.

MRDATA
Channel message-retry exit user data.

MREXIT
Channel message-retry exit name.

MRRTY
Channel message-retry count.

MRTMR
Channel message-retry time.

MSGDATA
Channel message exit user data.

MSGEXIT
Channel message exit names.

NETPRTY
The priority for the network connection.

NPMSPEED
Nonpersistent message speed.

PASSWORD
Password for initiating LU 6.2 session (if nonblank, this is displayed as asterisks on all platforms except z/OS).

PROPCTL
Message property control.

PUTAUT
Put authority.

QMNAME
Queue manager name.

RCVDATA
Channel receive exit user data.

RCVEXIT
Channel receive exit names.

SCYDATA
Channel security exit user data.

SCYEXIT
Channel security exit names.

SENDDATA
Channel send exit user data.

SENDEXIT
Channel send exit names.

SEQWRAP
Sequence number wrap value.

SEQWRAP
Sequence number wrap value.

SHARECNV
Sharing conversations value.

SHORTTMR
Short retry timer.

SSLCAUTH
Whether SSL client authentication is required.

SSLCIPH
Cipher specification for the SSL connection.

SSLPEER
Filter for the Distinguished Name from the certificate of the peer queue manager or client at the other end of the channel.

STATCHL
Statistics data collection.

TPNAME
LU 6.2 transaction program name.

TRPTYPE
Transport type.

USERID
User identifier for initiating LU 6.2 session.

XMITQ
Transmission queue name.

For more details of these parameters, see "DEFINE CHANNEL" on page 144.

DISPLAY CHINIT

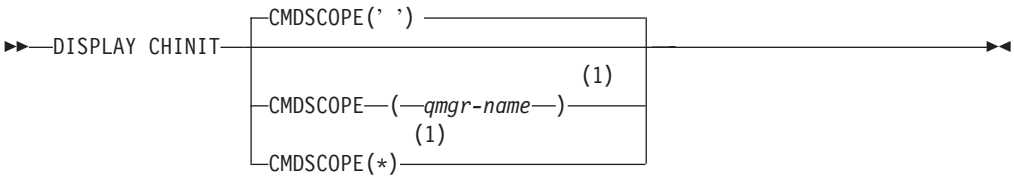
Use the MQSC command DISPLAY CHINIT to display information about the channel initiator. The command server must be running.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS CHI or DIS DQM

DISPLAY CHINIT



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for DISPLAY CHINIT

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 for DISPLAY CHINIT

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
   - 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 CHSTATUS

Use the MQSC command DISPLAY CHSTATUS to display the status of one or more channels.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	CR

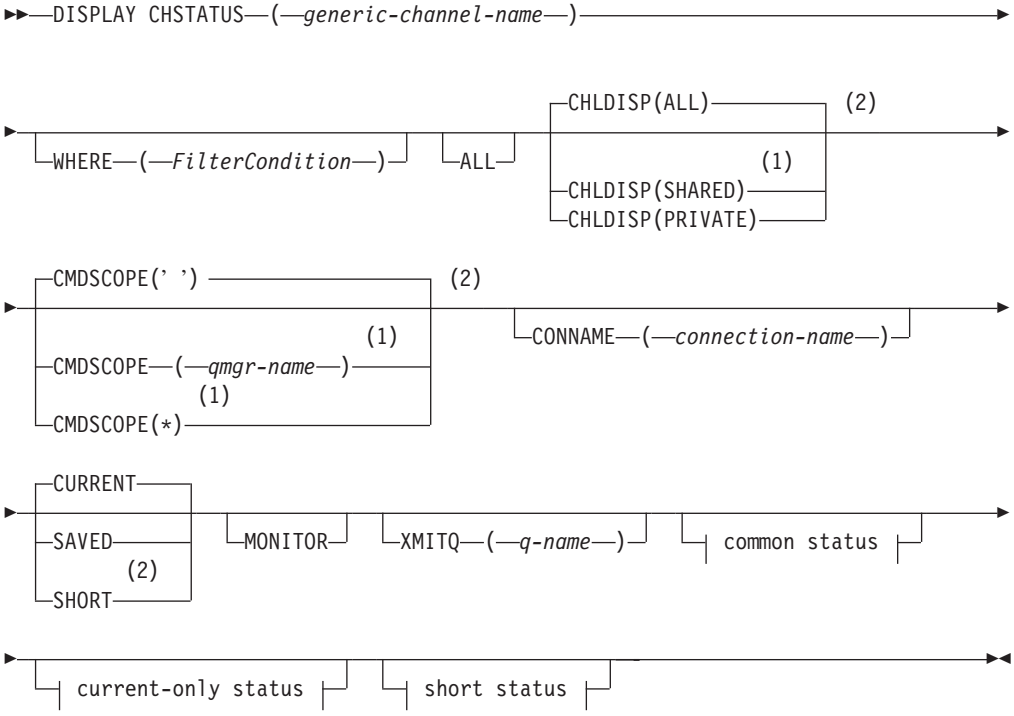
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note: On z/OS:

1. The command fails if the channel initiator has not been started.
2. The command server must be running.

Synonym: DIS CHS

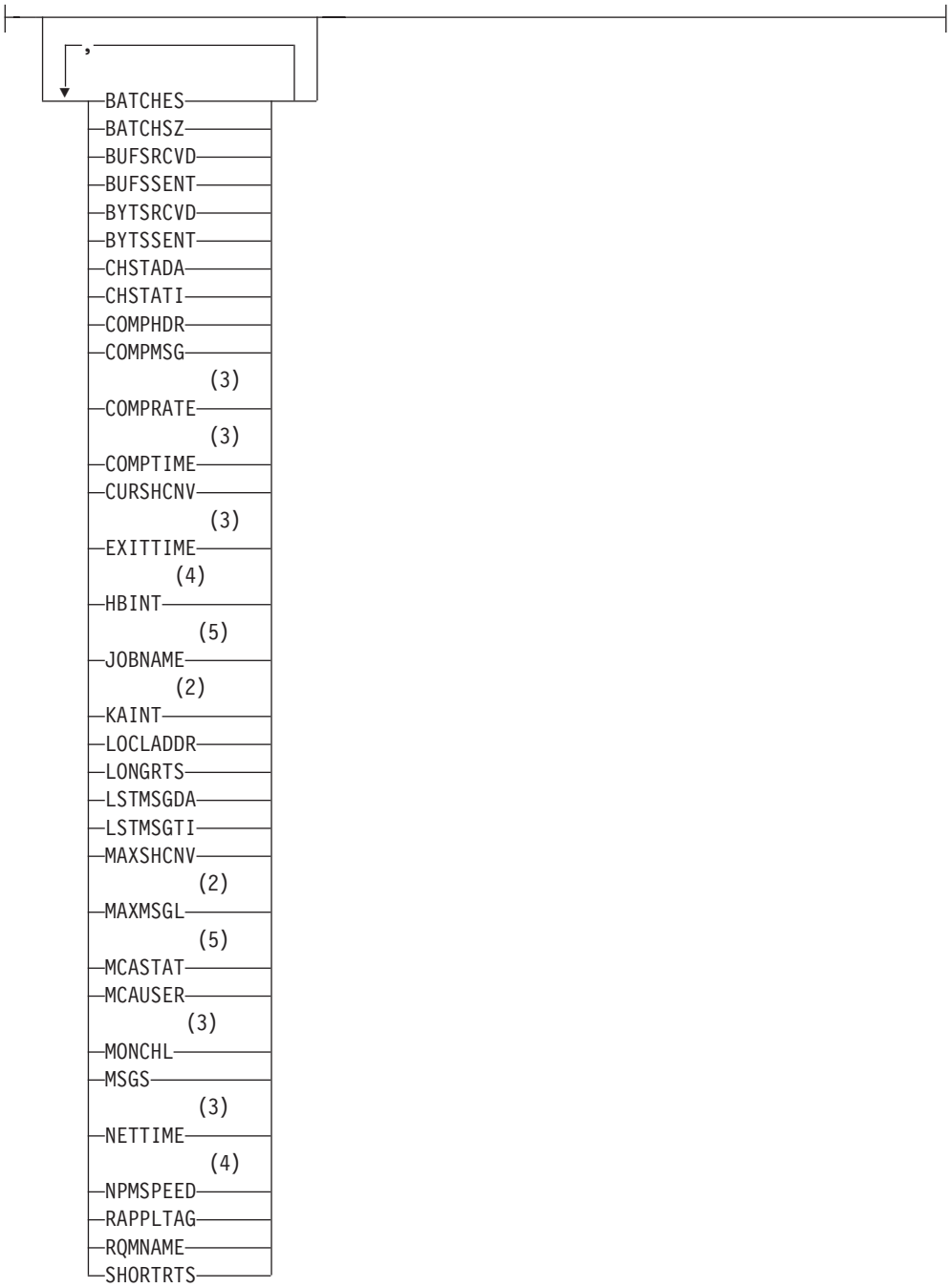
DISPLAY CHSTATUS



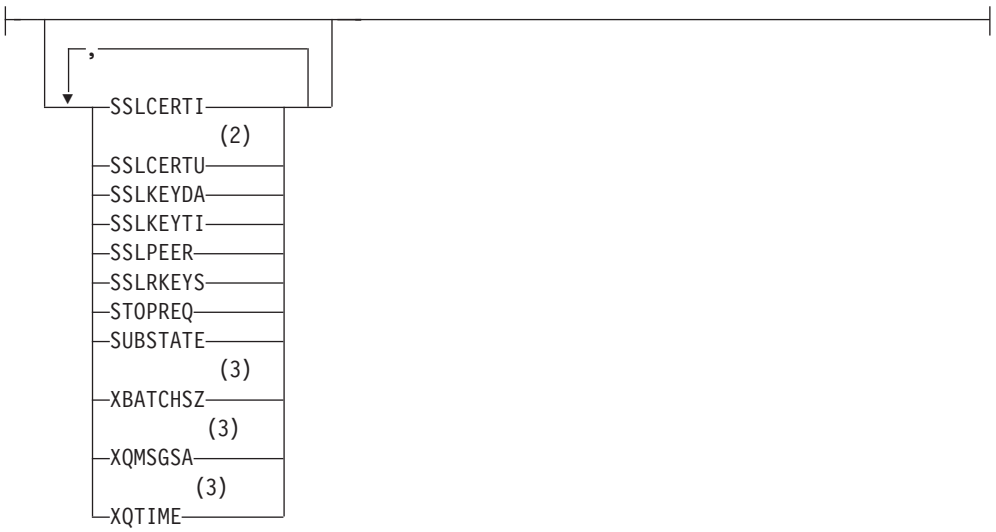
Common status:

,
CHLTYPE
CURLUWID
CURMSGS
CURSEQNO
INDOUBT
LSTLUWID
LSTSEQNO
STATUS

Current-only status:



Current-only status continued:



Short status:

Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

3 Also displayed by selection of the MONITOR parameter.

4 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

5 Ignored if specified on z/OS.

Parameter descriptions for DISPLAY CHSTATUS

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.

Status for all channels that meet the selection criteria is given, whether the channels were defined manually or automatically.

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, HP OpenVMS, HP-UX, Linux, i5/OS, 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 NPMSPEED 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.

Note: On Compaq NonStop Kernel, channel status is updated only at the boundaries of batch processing. Channel status information is not updated for every message transfer because of the potential impact on the performance of channels. This means that the common status data values are identical for both the current and saved sets.

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.

WHERE
Specify a filter condition to display status information for those channels that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
The parameter to be used to display attributes for this DISPLAY command. However, you cannot use the following parameters as filter keywords: CHLDISP, CMDSCOPE, COMPRATE, COMPTIME,

CURRENT, EXITTIME, JOBNAME (on z/OS), MCASTAT (on z/OS), MONITOR, NETTIME, SAVED, SHORT, XBATCHSZ, or XQTIME as filter keywords.

You cannot use CONNAME or XMITQ as filter keywords if you also use them to select channel status.

Status information for channels of a type for which the filter keyword is not valid is not displayed.

operator
This is used to determine whether a channel satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

CT Contains a specified item. If the *filter-keyword* is a list, you can use this to display objects the attributes of which contain the specified item.

EX Does not contain a specified item. If the *filter-keyword* is a list, you can use this to display objects the attributes of which do not contain the specified item.

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.

  You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter (for example, the value SDR on the CHLTYPE parameter), you can only use EQ or NE.
- A generic value. This is a character string with an asterisk at the end, for example ABC*. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

  You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.
- An item in a list of values. Use CT or EX as the operator. For example, if the value DEF is specified with the operator CT, all items where one of the attribute values is DEF are listed.

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.

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.

FIXSHARED

The status is for a shared channel, tied to a specific 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.

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.

You cannot use CMDSCOPE as a filter keyword.

Note: See Table 10 on page 362, Table 11 on page 362, and Table 12 on page 362 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, when using TCP, if CONNAME in the channel definition :

- Is blank or is in "host name" format, the channel status value has the resolved IP address.
- Includes the port number, the current channel status value includes the port number (except on z/OS), 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.

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.

MONITOR
Specify this to return the set of online monitoring parameters. These are COMPRATE, COMPTIME, EXITTIME, MONCHL, NETTIME, XBATCHSZ, XQMSGSA, and XQTIME. If you specify this parameter, any of the monitoring parameters that you request specifically have no effect; all monitoring parameters are still displayed.

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.

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, and for all channel types except server-connection channels)
- The remote partner application name (for server-connection channels)
- The type of status information returned (CURRENT, SAVED, or on z/OS only, SHORT)
- STATUS (except SAVED on z/OS)
- On z/OS, CHLDISP
- STOPREQ (only for current status)
- SUBSTATE

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.

CHLTYPE
The channel type. This is one of the following:

SDR A sender channel

SVR A server channel

RCVR A receiver channel

RQSTR
A requester channel

CLUSSDR
A cluster-sender channel

CLUSRCVR
A cluster-receiver channel

SVRCONN
A server-connection channel

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 NPMSPEED 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, HP OpenVMS, HP-UX, Linux, i5/OS, 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 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 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

Parameter descriptions for the current channel instances of the DISPLAY CHSTATUS command.

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, HP OpenVMS, HP-UX, Linux, i5/OS, 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).

COMPHDR
The technique used to compress the header data sent by the channel. Two values are displayed:

- The default header data compression value negotiated for this channel.
- The header data compression value used for the last message sent. The header data compression value can be altered in a sending channels message exit. If no message has been sent, the second value is blank.

COMPMSG
The technique used to compress the message data sent by the channel. Two values are displayed:

- The default message data compression value negotiated for this channel.
- The message data compression value used for the last message sent. The message data compression value can be altered in a sending channels message exit. If no message has been sent, the second value is blank.

COMPRATE
The compression rate achieved displayed to the nearest percentage. Two values are displayed:

- A value based on recent activity over a short period of time.
- A value based on activity over a longer period of time.

These values are reset every time the channel is started and are displayed only when the STATUS of the channel is RUNNING. If monitoring data is not being collected, or if no messages have been sent by the channel, the values are shown as blank.

A value is only displayed for this parameter if MONCHL is set for this channel.

COMPTIME
The amount of time per message, displayed in microseconds, spent during compression or decompression. Two values are displayed:

- A value based on recent activity over a short period of time.
- A value based on activity over a longer period of time.

These values are reset every time the channel is started and are displayed only when the STATUS of the channel is RUNNING. If monitoring data is not being collected, or if no messages have been sent by the channel, the values are shown as blank.

A value is only displayed for this parameter if MONCHL is set for this channel.

CURSHCNV
The CURSHCNV value is blank for all channel types other than server-connection channels. For each instance of a server-connection channel, the CURSHCNV output gives a count of the number of conversations currently running over that channel instance.

A value of zero indicates that the channel is running in a mode prior to that of WebSphere MQ Version 7.0, with regard to:

- Administrator stop-quiesce
- Heartbeating
- Read ahead
- Sharing conversations
- Client Asynchronous Consume

EXITTIME
Amount of time, displayed in microseconds, spent processing user exits per message. Two values are displayed:

- A value based on recent activity over a short period of time.
- A value based on activity over a longer period of time.

These values depend on the configuration and behavior of your system, as well as the levels of activity within it, and serve as an indicator that your system is performing normally. A significant variation in these values may indicate a problem with your system. They are reset every time the channel is started and are displayed only when the STATUS of the channel is RUNNING.

This parameter is also displayed when you specify the MONITOR parameter.

A value is only displayed for this parameter if MONCHL is set for this channel.

HBINT
The heartbeat interval being used for this session.

JOBNAME
Name of job currently serving the channel.

- On HP OpenVMS, this is the process identifier, displayed in hexadecimal.
- On i5/OS, 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.

You cannot use JOBNAME as a filter keyword on z/OS.

KAINT
The keepalive interval being used for this session. This is valid only on z/OS.

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.

In the case of a server-connection channel instance on which conversations are being shared, this is the time when the last MQI call completed on any of the conversations running on the channel instance.

MAXMSGL
The maximum message length being used for this session (valid only on z/OS).

MAXSHCNV
The MAXSHCNV value is blank for all channel types other than server-connection channels. For each instance of a server-connection channel, the MAXSHCNV output gives the negotiated maximum of the number of conversations that can run over that channel instance.

A value of zero indicates that the channel instance is running in a mode prior to that of WebSphere MQ Version 7.0, with regard to:

- Administrator stop-quiesce
- Heartbeating
- Read ahead
- Sharing conversations
- Client asynchronous consume

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.

You cannot use MCASTAT as a filter keyword on z/OS.

MCAUSER
The user ID used by the MCA. This can be the user ID set in the channel definition, the default user ID for MCA channels, a user ID transferred from a client if this is a server-connection channel, or a user ID specified by a security exit.

This parameter applies only to server-connection, receiver, requester, and cluster-receiver channels.

On server connection channels that share conversations, the MCAUSER field contains a user ID if all the conversations have the same MCA user ID value. If the MCA user ID in use varies across these conversations, the MCAUSER field contains a value of *.

The maximum length is 12 characters on z/OS; on other platforms, it is 64 characters.

MONCHL
Current level of monitoring data collection for the channel.

This parameter is also displayed when you specify the MONITOR parameter.

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).

In the case of a server-connection channel instance on which conversations are being shared, this is the total number of MQI calls handled on all of the conversations running on the channel instance.

NETTIME
Amount of time, displayed in microseconds, to send a request to the remote end of the channel and receive a response. Two values are displayed:

- A value based on recent activity over a short period of time.
- A value based on activity over a longer period of time.

These values depend on the configuration and behavior of your system, as well as the levels of activity within it, and serve as an indicator that your system is performing normally. A significant variation in these values may indicate a problem with your system. They are reset every time the channel is started and are displayed only when the STATUS of the channel is RUNNING.

This parameter applies only to sender, server, and cluster-sender channels.

This parameter is also displayed when you specify the MONITOR parameter.

A value is only displayed for this parameter if MONCHL is set for this channel.

NPMSPEED
The nonpersistent message handling technique being used for this session.

RAPPLTAG
The remote partner application name. This is the name of the client application at the remote end of the channel. This parameter applies only to server-connection channels.

RQMNAME
The queue manager name, or queue-sharing group name, of the remote system. This parameter does not apply to server-connection channels.

SHORTRTS
Number of short retry wait start attempts left. This applies only to sender or server channels.

SSLCERTI
The full Distinguished Name of the issuer of the remote certificate. The issuer is the Certificate Authority that issued the certificate.

The maximum length is 256 characters. This limit might mean that exceptionally long Distinguished Names are truncated.

SSLCERTU
The local user ID associated with the remote certificate. This is valid on z/OS only.

SSLKEYDA
Date on which the previous successful SSL secret key reset was issued.

SSLKEYTI
Time at which the previous successful SSL secret key reset was issued.

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.

SSLRKEYS
Number of successful SSL key resets. The count of SSL secret key resets is reset when the channel instance ends.

STOPREQ
Whether a user stop request is outstanding. This is either YES or NO.

SUBSTATE
Action being performed by the channel when this command is issued. The following substates are listed in precedence order, starting with the substate of the highest precedence:

ENDBATCH
Channel is performing end-of-batch processing.

SEND A request has been made to the underlying communication subsystem to send some data.

RECEIVE
A request has been made to the underlying communication subsystem to receive some data.

SERIALIZE
Channel is serializing its access to the queue manager. This is valid on z/OS only.

RESYNCH
Channel is resynchronizing with the partner.

HEARTBEAT
Channel is heartbeating with the partner.

SCYEXIT
Channel is running the security exit.

RCVEXIT
Channel is running one of the receive exits.

SENDEXIT
Channel is running one of the send exits.

MSGEXIT
Channel is running one of the message exits.

MREXIT
Channel is running the message retry exit.

CHADEXIT
Channel is running through the channel auto-definition exit.

NETCONNECT
A request has been made to the underlying communication subsystem to connect a partner machine.

SSLHANDSHK
Channel is processing an SSL handshake.

NAMESERVER
A request has been made to the name server.

MQPUT
A request has been made to the queue manager to put a message on the destination queue.

MQGET
A request has been made to the queue manager to get a message from the transmission queue (if this is an MCA channel) or from an application queue (if this is an MQI channel).

MQICALL
A MQ API call, other than MQPUT and MQGET, is being executed.

COMPRESS
Channel is compressing or decompressing data.

Not all substates are valid for all channel types or channel states. There are occasions when no substate is valid, at which times a blank value is returned.

For channels running on multiple threads, this parameter displays the substate of the highest precedence.

XBATCHSZ
Size of the batches transmitted over the channel. Two values are displayed:

- A value based on recent activity over a short period of time.
- A value based on activity over a longer period of time.

These values depend on the configuration and behavior of your system, as well as the levels of activity within it, and serve as an indicator that your system is performing normally. A significant variation in these values may

indicate a problem with your system. They are reset every time the channel is started and are displayed only when the STATUS of the channel is RUNNING.

This parameter does not apply to server-connection channels.

This parameter is also displayed when you specify the MONITOR parameter.

A value is only displayed for this parameter if MONCHL is set for this channel.

XQMSGSA
Number of messages queued on the transmission queue available to the channel for MQGETs.

This parameter has a maximum displayable value of 999. If the number of messages available exceeds 999, a value of 999 is displayed.

On z/OS, if the transmission queue is not indexed by *CorrelId*, this value is shown as blank.

This parameter applies to cluster-sender channels only.

This parameter is also displayed when you specify the MONITOR parameter.

A value is only displayed for this parameter if MONCHL is set for this channel.

XQTIME
The time, in microseconds, that messages remained on the transmission queue before being retrieved. The time is measured from when the message is put onto the transmission queue until it is retrieved to be sent on the channel and, therefore, includes any interval caused by a delay in the putting application.

Two values are displayed:

- A value based on recent activity over a short period of time.
- A value based on activity over a longer period of time.

These values depend on the configuration and behavior of your system, as well as the levels of activity within it, and serve as an indicator that your system is performing normally. A significant variation in these values may indicate a problem with your system. They are reset every time the channel is started and are displayed only when the STATUS of the channel is RUNNING.

This parameter applies only to sender, server, and cluster-sender channels.

This parameter is also displayed when you specify the MONITOR parameter.

A value is only displayed for this parameter if MONCHL is set for this channel.

Short status

The following information applies only to current channel instances.

QMNAME
The name of the queue manager that owns the channel instance.

Usage notes for DISPLAY CHSTATUS

- The status information that is returned for various combinations of CHLDISP, CMDSCOPE, and status type are summarized in Table 10, Table 11, and Table 12.

Table 10. CHLDISP and CMDSCOPE for DISPLAY CHSTATUS CURRENT

CHLDISP	CMDSCOPE( ) or CMDSCOPE (local-qmgr)	CMDSCOPE (qmgr-name)	CMDSCOPE(*)
PRIVATE	Common and current-only status for current private channels on the local queue manager	Common and current-only status for current private channels on the named queue manager	Common and current-only status for current private channels on all queue managers
SHARED	Common and current-only status for current shared channels on the local queue manager	Common and current-only status for current shared channels on the named queue manager	Common and current-only status for current shared channels on all queue managers
ALL	Common and current-only status for current private and shared channels on the local queue manager	Common and current-only status for current private and shared channels on the named queue manager	Common and current-only status for current private and shared channels on all active queue managers

Table 11. CHLDISP and CMDSCOPE for DISPLAY CHSTATUS SHORT

CHLDISP	CMDSCOPE( ) or CMDSCOPE (local-qmgr)	CMDSCOPE (qmgr-name)	CMDSCOPE(*)
PRIVATE	STATUS and short status for current private channels on the local queue manager	STATUS and short status for current private channels on the named queue manager	STATUS and short status for current private channels on all active queue managers
SHARED	STATUS and short status for current shared channels on all active queue managers in the queue-sharing group	Not permitted	Not permitted
ALL	STATUS and short status for current private channels on the local queue manager and current shared channels in the queue-sharing group(1)	STATUS and short status for current private channels on the named queue manager	STATUS and short status for current private, and shared, channels on all active queue managers in the queue-sharing group(1)

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 12. CHLDISP and CMDSCOPE for DISPLAY CHSTATUS SAVED

CHLDISP	CMDSCOPE( ) or CMDSCOPE (local-qmgr)	CMDSCOPE (qmgr-name)	CMDSCOPE(*)
PRIVATE	Common status for saved private channels on the local queue manager	Common status for saved private channels on the named queue manager	Common status for saved private channels on all active queue managers
SHARED	Common status for saved shared channels on all active queue managers in the queue-sharing group	Not permitted	Not permitted

Table 12. CHLDISP and CMDSCOPE for DISPLAY CHSTATUS SAVED (continued)

CHLDISP	CMDSCOPE( ) or CMDSCOPE (local-qmgr)	CMDSCOPE (qmgr-name)	CMDSCOPE(*)
ALL	Common status for saved private channels on the local queue manager and saved shared channels in the queue-sharing group	Common status for saved private channels on the named queue manager	Common status for saved private, and shared, channels on all active queue managers in the queue-sharing group

- On z/OS, if any numeric parameter exceeds 999 999 999, it is displayed as 999 999 999.

DISPLAY CLUSQMGR

Use the MQSC command DISPLAY CLUSQMGR to display information about cluster channels for queue managers in a cluster.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

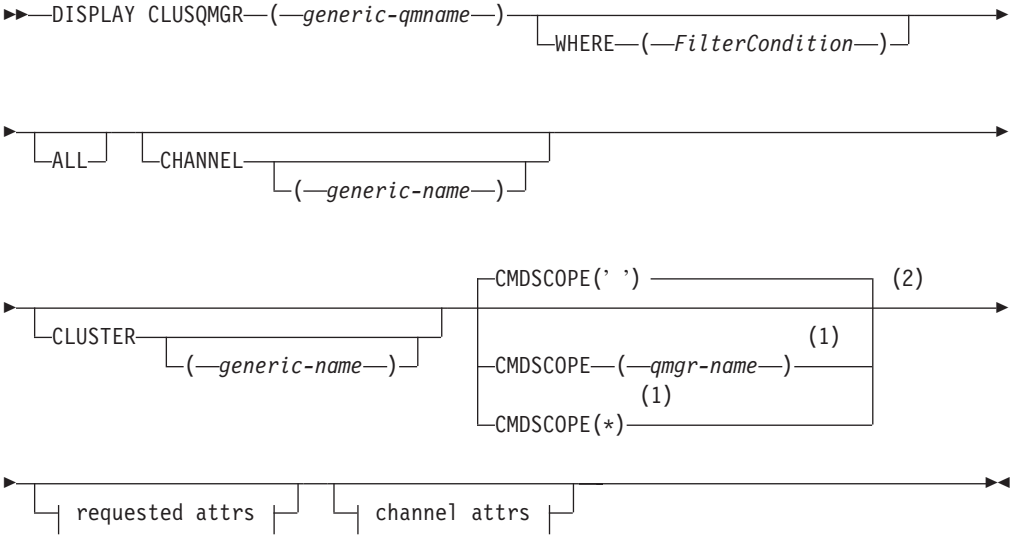
Unlike the DISPLAY CHANNEL command, this command includes information about cluster channels that are auto-defined, and the status of cluster channels.

Note:

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.

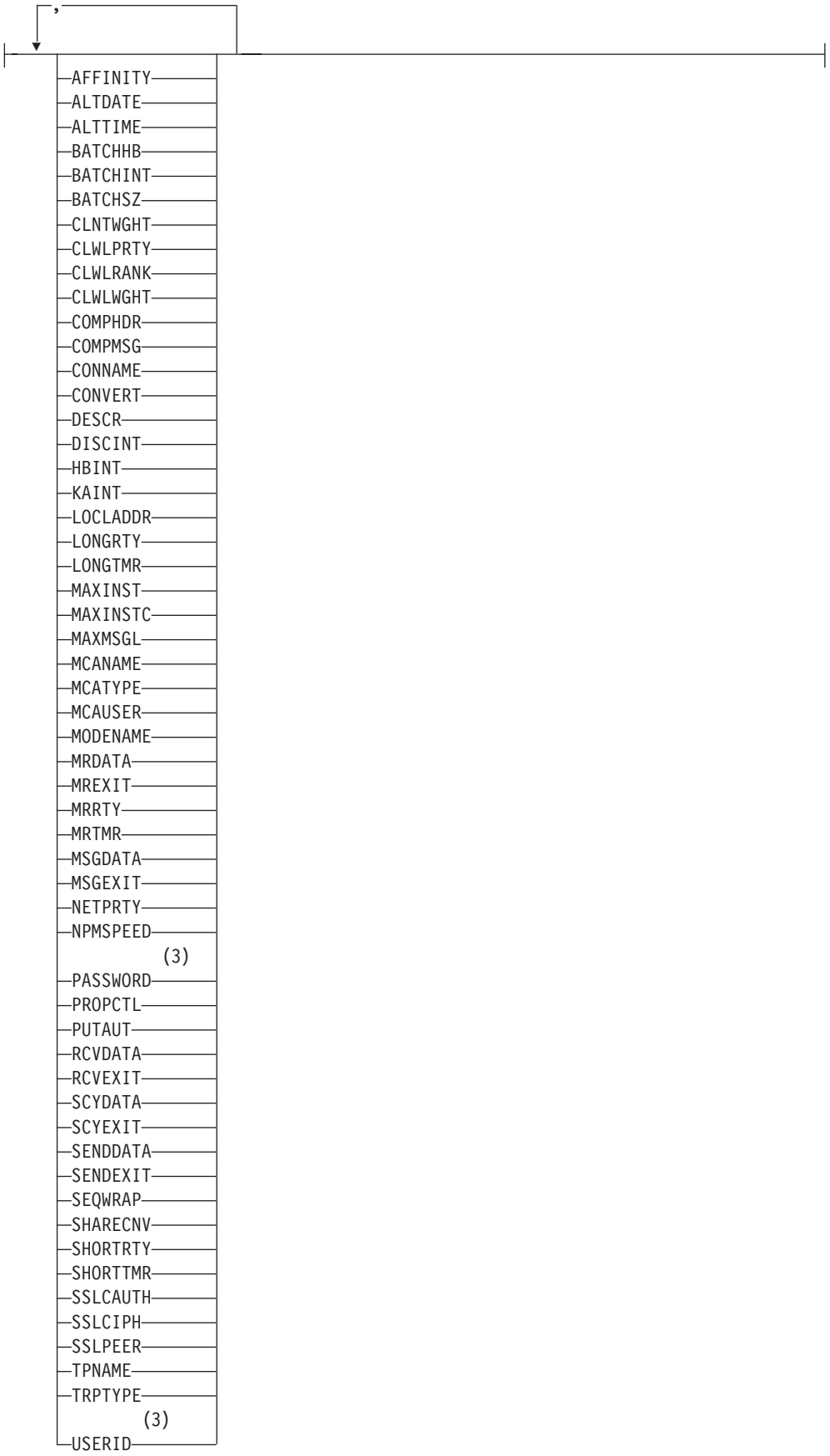
Synonym: DIS CLUSQMGR

DISPLAY CLUSQMGR



Requested attrs:

Channel attrs:


,
AFFINITY
ALTDATE
ALTTIME
BATCHHB
BATCHINT
BATCHSZ
CLNTWGHT
CLWLPRTY
CLWLRANK
CLWLWGHT
COMPHDR
COMPMSG
CONNAME
CONVERT
DESCR
DISCINT
HBINT
KAINT
LOCLADDR
LONGRTY
LONGTMR
MAXINST
MAXINSTC
MAXMSGL
MCANAME
MCATYPE
MCAUSER
MODENAME
MRDATA
MREXIT
MRRTY
MRTMR
MSGDATA
MSGEXIT
NETPRTY
NPMSPEED
(3)
PASSWORD
PROPCTL
PUTAUT
RCVDATA
RCVEXIT
SCYDATA
SCYEXIT
SENDDATA
SENDEXIT
SEQWRAP
SHARECNV
SHORTRTY
SHORTTMR
SSLCAUTH
SSLCIPH
SSLPEER
TPNAME
TRPTYPE
(3)
USERID

Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

3 Not valid on z/OS.

Parameter descriptions for DISPLAY CLUSQMGR

(*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.

WHERE
Specify a filter condition to display only those cluster channels that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Almost any parameter that can be used to display attributes for this DISPLAY command. However, you cannot use the CMDSCOPE or MCANAME parameters as filter keywords. You cannot use CHANNEL or CLUSTER as filter keywords if you use them to select cluster queue managers.

operator
This is used to determine whether a cluster channel satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

CT Contains a specified item. If the *filter-keyword* is a list, you can use this to display objects the attributes of which contain the specified item.

EX Does not contain a specified item. If the *filter-keyword* is a list, you can use this to display objects the attributes of which do not contain the specified item.

CTG Contains an item which matches a generic string that you provide as a *filter-value*. If the *filter-keyword* is a list, you can use this to display objects the attributes of which match the generic string.

EXG Does not contain any item which matches a generic string that

you provide as a *filter-value*. If the *filter-keyword* is a list, you can use this to display objects the attributes of which do not match the generic string.

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.

  You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter (for example, the value STARTING on the STATUS parameter), you can only use EQ or NE.
- A generic value. This is a character string (such as the character string you supply for the CONNAME parameter) with an asterisk at the end, for example ABC*. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

  You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.
- An item in a list of values. The value can be explicit or, if it is a character value, it can be explicit or generic. If it is explicit, use CT or EX as the operator. For example, if the value DEF is specified with the operator CT, all items where one of the attribute values is DEF are listed. If it is generic, use CTG or EXG as the operator. If ABC* is specified with the operator CTG, all items where one of the attribute values begins with ABC are listed.

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.

On z/OS this is also the default if you specify a filter condition using the WHERE parameter, but on other platforms only requested attributes are displayed.

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.

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.

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.

You cannot use CMDSCOPE as a filter keyword.

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.

CLUSDATE
The date on which the definition became available to the local queue manager, in the form `yyyy-mm-dd`.

CLUSTIME
The time at which the definition became available to the local queue manager, in the form `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.

QMID
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 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
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

AFFINITY
The channel affinity attribute.

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

CLNTWGHT
The client channel weighting.

CLWLPRTY
The priority of the channel for the purposes of cluster workload distribution.

CLWLRANK
The rank of the channel for the purposes of cluster workload distribution.

CLWLWGHT
The weighting of the channel for the purposes of cluster workload distribution.

COMPHDR
The list of header data compression techniques supported by the channel.

COMPMSG
The list of message data compression techniques supported by the channel.

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

LONGTMR
Long retry timer

MAXINST
The maximum number of instances of a server-connection channel that are permitted to run simultaneously.

MAXINSTC
The maximum number of instances of a server-connection channel, started from a single client, that are permitted to run simultaneously.

MAXMSGL
Maximum message length for channel

MCANAME
Message channel agent name.

You cannot use MCANAME as a filter keyword.

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 count

MRTMR
Channel message-retry time

MSGDATA
Channel message exit user data

MSGEXIT
Channel message exit names

NETPRTY
The priority for the network connection

NPMSPEED
Nonpersistent message speed

PASSWORD
Password for initiating LU 6.2 session (if nonblank, this is displayed as asterisks)

PROPCTL
Message property control.

PUTAUT
Put authority

RCVDATA
Channel receive exit user data

RCVEXIT
Channel receive exit names

SCYDATA
Channel security exit user data

SCYEXIT
Channel security exit name

SENDDATA
Channel send exit user data

SENDEXIT
Channel send exit names

SEQWRAP
Sequence number wrap value

SHARECNV
Sharing conversations value.

SHORTRTY
Short retry count

SHORTTMR
Short retry timer

SSLCAUTH
Whether SSL client authentication is required.

SSLCIPH
Cipher specification for the SSL connection.

SSLPEER
Filter for the Distinguished Name from the certificate of the peer queue manager or client at the other end of the channel.

TRPTYPE
Transport type

TPNAME
LU 6.2 transaction program name

USERID
User identifier for initiating LU 6.2 session

For more information on channel parameters, see “DEFINE CHANNEL” on page 144.

DISPLAY CMDSERV

Use the MQSC command DISPLAY CMDSERV to display the status of the command server.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS CS

DISPLAY CMDSERV

Usage notes for DISPLAY CMDSERV

1. The command server takes messages from the system command input queue, and commands using CMDSCOPE, and processes them. DISPLAY CMDSERV 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 CMDSERV in progress

   STOPPING
   STOP CMDSERV in progress

   STOPPED
   STOP CMDSERV completed

   RUNNING
   Available to process commands, currently processing a message

   WAITING
   Available to process commands, currently waiting for a message

DISPLAY CONN

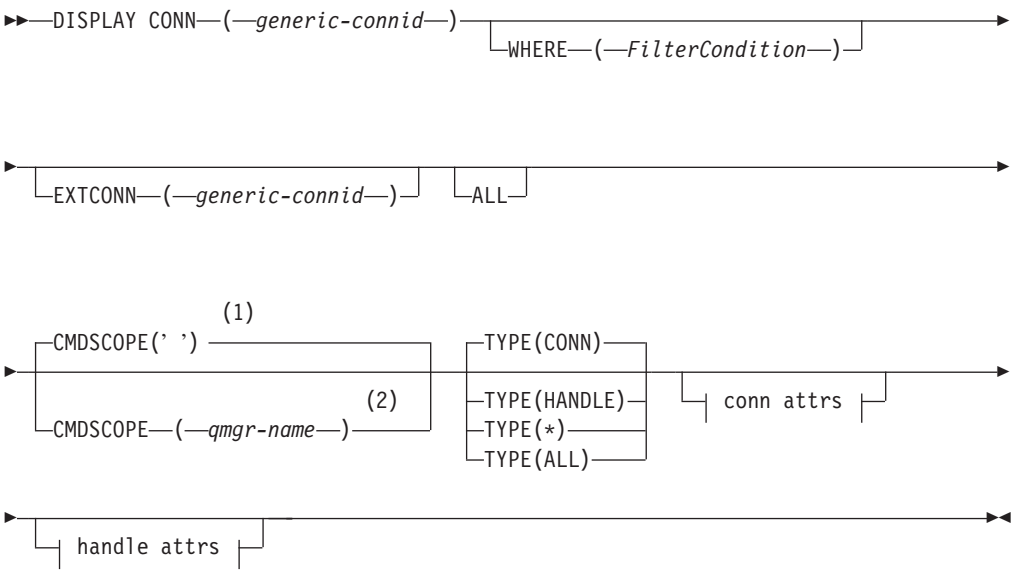
Use the MQSC command DISPLAY CONN to display connection information about the applications connected to the queue manager. This is a useful command because it enables you to identify applications with long-running units of work.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

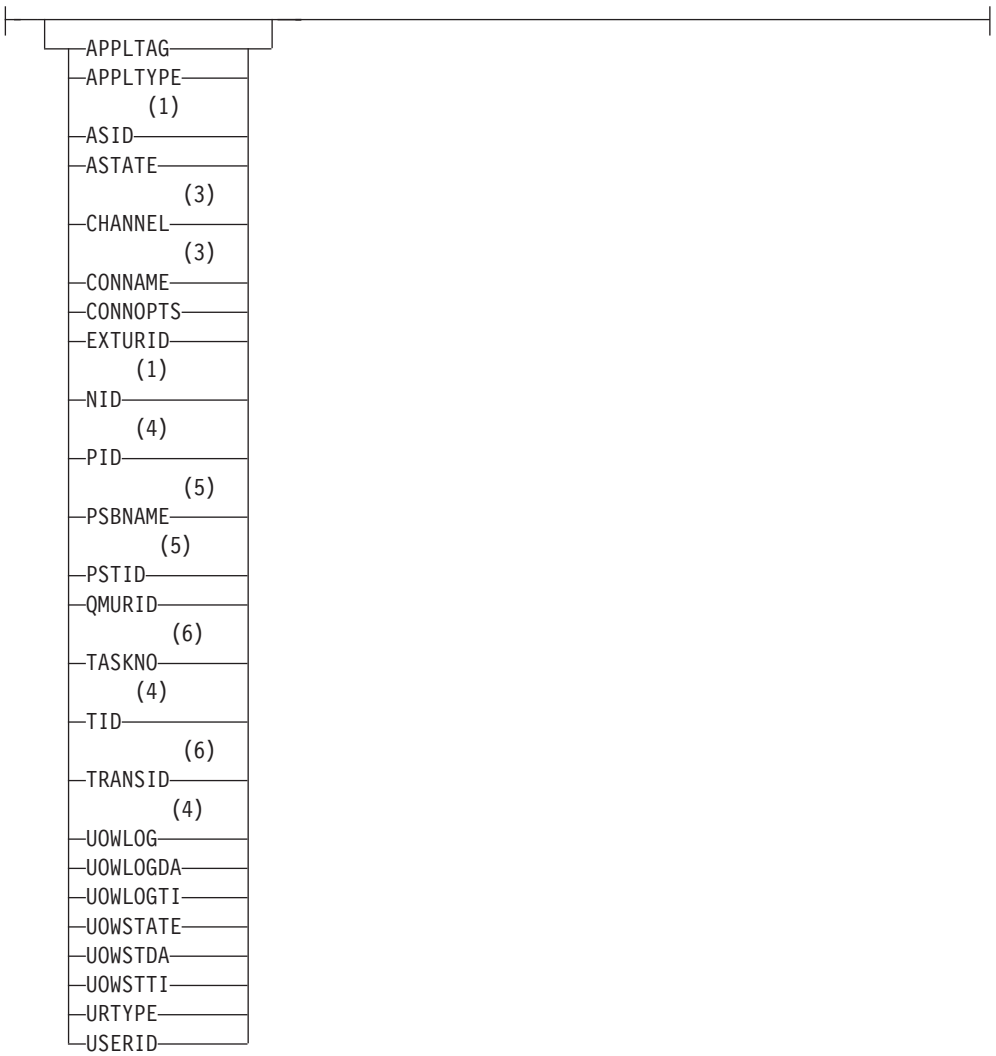
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS CONN

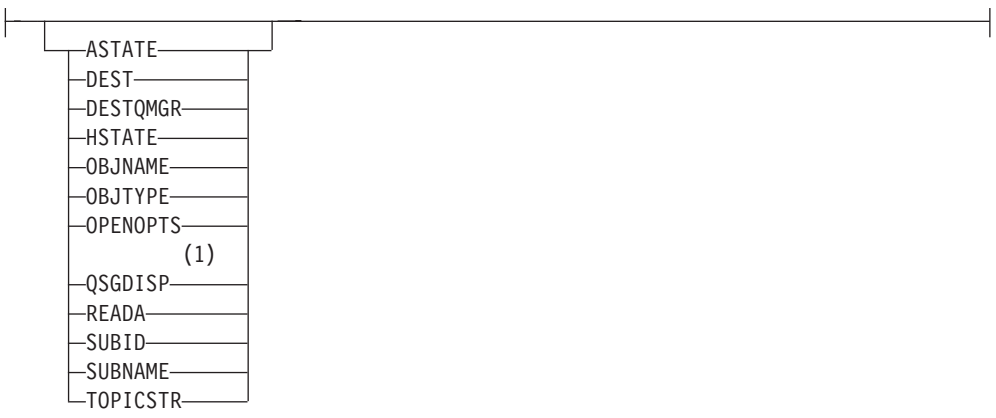
DISPLAY CONN



Conn attrs:



Handle attrs:



Notes:

1 Valid only on z/OS.

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

3 Valid only when the connection is associated with a channel.

4 Not valid on z/OS.

5 IMS only.

6 CICS for z/OS only.

Parameter descriptions for DISPLAY CONN

You must specify a connection for which you want to display information. This can be a specific connection identifier or a generic connection identifier. A single asterisk (*) can be used as a generic connection identifier to display information for all connections.

(*generic-connid***)**
The identifier of the connection definition for which information is to be displayed. A single asterisk (*) specifies that information for all connection identifiers is to be displayed.

When an application connects to WebSphere MQ, it is given a unique 24–byte connection identifier (ConnectionId). The value for CONN is formed by converting the last eight bytes of the ConnectionId to its 16–character hexadecimal equivalent.

WHERE
Specify a filter condition to display only those connections that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Almost any parameter that can be used to display attributes for this DISPLAY command. However, you cannot use the CMDSCOPE, EXTCONN, QSGDISP, TYPE, and EXTURID parameters as filter keywords.

operator
This is used to determine whether a connection satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

CT Contains a specified item. If the *filter-keyword* is a list, you can use this to display objects the attributes of which contain the specified item. You cannot use the CONNOPTS value MQCNO_STANDARD_BINDING with this operator.

EX Does not contain a specified item. If the *filter-keyword* is a list, you can use this to display objects the attributes of which do not contain the specified item. You cannot use the CONNOPTS value MQCNO_STANDARD_BINDING with this operator.

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:
- An explicit value, that is a valid value for the attribute being tested.

You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter (for example, the value NONE on the UOWSTATE parameter), you can only use EQ or NE.

- A generic value. This is a character string (such as the character string in the APPLTAG parameter) with an asterisk at the end, for example ABC*. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

  You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.
- An item in a list of values. Use CT or EX as the operator. For example, if the value DEF is specified with the operator CT, all items where one of the attribute values is DEF are listed.

ALL Specify this to display all the connection information of the requested type for each specified connection. This is the default if you do not specify a generic identifier, 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.

' ' 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.

You cannot use CMDSCOPE as a filter keyword.

EXTCONN

The value for EXTCONN is based on the first sixteen bytes of the ConnectionId converted to its 32–character hexadecimal equivalent.

Connections are identified by a 24-byte connection identifier. The connection identifier comprises a prefix, which identifies the queue manager, and a suffix which identifies the connection to that queue manager. By default, the prefix is for the queue manager currently being administered, but you can specify a prefix explicitly by using the EXTCONN parameter. Use the CONN parameter to specify the suffix.

When connection identifiers are obtained from other sources, specify the fully qualified connection identifier (both EXTCONN and CONN) to avoid possible problems related to non-unique CONN values.

Do not specify both a generic value for CONN and a non-generic value for EXTCONN.

You cannot use EXTCONN as a filter keyword.

TYPE Specifies the type of information to be displayed. Values are:

CONN
Connection information for the specified connection. On z/OS, this includes threads which may be logically or actually disassociated from a connection, together with those that are in-doubt and for which external intervention is needed to resolve them. These latter threads are those that DIS THREAD TYPE(INDOUBT) would show.

HANDLE
Information relating to any objects opened by the specified connection.

***** Display all available information relating to the connection.

ALL Display all available information relating to the connection.

Conn attrs

If TYPE is set to CONN, the following information is always returned for each connection that satisfies the selection criteria, except where indicated:

- Connection identifier (CONN parameter)
- Type of information returned (TYPE parameter)

The following parameters can be specified for TYPE(CONN) to request additional information for each connection. If a parameter is specified that is not relevant for the connection, operating environment, or type of 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
- i5/OS job name
- UNIX process
- Windows process

  Note: This consists of the full program path and executable file name. If it is more than 28 characters long, only the last 28 characters will be shown.
- Internal queue manager process name

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

i5/OS An i5/OS application

SYSTEM
Queue manager

UNIX A UNIX application

USER A user application

WINDOWSNT
A Windows application

ASID A 4-character address-space identifier of the application identified by APPLTAG. It distinguishes duplicate values of APPLTAG.

This parameter is returned only on z/OS when the APPLTYPE parameter does not have the value SYSTEM.

This parameter is valid only on z/OS.

ASTATE
The state of asynchronous consumption on this connection handle.

Possible values are:

SUSPENDED
An MQCTL call with the Operation parameter set to MQOP_SUSPEND has been issued against the connection handle so that so that asynchronous message consumption is temporarily suspended on this connection.

STARTED
An MQCTL call with the Operation parameter set to MQOP_START has been issued against the connection handle so that asynchronous message consumption can proceed on this connection.

STARTWAIT
An MQCTL call with the Operation parameter set to MQOP_START_WAIT has been issued against the connection handle so that asynchronous message consumption can proceed on this connection.

STOPPED
An MQCTL call with the Operation parameter set to MQOP_STOP has been issued against the connection handle so that asynchronous message consumption cannot currently proceed on this connection.

NONE
No MQCTL call has been issued against the connection handle. Asynchronous message consumption cannot currently proceed on this connection.

CHANNEL
The name of the channel that owns the connection. If there is no channel associated with the connection, this parameter is blank.

CONNAME
The connection name associated with the channel that owns the connection. If there is no channel associated with the connection, this parameter is blank.

CONNOPTS
The connect options currently in force for this application connection. Possible values are:

- MQCNO_HANDLE_SHARE_BLOCK
- MQCNO_HANDLE_SHARE_NO_BLOCK
- MQCNO_HANDLE_SHARE_NONE
- MQCNO_SHARED_BINDING
- MQCNO_STANDARD_BINDING
- MQCNO_ISOLATED_BINDING
- MQCNO_FASTPATH_BINDING
- MQCNO_SERIALIZE_CONN_TAG_Q_MGR
- MQCNO_SERIALIZE_CONN_TAG_QSG
- MQCNO_RESTRICT_CONN_TAG_Q_MGR
- MQCNO_RESTRICT_CONN_TAG_QSG
- MQCNO_ACCOUNTING_Q_ENABLED
- MQCNO_ACCOUNTING_Q_DISABLED
- MQCNO_ACCOUNTING_MQI_ENABLED
- MQCNO_ACCOUNTING_MQI_DISABLED

You cannot use the value MQCNO_STANDARD_BINDING as a filter value with the CT and EX operators on the WHERE parameter.

EXTURID
The external unit of recovery identifier associated with this connection. Its format is determined by the value of URTYPE.

You cannot use EXTURID as a filter keyword.

NID Origin identifier, set only if the value of UOWSTATE is UNRESOLVED. This is a unique token identifying the unit of work within the queue manager. It is of the form `origin-node.origin-urid` where

- `origin-node` identifies the originator of the thread, except in the case where APPLTYPE is set to RRSBATCH, when it is omitted.
- `origin-urid` is the hexadecimal number assigned to the unit of recovery by the originating system for the specific thread to be resolved.

This parameter is valid only on z/OS.

PID Number specifying the process identifier of the application that is connected to the queue manager.

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.

QMURID
The queue manager unit of recovery identifier. On z/OS, this is a 6–byte log RBA, displayed as 12 hexadecimal characters. On platforms other than z/OS, this is an 8–byte transaction identifier, displayed as `m.n` where `m` and `n` are the decimal representation of the first and last 4 bytes of the transaction identifier.

You can use QMURID as a filter keyword. On z/OS, you must specify the filter value as a hexadecimal string. On platforms other than z/OS, you must specify the filter value as a pair of decimal numbers separated by a period (.). You can only use the EQ, NE, GT, LT, GE, or LE filter operators.

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 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.

UOWLOG
The filename of the extent to which the transaction associated with this connection first wrote.

This parameter is valid only on platforms other than z/OS.

UOWLOGDA
The date that the transaction associated with the current connection first wrote to the log.

UOWLOGTI
The time that the transaction associated with the current connection first wrote to the log.

UOWSTATE
The state of the unit of work. It is one of the following:

NONE
There is no unit of work.

ACTIVE
The unit of work is active.

PREPARED
The unit of work is in the process of being committed.

UNRESOLVED
The unit of work is in the second phase of a two-phase commit operation. WebSphere MQ holds resources on its behalf and external intervention is required to resolve it. This might be as simple as starting the recovery coordinator (such as CICS, IMS, or RRS) or it might involve a more complex operation such as using the RESOLVE INDOUBT command. The UNRESOLVED value can occur only on z/OS.

UOWSTDA
The date that the transaction associated with the current connection was started.

UOWSTTI
The time that the transaction associated with the current connection was started.

URTYPE
The type of unit of recovery as seen by the queue manager. It is one of the following:
- CICS (valid only on z/OS)
- XA
- RRS (valid only on z/OS)
- IMS (valid only on z/OS)
- QMGR

URTYPE identifies the EXTURID type and not the type of the transaction coordinator. When URTYPE is QMGR, the associated identifier is in QMURID (and not EXTURID).

USERID
The user identifier associated with the connection.

This parameter is not returned when APPLTYPE has the value SYSTEM.

Handle attrs

If TYPE is set to HANDLE, the following information is always returned for each connection that satisfies the selection criteria, except where indicated:

- Connection identifier (CONN parameter)
- Read ahead status (DEFREADA parameter)
- Type of information returned (TYPE parameter)
- Handle status (HSTATE)
- Object name (OBJNAME parameter)
- Object type (OBJTYPE parameter)

The following parameters can be specified for TYPE(HANDLE) to request additional information for each queue. If a parameter is specified that is not relevant for the connection, operating environment, or type of status information requested, that parameter is ignored.

ASTATE
The state of the asynchronous consumer on this object handle.

Possible values are:

ACTIVE
An MQCB call has set up a function to call back to process

messages asynchronously and the connection handle has been started so that asynchronous message consumption can proceed.

INACTIVE
An MQCB call has set up a function to call back to process messages asynchronously but the connection handle has not yet been started, or has been stopped or suspended, so that asynchronous message consumption cannot currently proceed.

SUSPENDED
The asynchronous consumption call back has been suspended so that asynchronous message consumption cannot currently proceed on this object handle. This can be either because an MQCB call with Operation MQOP_SUSPEND has been issued against this object handle by the application, or because it has been suspended by the system. If it has been suspended by the system, as part of the process of suspending asynchronous message consumption the call back function will be called with the reason code that describes the problem resulting in suspension. This will be reported in the Reason field in the MQCBC structure that is passed to the call back function.

For asynchronous message consumption to proceed, the application must issue an MQCB call with the Operation parameter set to MQOP_RESUME.

SUSPTEMP
The asynchronous consumption call back has been temporarily suspended by the system so that asynchronous message consumption cannot currently proceed on this object handle. As part of the process of suspending asynchronous message consumption, the call back function will be called with the reason code that describes the problem resulting in suspension. This will be reported in the Reason field in the MQCBC structure passed to the call back function.

The call back function will be called again when asynchronous message consumption is resumed by the system, when the temporary condition has been resolved.

NONE
An MQCB call has not been issued against this handle, so no asynchronous message consumption is configured on this handle.

DEST The destination queue for messages that are published to this subscription. This parameter is only relevant for handles of subscriptions to topics.

DESTQMGR
The destination queue manager for messages that are published to this subscription. This parameter is relevant only for handles of subscriptions to topics. If DEST is a queue that is hosted on the local queue manager, this parameter will contain the local queue manager name. If DEST is a queue that is hosted on a remote queue manager, this parameter will contain the name of the remote queue manager.

HSTATE
The state of the handle.

Possible values are:

ACTIVE
An API call from this connection is currently in progress for this object. If the object is a queue, this condition can arise when an MQGET WAIT call is in progress.

If there is an MQGET SIGNAL outstanding, then this does not mean, by itself, that the handle is active.

INACTIVE
No API call from this connection is currently in progress for this object. If the object is a queue, this condition can arise when no MQGET WAIT call is in progress.

OBJNAME
The name of an object that the connection has open.

OBJTYPE
The type of the object that the connection has open. If this handle is that of a subscription to a topic, then the SUBID parameter identifies the subscription. You can then use the DISPLAY SUB command to find all the details about the subscription.

It is one of the following:
- QUEUE
- PROCESS
- QMGR
- STGCLASS (valid only on z/OS)
- NAMELIST
- CHANNEL
- AUTHINFO
- TOPIC

OPENOPTS
The open options currently in force for the connection for the object. This parameter is not relevant for a subscription. Use the value in the SUBID parameter and the DISPLAY SUB command to find the details about the subscription.

Possible values are:

MQOO_INPUT_AS_Q_DEF
Open queue to get messages using queue-defined default.

MQOO_INPUT_SHARED
Open queue to get messages with shared access.

MQOO_INPUT_EXCLUSIVE
Open queue to get messages with exclusive access.

MQOO_BROWSE
Open queue to browse messages.

MQOO_OUTPUT
Open queue or topic to put messages.

MQOO_INQUIRE
Open queue to inquire attributes.

MQOO_SET
Open queue to set attributes.

MQOO_BIND_ON_OPEN
Bind handle to destination when queue is found.

MQOO_BIND_NOT_FIXED
Do not bind to a specific destination.

MQOO_SAVE_ALL_CONTEXT
Save context when message retrieved.

MQOO_PASS_IDENTITY_CONTEXT
Allow identity context to be passed.

MQOO_PASS_ALL_CONTEXT
Allow all context to be passed.

MQOO_SET_IDENTITY_CONTEXT
Allow identity context to be set.

MQOO_SET_ALL_CONTEXT
Allow all context to be set.

MQOO_ALTERNATE_USER_AUTHORITY
Validate with specified user identifier.

MQOO_FAIL_IF_QUIESCING
Fail if queue manager is quiescing.

QSGDISP
Indicates the disposition of the object. 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).

You cannot use QSGDISP as a filter keyword.

READA
The read ahead connection status.

Possible values are:

NO Read ahead of non-persistent messages is not enabled for this object.

YES Read ahead of non-persistent message is enabled for this object and is being used efficiently.

BACKLOG
Read ahead of non-persistent messages is enabled for this object. Read ahead is not being used efficiently because the client has been sent a large number of messages which are not being consumed.

INHIBITED
Read ahead was requested by the application but has been inhibited because of incompatible options specified on the first MQGET call.

SUBID
The internal, all-time unique identifier of the subscription. This parameter is relevant only for handles of subscriptions to topics.

Not all subscriptions show up in DISPLAY CONN; only those that have current handles open to the subscription show up. You can use the DISPLAY SUB command to see all subscriptions.

SUBNAME
The application's unique subscription name that is associated with the handle. This parameter is relevant only for handles of subscriptions to topics. Not all subscriptions will have a subscription name.

TOPICSTR
The resolved topic string. This parameter is relevant for handles with OBJTYPE(TOPIC). For any other object type, this parameter is blank.

Full attrs

If TYPE is set to *, or ALL, both Conn attributes and Handle attributes are returned for each connection that satisfies the selection criteria.

Usage notes for DISPLAY CONN

1. This command is issued internally by WebSphere MQ on z/OS when taking a checkpoint, and when the queue manager is starting and stopping, so that a list of units of work that are in doubt at the time is written to the z/OS console log.
2. The TOPICSTR parameter might contain characters that cannot be translated into printable characters when the command output is displayed. On z/OS, these non-printable characters will be displayed as blanks. On distributed platforms using runmqsc, these non-printable characters will be displayed as dots.
3. The state of asynchronous consumers, ASTATE, reflects that of the server-connection proxy on behalf of the client application; it does not reflect the client application state.

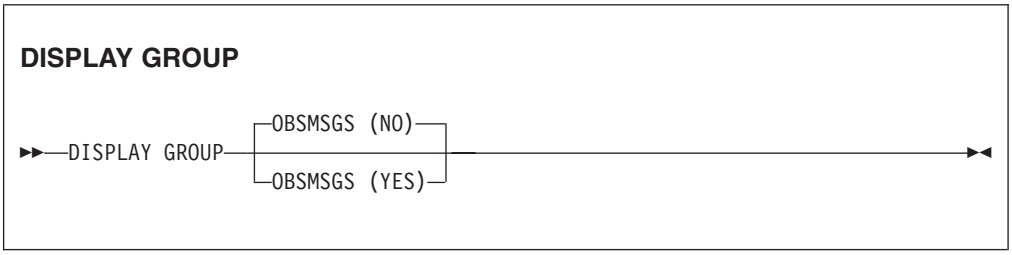
DISPLAY GROUP

Use the MQSC command DISPLAY GROUP to display information about the queue-sharing group to which the queue manager is connected. This command is valid only when the queue manager is a member of a queue-sharing group.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS GROUP



Parameter descriptions for DISPLAY GROUP

OBSMSGS
Specifies whether the command additionally looks for obsolete messages in DB2. This is optional. Possible values are:

NO Obsolete messages in DB2 are not looked for. This is the default value.

YES Obsolete messages in DB2 are looked for and messages containing information about any found are returned.

Usage notes for DISPLAY GROUP

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.
   - If you specify OBSMSGS (YES), whether queue managers in the group contain obsolete messages in DB2

DISPLAY LISTENER

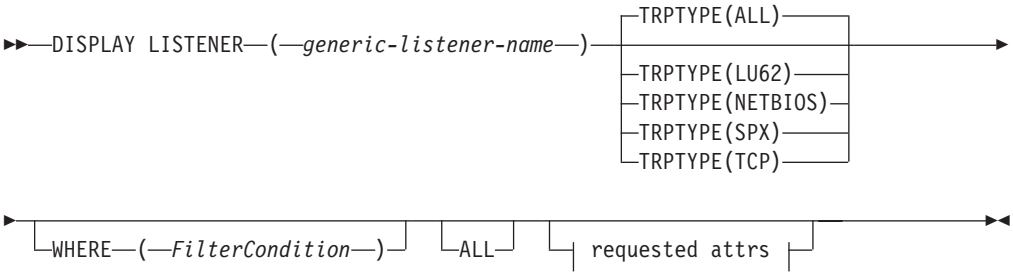
Use the MQSC command DISPLAY LISTENER to display information about a listener.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	

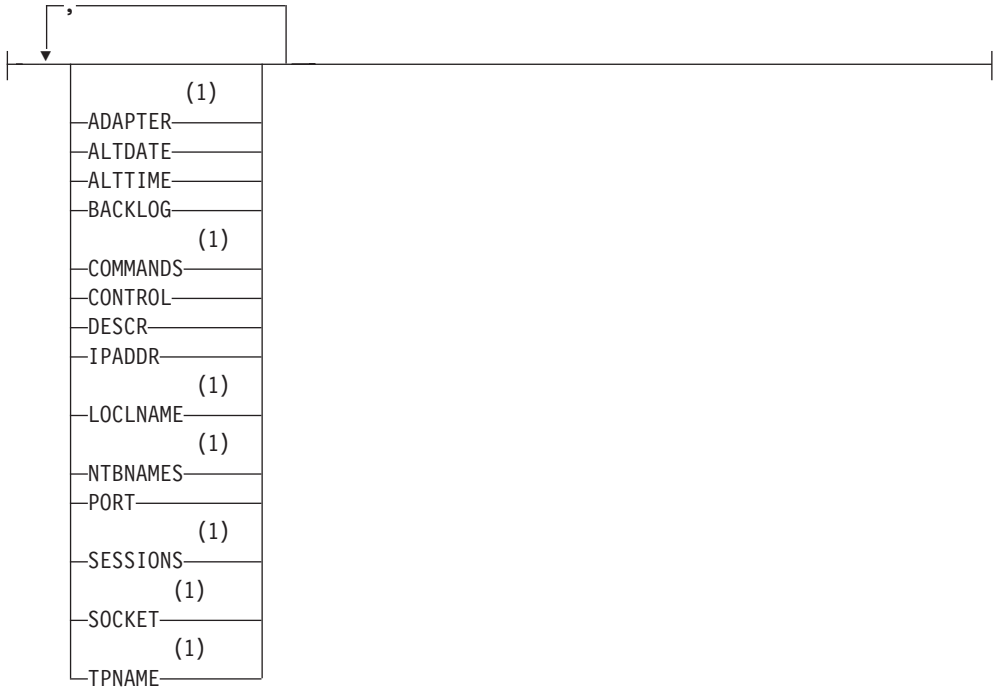
The values displayed describe the current definition of the listener. If the listener has been altered since it was started, the currently running instance of the listener object may not have the same values as the current definition.

Synonym: DIS LSTR

DISPLAY LISTENER



Requested attrs:



Notes:

1 Valid only on Windows.

Keyword and parameter descriptions for DISPLAY LISTENER

You must specify a listener for which you want to display information. You can specify a listener by using either a specific listener name or a generic listener name. By using a generic listener name, you can display either:

- Information about all listener definitions, by using a single asterisk (*), or
- Information about one or more listeners that match the specified name.

(*generic-listener-name***)**
The name of the listener definition for which information is to be displayed. A single asterisk (*) specifies that information for all listener identifiers is to be displayed. A character string with an asterisk at the end matches all listeners with the string followed by zero or more characters.

TRPTYPE
Transmission protocol. If you specify this parameter, it must follow directly after the *generic-listener-name* parameter. If you do not specify this parameter, a default of ALL is assumed. Values are:

ALL This is the default value and displays information for all listeners.

LU62 Displays information for all listeners defined with a value of LU62 in their TRPTYPE parameter.

NETBIOS
Displays information for all listeners defined with a value of NETBIOS in their TRPTYPE parameter.

SPX Displays information for all listeners defined with a value of SPX in their TRPTYPE parameter.

TCP Displays information for all listeners defined with a value of TCP in their TRPTYPE parameter.

WHERE
Specify a filter condition to display information for those listeners that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Any parameter that can be used to display attributes for this DISPLAY command.

operator
This is used to determine whether a listener satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.
- A generic value. This is a character string. with an asterisk at the end, for example ABC*. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

  You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.

ALL Specify this to display all the listener information for each specified listener. 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 identifier, and do not request any specific parameters.

On z/OS this is also the default if you specify a filter condition using the WHERE parameter, but on other platforms only requested attributes are 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.

ADAPTER
The adapter number on which NetBIOS listens.

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`.

BACKLOG
The number of concurrent connection requests that the listener supports.

COMMANDS
The number of commands that the listener can use.

CONTROL
How the listener is to be started and stopped:

MANUAL
The listener is not to be started automatically or stopped automatically. It is to be controlled by use of the START LISTENER and STOP LISTENER commands.

QMGR
The listener being defined is to be started and stopped at the same time as the queue manager is started and stopped.

STARTONLY
The listener is to be started at the same time as the queue manager is started, but is not requested to stop when the queue manager is stopped.

DESCR
Descriptive comment.

IPADDR
The listener's IP address.

LOCLNAME
The NetBIOS local name that the listener uses.

NTBNAMES
The number of names that the listener can use.

PORT The port number for TCP/IP.

SESSIONS
The number of sessions that the listener can use.

SOCKET
SPX socket.

TPNAME
The LU6.2 transaction program name.

For more information on these parameters, see "DEFINE LISTENER" on page 208.

DISPLAY LOG

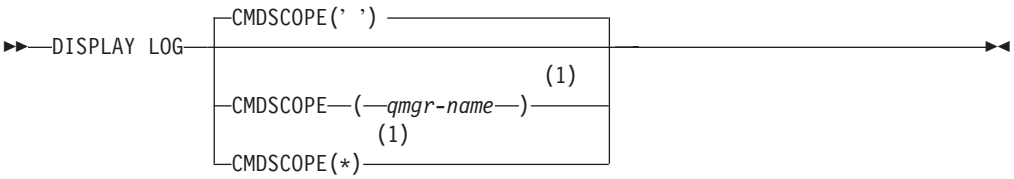
Use the MQSC command DISPLAY LOG to display log system parameters and information.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS LOG

DISPLAY LOG



Notes:

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

Parameter descriptions for DISPLAY LOG

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 for DISPLAY LOG

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.

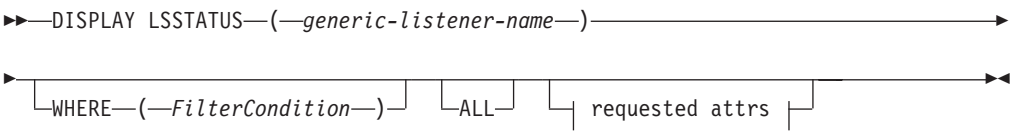
DISPLAY LSSTATUS

Use the MQSC command DISPLAY LSSTATUS to display status information for one or more listeners.

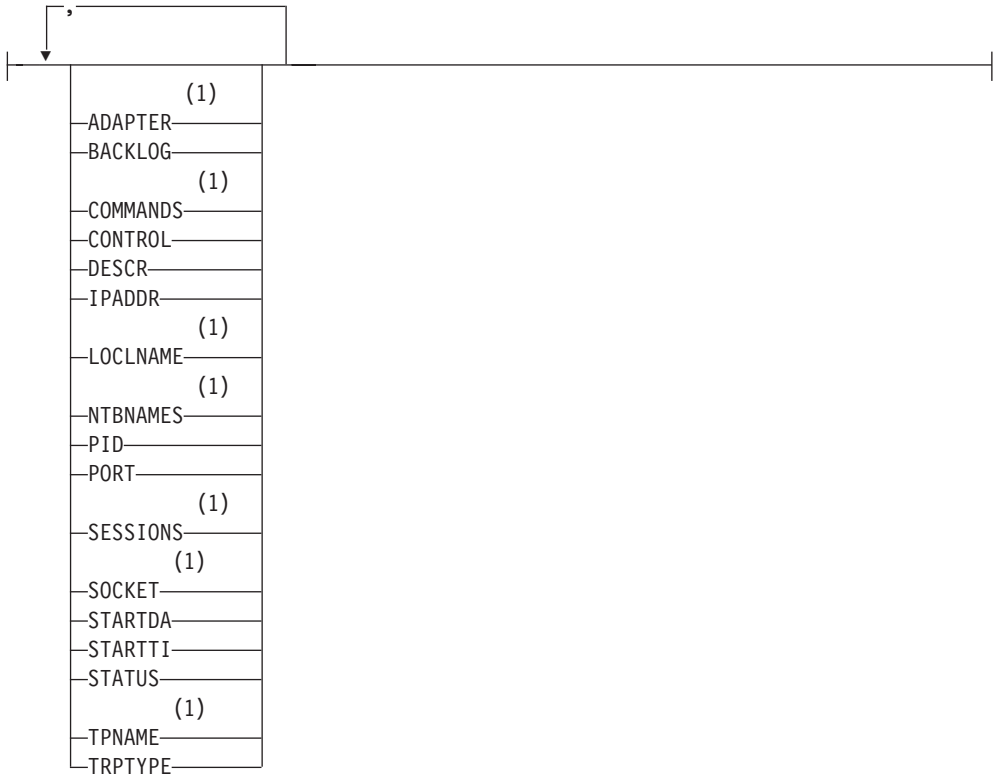
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	

Synonym: DIS LSSTATUS

DISPLAY LSSTATUS



Requested attrs:



Notes:

1 Valid only on Windows.

Keyword and parameter descriptions for DISPLAY LSSTATUS

You must specify a listener for which you want to display status information. You can specify a listener by using either a specific listener name or a generic listener name. By using a generic listener name, you can display either:

- Status information for all listener definitions, by using a single asterisk (*), or
- Status information for one or more listeners that match the specified name.

(*generic-listener-name***)**
The name of the listener definition for which status information is to be displayed. A single asterisk (*) specifies that information for all connection identifiers is to be displayed. A character string with an asterisk at the end matches all listeners with the string followed by zero or more characters.

WHERE
Specify a filter condition to display information for those listeners that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Any parameter that can be used to display attributes for this DISPLAY command.

operator
This is used to determine whether a listener satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:
- An explicit value, that is a valid value for the attribute being tested.
- A generic value. This is a character string. with an asterisk at the end, for example ABC*. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

  You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.

ALL Display all the status information for each specified listener. This is the default if you do not specify a generic name, and do not request any specific parameters.

On z/OS this is also the default if you specify a filter condition using the WHERE parameter, but on other platforms only requested attributes are 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.

ADAPTER
The adapter number on which NetBIOS listens.

BACKLOG
The number of concurrent connection requests that the listener supports.

CONTROL
How the listener is to be started and stopped:

MANUAL
The listener is not to be started automatically or stopped automatically. It is to be controlled by use of the START LISTENER and STOP LISTENER commands.

QMGR
The listener being defined is to be started and stopped at the same time as the queue manager is started and stopped.

STARTONLY
The listener is to be started at the same time as the queue manager is started, but is not requested to stop when the queue manager is stopped.

DESCR
Descriptive comment.

IPADDR
The listener's IP address.

LOCLNAME
The NetBIOS local name that the listener uses.

NTBNAMES
The number of names that the listener can use.

PID The operating system process identifier associated with the listener.

PORT The port number for TCP/IP.

SESSIONS
The number of sessions that the listener can use.

SOCKET
SPX socket.

STARTDA
The date on which the listener was started.

STARTTI
The time at which the listener was started.

STATUS
The current status of the listener. It can be one of:

RUNNING
The listener is running.

STARTING
The listener is in the process of initializing.

STOPPING
The listener is stopping.

TPNAME
The LU6.2 transaction program name.

TRPTYPE
Transport type.

For more information on these parameters, see “DEFINE LISTENER” on page 208.

DISPLAY MAXSMSGS

Use the MQSC command DISPLAY MAXSMSGS to see the maximum number of messages that a task can get or put within a single unit of recovery.

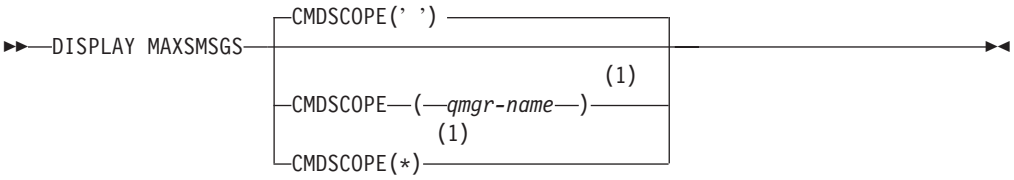
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note: This command is valid only on z/OS and is retained for compatibility with earlier releases, 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



Notes:

1 Valid only on full function WebSphere MQ for z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for DISPLAY MAXSMSGS

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.

DISPLAY NAMELIST

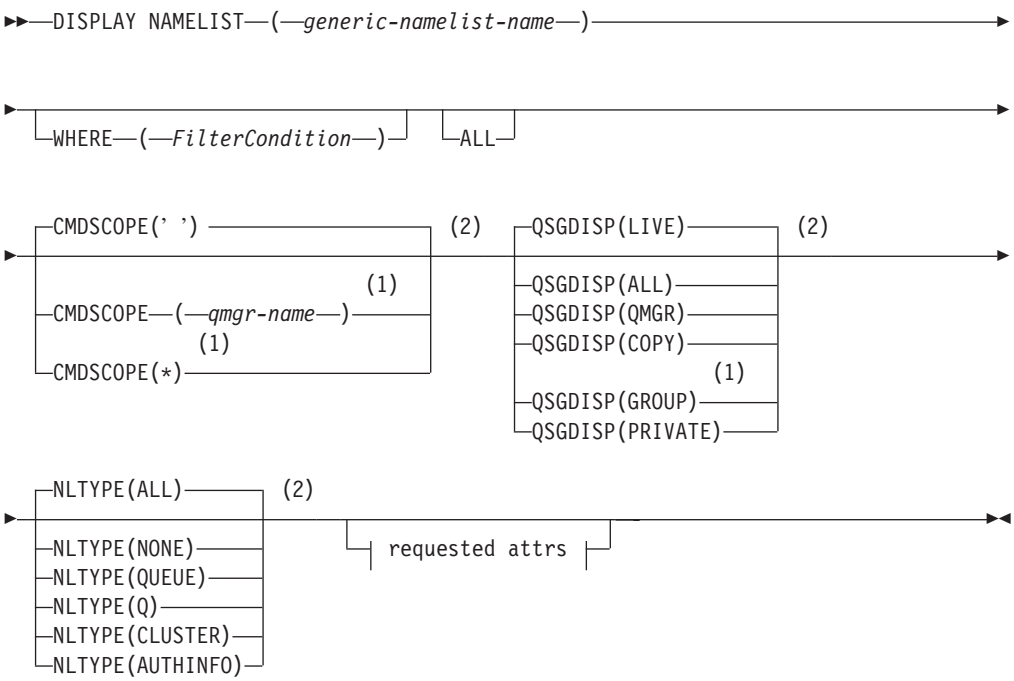
Use the MQSC command DISPLAY NAMELIST to display the names in a namelist.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS NL

DISPLAY NAMELIST



Requested attrs:

Notes:

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

2 Valid only on z/OS.

Parameter descriptions for DISPLAY NAMELIST

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 WebSphere MQ objects" on page 5). A trailing asterisk (*) matches all namelists with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all namelists.

WHERE
Specify a filter condition to display only those namelists that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Almost any parameter that can be used to display attributes for this DISPLAY command. However, you cannot use the CMDSCOPE or QSGDISP parameters as filter keywords. You cannot use NLTYPE as a filter keyword if you also use it to select namelists.

operator
This is used to determine whether a namelist satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

CT Contains a specified item. If the *filter-keyword* is a list, you can use this to display objects the attributes of which contain the specified item.

EX Does not contain a specified item. If the *filter-keyword* is a list, you can use this to display objects the attributes of which do not contain the specified item.

CTG Contains an item which matches a generic string that you provide as a *filter-value*. If the *filter-keyword* is a list, you can use this to display objects the attributes of which match the generic string.

EXG Does not contain any item which matches a generic string that you provide as a *filter-value*. If the *filter-keyword* is a list, you can use this to display objects the attributes of which do not match the generic string.

filter-value

The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.

  You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter (for example, the value NONE on the NLTYPE parameter), you can only use EQ or NE.
- A generic value. This is a character string (such as the character string you supply for the DESCR parameter) with an asterisk at the end, for example ABC*. The characters must be valid for the attribute you are testing. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

  You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.
- An item in a list of values. The value can be explicit or, if it is a character value, it can be explicit or generic. If it is explicit, use CT or EX as the operator. For example, if the value DEF is specified with the operator CT, all items where one of the attribute values is DEF are listed. If it is generic, use CTG or EXG as the operator. If ABC* is specified with the operator CTG, all items where one of the attribute values begins with ABC are listed.

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.

On z/OS this is also the default if you specify a filter condition using the WHERE parameter, but on other platforms only requested attributes are displayed.

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.

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:

QMGR

The object was defined with QSGDISP(QMGR).

GROUP

The object was defined with QSGDISP(GROUP).

COPY The object was defined with QSGDISP(COPY).

You cannot use QSGDISP as a filter keyword.

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, on z/OS, their NLTYPEs and QSGDISP 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

NAMCOUNT
Number of names in the list

NAMES
List of names

See "DEFINE NAMELIST" on page 216 for more information about the individual parameters.

DISPLAY PROCESS

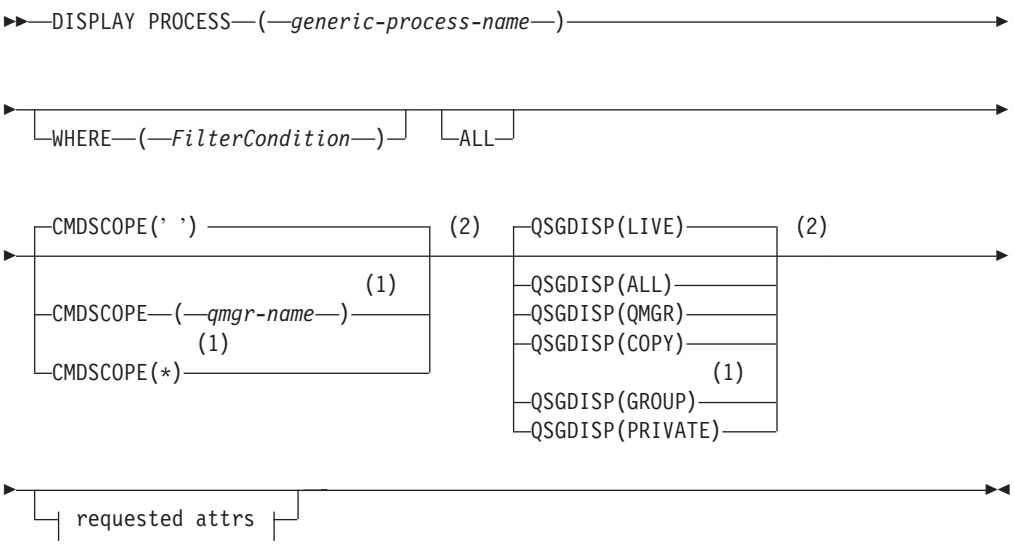
Use the MQSC command DISPLAY PROCESS to display the attributes of one or more WebSphere MQ processes.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

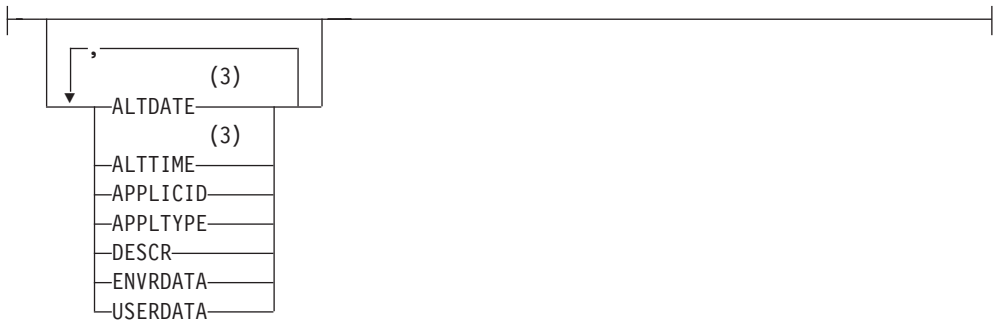
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS PRO

DISPLAY PROCESS



Requested attrs:



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

3 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

Parameter descriptions for DISPLAY PROCESS

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
- One or more processes that match the specified name

(generic-process-name)
The name of the process definition to be displayed (see "Rules for naming WebSphere MQ objects" on page 5). 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 defined to the local queue manager.

WHERE
Specify a filter condition to display only those process definitions that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Almost any parameter that can be used to display attributes for this DISPLAY command. However, you cannot use the CMDSCOPE or QSGDISP parameters as filter keywords.

operator
This is used to determine whether a process definition satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.

  You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter (for example, the value DEF on the APPLTYPE parameter), you can only use EQ or NE.
- A generic value. This is a character string (such as the character string you supply for the DESCR parameter) with an asterisk at the end, for example ABC*. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are

listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

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, HP OpenVMS, HP-UX, Linux, i5/OS, 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 Compaq NonStop Kernel, if no parameters are specified (and the ALL parameter is not specified), the default is that the process names are returned.

On z/OS this is also the default if you specify a filter condition using the WHERE parameter, but on other platforms only requested attributes are displayed.

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.

You cannot use CMDSCOPE as a filter keyword.

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:

QMGR
The object was defined with QSGDISP(QMGR).

GROUP
The object was defined with QSGDISP(GROUP).

COPY The object was defined with QSGDISP(COPY).

You cannot use QSGDISP as a filter keyword.

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, on z/OS only, QSGDISP 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. In addition to the values listed for this parameter in "Parameter descriptions for DEFINE PROCESS" on page 222, the value SYSTEM can be displayed. This indicates that the application type is a queue manager.

DESCR
Description

ENVRDATA
Environment data

USERDATA
User data

See "DEFINE PROCESS" on page 221 for more information about individual parameters.

DISPLAY PUBSUB

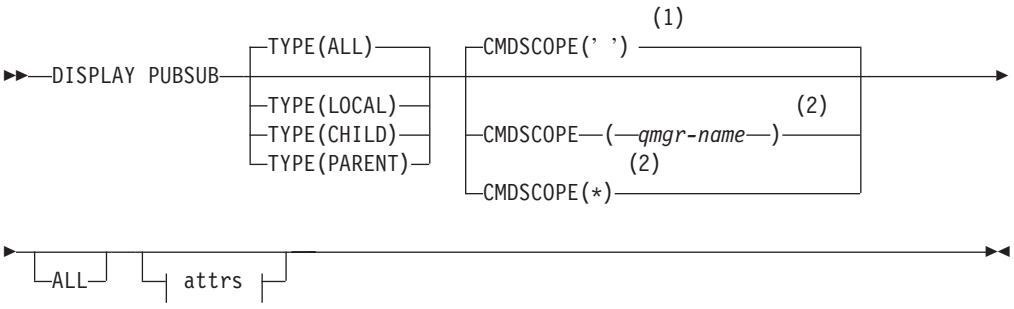
Use the MQSC command DISPLAY PUBSUB to display publish/subscribe status information for a queue manager.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: None

DISPLAY PUBSUB



Attrs:

Notes:

1 Valid only on z/OS.

2 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for DISPLAY PUBSUB

This describes the parameters for the MQSC command DISPLAY PUBSUB

TYPE The type of publish/subscribe connections.

ALL Display the publish/subscribe status for this queue manager and for parent and child hierarchical connections.

CHILD
Display the publish/subscribe status for child connections.

LOCAL
Display the publish/subscribe status for this queue manager.

PARENT
Display the publish/subscribe status for the parent connection.

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.

You cannot use CMDSCOPE as a filter keyword.

Returned parameters

A group of returned parameters is returned, containing the attributes TYPE, QMNAME, and STATUS. This group is returned for the current queue manager if you set TYPE to LOCAL or ALL, for the parent queue manager if you set TYPE to PARENT or ALL, and for each child queue manager if you set TYPE to CHILD or ALL.

TYPE

CHILD
A child connection.

LOCAL
Information for this queue manager.

PARENT
The parent connection.

QMNAME
The name of the current queue manager or the remote queue manager connected as a parent or a child.

STATUS
The status of the publish/subscribe engine or the hierarchical connection.

When TYPE is LOCAL, the following values can be returned:

ACTIVE
The publish/subscribe engine and the queued publish/subscribe interface are running. It is therefore possible to publish or subscribe using the application programming interface and the queues that are monitored by the queued publish/subscribe interface.

COMPAT
The publish/subscribe engine is running. It is therefore possible to publish or subscribe by using the application programming interface. The queued publish/subscribe interface is not running.

Therefore, any message that is put to the queues that are monitored by the queued publish/subscribe interface will not be acted upon by WebSphere MQ.

ERROR
The publish/subscribe engine has failed. Check your error logs to determine the reason for the failure.

INACTIVE
The publish/subscribe engine and the queued publish/subscribe interface are not running. It is therefore not possible to publish or subscribe using the application programming interface. Any publish/subscribe messages that are put to the queues that are monitored by the queued publish/subscribe interface will not be acted upon by Websphere MQ.

STARTING
The publish/subscribe engine is initializing and is not yet operational.

STOPPING
The publish/subscribe engine is stopping.

When TYPE is PARENT, the following values can be returned:

ACTIVE
The connection with the parent queue manager is active.

ERROR
This queue manager is unable to initialize a connection with the parent queue manager because of a configuration error. Possible causes include:

- Transmit queue is not defined.
- Transmit queue put is disabled.

REFUSED
The connection has been refused by the parent queue manager. This might be caused by the following:

- The parent queue manager already has a child queue manager with the same name as this queue manager.
- The parent queue manager has used the command RESET QMGR TYPE(PUBSUB) CHILD to remove this queue manager as one of its children.

STARTING
The queue manager is attempting to request that another queue manager become its parent.

STOPPING
The queue manager is disconnecting from its parent.

When TYPE is CHILD, the following values can be returned:

ACTIVE
The connection with the child queue manager is active.

ERROR
This queue manager is unable to initialize a connection with the child queue manager because of a configuration error. Possible causes include:

- Transmit queue is not defined.
- Transmit queue put is disabled.

STARTING
Another queue manager is attempting to request that this queue manager become its parent.

STOPPING
The queue manager is disconnecting.

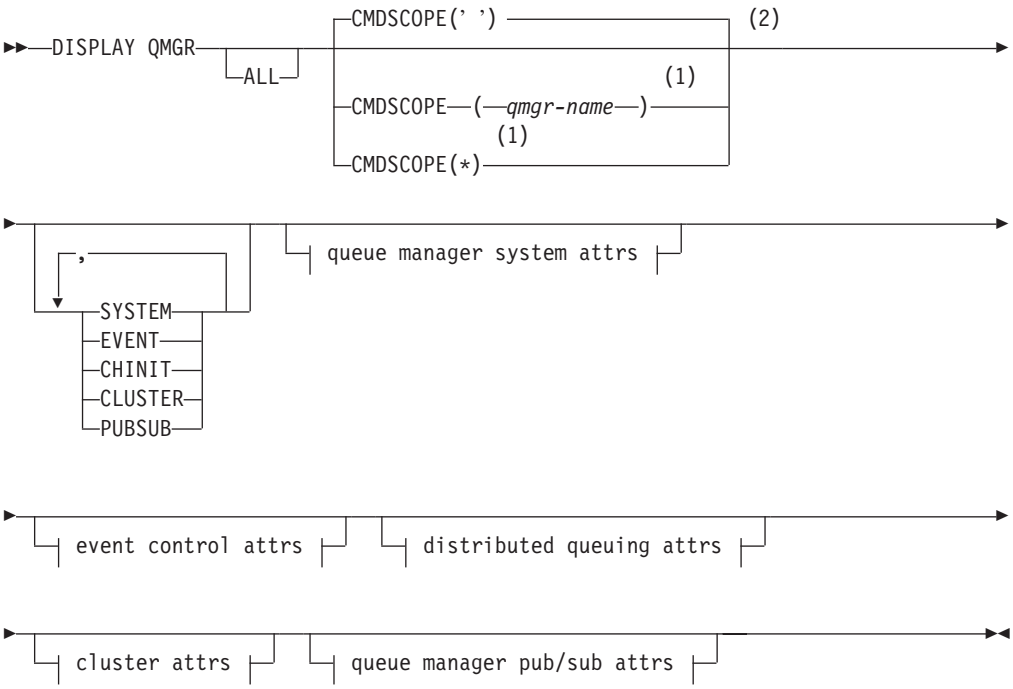
DISPLAY QMGR

Use the MQSC command DISPLAY QMGR to display the queue manager parameters for this queue manager.

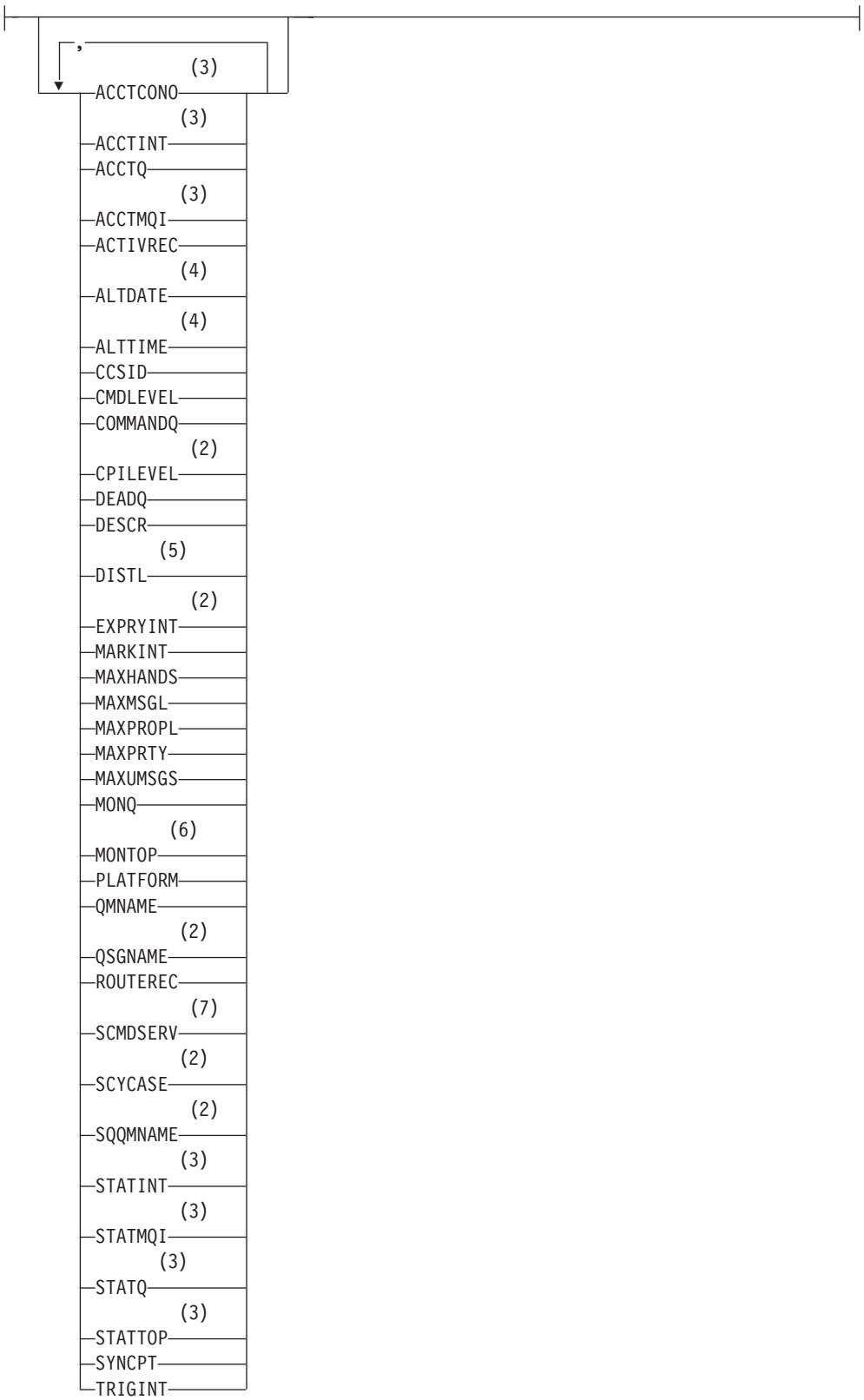
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

Synonym: DIS QMGR

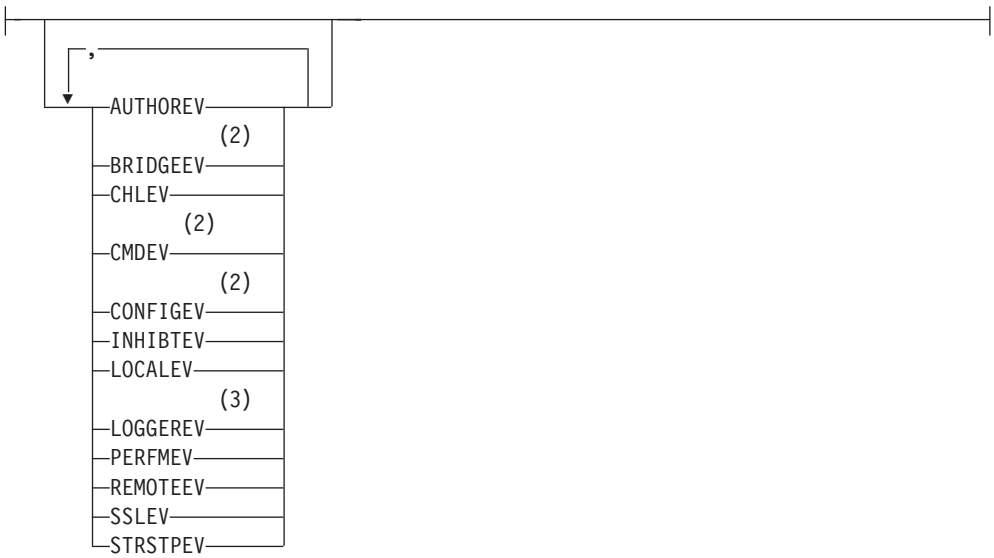
DISPLAY QMGR



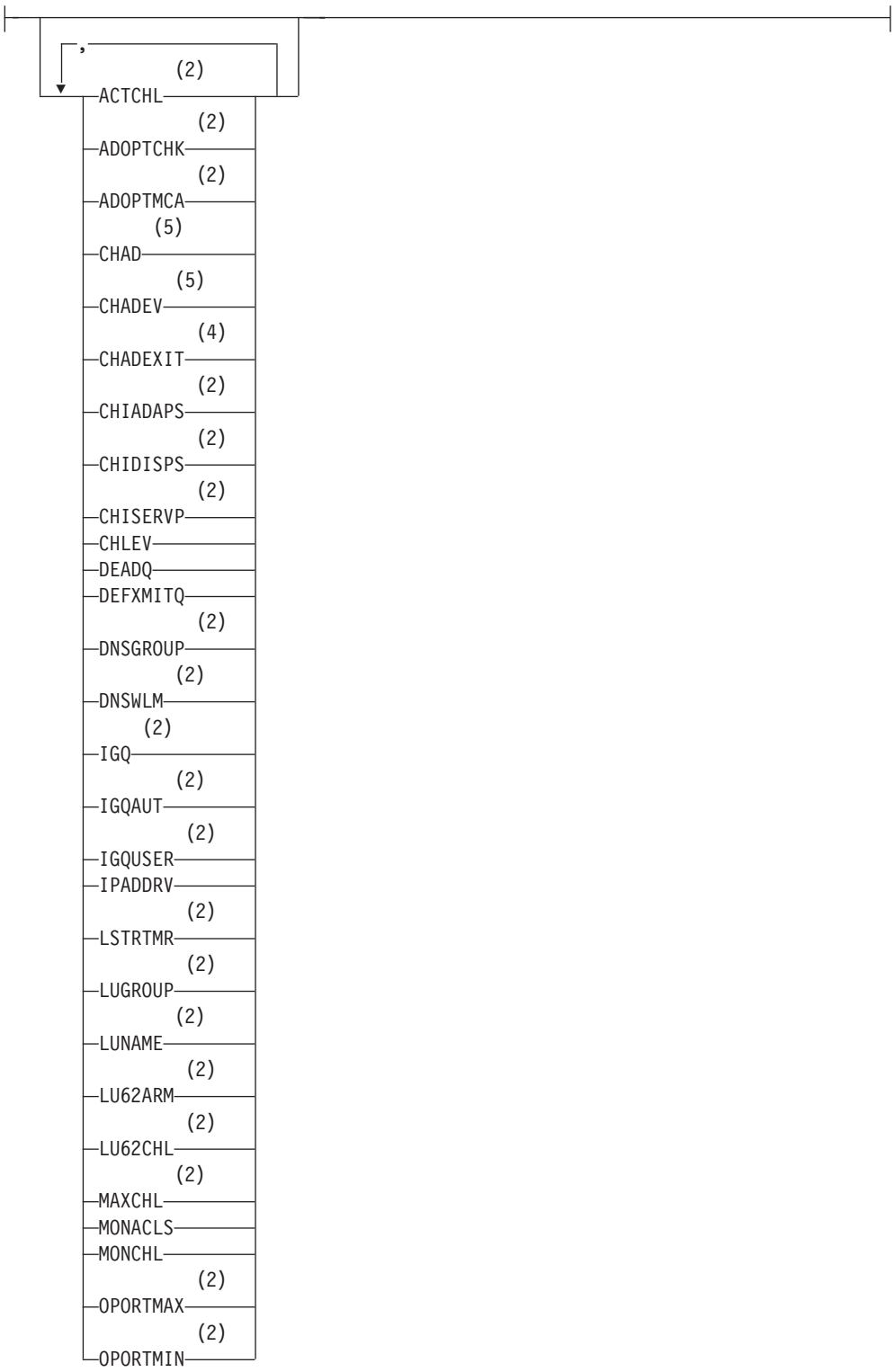
Queue manager system attributes:



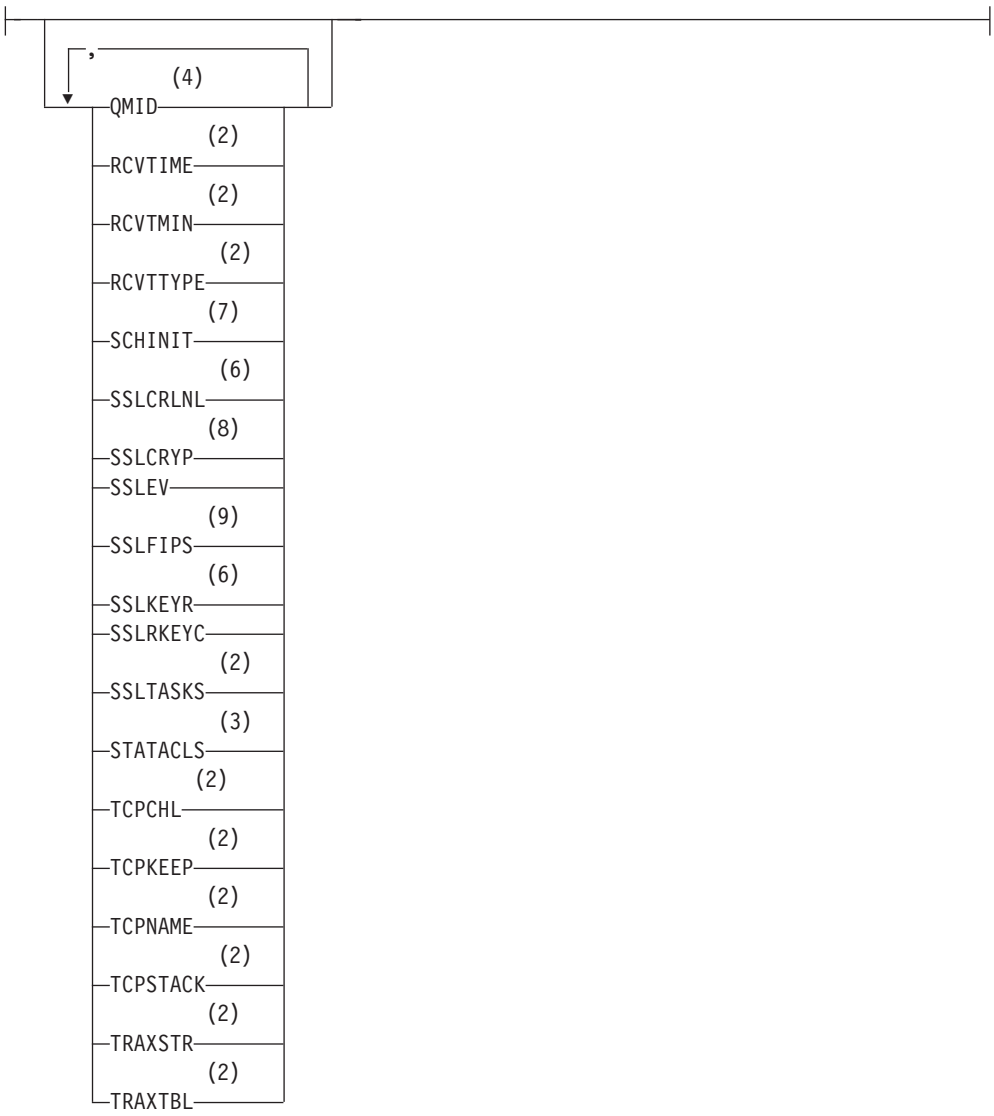
Event control attrs:



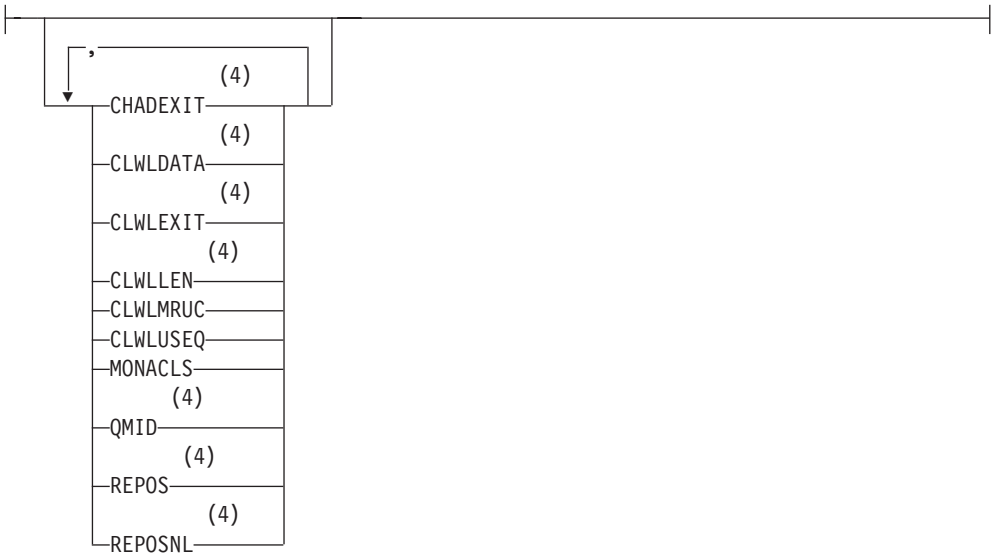
Distributed queuing attrs:



Distributed queuing attrs continued:


,
(4)
QMID
(2)
RCVTIME
(2)
RCVTMIN
(2)
RCVTTYPE
(7)
SCHINIT
(6)
SSLCRLNL
(8)
SSLCRYP
SSLEV
(9)
SSLFIPS
(6)
SSLKEYR
SSLRKEYC
(2)
SSLTASKS
(3)
STATACLS
(2)
TCPCHL
(2)
TCPKEEP
(2)
TCPNAME
(2)
TCPSTACK
(2)
TRAXSTR
(2)
TRAXTBL

Cluster attrs:



Queue manager pub/sub attrs:

Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

3 Valid only on i5/OS, UNIX systems, and Windows.

4 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

5 Valid only on AIX, HP OpenVMS, HP-UX, Linux, , i5/OS, Solaris, and Windows.

6 Valid only on AIX, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

7 Valid only on AIX, Compaq NSK, HP OpenVMS, HP-UX, Linux,, i5/OS, Solaris, and Windows.

8 Valid only on AIX, HP-UX, Linux, Solaris, and Windows.

9 Not valid on z/OS or i5/OS.

Parameter descriptions for DISPLAY QMGR

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, HP OpenVMS, HP-UX, Linux, i5/OS, 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.

' ' 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.

SYSTEM
Specify this to display the set of queue manager system attributes that are available in the 'Queue manager system attrs' list. See "Requested parameters" on page 421 for information about these parameters.

If you specify this parameter, any request you make to display individual parameters within this set has no effect.

EVENT
Specify this to display the set of event control attributes that are available in the 'Event control attrs' list. See "Requested parameters" on page 421 for information about these parameters.

If you specify this parameter, any request you make to display individual parameters within this set has no effect.

CHINIT
Specify this to display the set of attributes relating to distributed queuing that are available in the 'Distributed queuing attrs' list. You can also specify DQM to display the same set of attributes. See "Requested parameters" on page 421 for information about these parameters.

If you specify this parameter, any request you make to display individual parameters within this set has no effect.

CLUSTER
Specify this to display the set of attributes relating to clustering that are available in the 'Cluster attrs' list. See "Requested parameters" on page 421 for information about these parameters.

If you specify this parameter, any request you make to display individual parameters within this set has no effect.

PUBSUB
Specify this to display the set of attributes relating to publish/subscribe that are available in the 'Queue manager pub/sub attrs' list. See "Requested parameters" for information about these parameters.

If you specify this parameter, any request you make to display individual parameters within this set has no effect.

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:

ACCTCONO
Whether the settings of the ACCTQMQI and ACCTQ queue manager parameters may be overridden. This is valid only on i5/OS, UNIX systems, and Windows.

ACCTINT
The interval at which intermediate accounting records are written. This is valid only on i5/OS, UNIX systems, and Windows.

ACCTMQI
Whether accounting information is to be collected for MQI data. This is valid only on i5/OS, UNIX systems, and Windows.

ACCTQ
Whether accounting data collection is to be enabled for queues.

ACTCHL
The maximum number of channels that can be active at any time.

This parameter is valid only on z/OS.

ACTIVREC
Whether activity reports are to be generated if requested in the message.

ADOPTCHK
Which elements are checked to determine whether an MCA should be adopted when a new inbound channel is detected with the same name as an already active MCA.

This parameter is valid only on z/OS.

ADOPTMCA
Whether an orphaned MCA instance is to be restarted when a new inbound channel request matching the ADOPTCHK parameters is detected.

This parameter is valid only on z/OS.

ALTDATE
The date on which the definition was last altered, in the form `yyyy-mm-dd`.

This parameter is valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, 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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

AUTHOREV
Whether authorization events are generated.

BRIDGEEV
On z/OS only, whether IMS Bridge 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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

CHADEV
Whether auto-definition events are enabled. This parameter is valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

CHADEXIT
The name of the channel auto-definition exit. This parameter is valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows and z/OS.

CHIADAPS
The number of adapter subtasks to use to process WebSphere MQ calls.

This parameter is valid only on z/OS.

CHIDISPS
The number of dispatchers to use for the channel initiator.

This parameter is valid only on z/OS.

CHISERVP
This field is reserved for IBM use only.

CHLEV
Whether channel events are generated.

CLWLEXIT
The name of the cluster workload exit.

This parameter is valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

CLWLDATA
The data passed to the cluster workload exit.

This parameter is valid only on AIX, HP OpenVMS, HP-UX, Linux, , i5/OS, 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, HP OpenVMS, HP-UX, i5/OS, Solaris, Windows, and z/OS.

CLWLMRUC
The maximum number of outbound cluster channels.

CLWLUSEQ
The behavior of MQPUTs for queues where CLWLUSEQ has a value of QMGR.

CMDEV
Whether command events are generated.

This parameter is valid only on 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.

This parameter is valid only on z/OS.

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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

DNSGROUP
The name of the group that the TCP listener handling inbound transmissions for the queue-sharing group should join when using Workload Manager for Dynamic Domain Name Services support (WLM/DNS).

This parameter is valid only on z/OS.

DNSWLM
Whether the TCP listener that handles inbound transmissions for the queue-sharing group should register with WLM/DNS.

This parameter is valid only on z/OS.

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.

IPADDRV
Whether to use an IPv4 or IPv6 IP address for a channel connection in ambiguous cases.

LOCALEV
Whether local error events are generated.

LOGGEREV
Whether recovery log events are generated. This is valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

LSTRTMR
The time interval, in seconds, between attempts by WebSphere MQ to restart the listener after an APPC or TCP/IP failure.

This parameter is valid only on z/OS.

LUGROUP
The generic LU name to be used by the LU 6.2 listener that handles inbound transmissions for the queue-sharing group.

This parameter is valid only on z/OS.

LUNAME
The name of the LU to use for outbound LU 6.2 transmissions.

This parameter is valid only on z/OS.

LU62ARM
The suffix of the APPCPM member of SYS1.PARMLIB. This suffix

nominates the LUADD for this channel initiator. When automatic restart manager (ARM) restarts the channel initiator, the z/OS command SET APPC=xx is issued.

This parameter is valid only on z/OS.

LU62CHL
The maximum number of channels that can be current, or clients that can be connected, that use the LU 6.2 transmission protocol. If the value of LU62CHL is zero, the LU 6.2 transmission protocol is not used.

This parameter is valid only on z/OS.

MARKINT
The mark browse interval in milliseconds.

MAXCHL
The maximum number of channels that can be current (including server-connection channels with connected clients).

This parameter is valid only on z/OS.

MAXHANDS
The maximum number of open handles that any one connection can have at any one time.

Note: On Compaq NonStop Kernel, this parameter is ignored.

MAXMSGL
The maximum message length that can be handled by the queue manager. Individual queues or channels might have a smaller maximum than this.

MAXPROPL(*integer***)**
The maximum length of property data in bytes that can be associated with a message.

This parameter is valid only on i5/OS, UNIX systems, Windows, and z/OS.

MAXPRTY
The maximum priority. This is 9.

MAXUMSGS
Maximum number of uncommitted messages within one syncpoint.

Note: On Compaq NonStop Kernel, this parameter is ignored.

MONACLS
Whether online monitoring data is to be collected for auto-defined cluster-sender channels, and, if so, the rate of data collection.

MONCHL
Whether online monitoring data is to be collected for channels, and, if so, the rate of data collection.

MONQ
Whether online monitoring data is to be collected for queues, and, if so, the rate of data collection.

OPORTMAX
The maximum value in the range of port numbers to be used when binding outgoing channels.

This parameter is valid only on z/OS.

OPORTMIN
The minimum value in the range of port numbers to be used when binding outgoing channels.

This parameter is valid only on z/OS.

PARENT
The name of the queue manager to which this queue manager is connected hierarchically as its child.

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.

PSMODE
Controls whether the publish/subscribe engine and the queued publish/subscribe interface are running, and therefore controls whether applications can publish or subscribe by using the application programming interface and the queues that are monitored by the queued publish/subscribe interface.

PSNPMSG
If the queued pubsub daemon cannot process a non-persistent input message it might attempt to write the input message to the dead-letter queue (depending on the report options of the input message). If the attempt to write the input message to the dead-letter queue fails, and the MQRO_DISCARD_MSG report option was specified on the input message or PSNPMSG=DISCARD, the broker discards the input message. If PSNPMSG=KEEP is specified, the daemon will only discard the input message if the MQRO_DISCARD_MSG report option was set in the input message.

PSNPRES
If the queued pubsub daemon attempts to generate a response message in response to a non-persistent input message, and the response message cannot be delivered to the reply-to queue, this attribute indicates whether the daemon should try to write the undeliverable message to the dead-letter queue or whether to discard the message.

PSRTYCNT
When the queued pubsub daemon fails to process a command message under syncpoint (for example a publish message that cannot be delivered to a subscriber because the subscriber queue is full and it is not possible to put the publication on the dead letter queue), the unit of work is backed out and the command retried this number of times before the broker attempts to process the command message according to its report options instead.

PSSYNCPT
If this attribute is set to IFPER, when the queued pubsub daemon reads a publish or delete publication messages from a stream queue during normal operation then it specifies MQGMO_SYNCPOINT_IF_PERSISTENT. This makes the queued pubsub daemon receive non-persistent messages outside syncpoint. If the daemon receives a publication outside syncpoint, the daemon forwards that publication to subscribers known to it outside syncpoint.

When using PSSYNCPT=IFPER it is possible that a non-persistent publication might not be delivered to all subscribers known to it (for example, if an immediate queue manager shutdown occurred while a publish message was being processed). If PSSYNCPT=IFPER is specified, the daemon performance for publishing non-persistent publication improves.

QMID
The internally generated unique name of the queue manager.

This parameter is valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

QMNAME
The name of the local queue manager. See "Rules for naming WebSphere MQ objects" on page 5.

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.

RCVTIME
The approximate length of time that a TCP/IP channel waits to receive data, including heartbeats, from its partner before returning to an inactive state. This is the numeric value qualified by RCVTTYPE.

This parameter is valid only on z/OS.

RCVTMIN
The minimum length of time that a TCP/IP channel waits to receive data, including heartbeats, from its partner before returning to an inactive state.

This parameter is valid only on z/OS.

RCVTTYPE
The qualifier to apply to the value in RCVTIME.

This parameter is valid only on z/OS.

REMOTEEV
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, HP OpenVMS, HP-UX, Linux, i5/OS, 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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

ROUTEREC
Whether trace-route information is to be recorded if requested in the message.

SCHINIT
Whether the channel initiator is to be started automatically when the queue manager starts.

This parameter is valid only on AIX, Compaq NSK, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

SCMDSERV
Whether the command server is to be started automatically when the queue manager starts.

This parameter is valid only on AIX, Compaq NSK, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.

SCYCASE
Whether the security profiles are uppercase or mixed case.

This parameter is valid only on z/OS.

Note that, if this parameter has been altered but the REFRESH SECURITY command has not yet been issued, the queue manager might not be using the case of profiles you expect. Use DISPLAY SECURITY to verify which case of profiles is actually in use.

SQQMNAME
When a queue manager makes an MQOPEN call for a shared queue and the queue manager that is specified in the *ObjectQmgrName* parameter of the MQOPEN call is in the same queue-sharing group as the processing queue manager, the SQQMNAME attribute specifies whether the *ObjectQmgrName* is used or whether the processing queue manager opens the shared queue directly.

This parameter is valid only on z/OS.

SSLCRLNL
Indicates the namelist of AUTHINFO objects being used for the queue manager for CRL checking.

SSLCRYP
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 xxxxxx. This is valid only on AIX, HP-UX, Linux, Solaris, and Windows

SSLEV
Whether SSL events are generated.

SSLFIPS
Whether only FIPS-certified algorithms are to be used if cryptography is executed in WebSphere MQ itself.

SSLKEYR
Indicates the name of the Secure Sockets Layer key repository.

SSLRKEYC
Indicates the number of unencrypted bytes sent and received within an SSL conversation before the secret key is renegotiated.

SSLTASKS
On z/OS only, indicates the number of server subtasks to use for processing SSL calls.

STATACLS
Whether statistics data is to be collected for auto-defined cluster-sender channels, and, if so, the rate of data collection. This is valid only on i5/OS, UNIX systems, and Windows.

STATCHL
Whether statistics data is to be collected for channels, and, if so, the rate of data collection. This is valid only on i5/OS, UNIX systems, and Windows.

STATINT
The interval at which statistics monitoring data is written to the monitoring queue. This is valid only on i5/OS, UNIX systems, and Windows.

STATMQI
Whether statistics monitoring data is to be collected for the queue manager. This is valid only on i5/OS, UNIX systems, and Windows.

STATQ
Whether statistics data is to be collected for queues. This is valid only on i5/OS, UNIX systems, and Windows.

STRSTPEV
Whether start and stop events are generated.

SYNCPT
Whether syncpoint support is available with the queue manager.

TCPCHL
The maximum number of channels that can be current, or clients that can be connected, that use the TCP/IP transmission protocol. If zero, the TCP/IP transmission protocol is not used.

This parameter is valid only on z/OS.

TCPKEEP
Whether the KEEPALIVE facility is to be used to check that the other end of the connection is still available. If it is not available, the channel is closed.

This parameter is valid only on z/OS.

TCPNAME
The name of the TCP/IP system that you are using.

This parameter is valid only on z/OS.

TCPSTACK
Whether the channel initiator may use only the TCP/IP address space specified in TCPNAME, or may optionally bind to any selected TCP/IP address.

This parameter is valid only on z/OS.

TRAXSTR
Whether channel initiator trace should start automatically.

This parameter is valid only on z/OS.

TRAXTBL
The size, in megabytes, of the channel initiator's trace data space.

This parameter is valid only on z/OS.

TREELIFE
The lifetime of non-administrative topics.

TRIGINT
The trigger interval.

For more details of these parameters, see “ALTER QMGR” on page 44.

DISPLAY QMSTATUS

Use the MQSC command DISPLAY QMSTATUS to display status information associated with this queue manager.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	

Synonym: DIS QMSTATUS

DISPLAY QMSTATUS

Requested attrs:

Parameter descriptions for DISPLAY QMSTATUS

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 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.

CHINIT
The status of the channel initiator reading SYSTEM.DEFAULT.INITIATION.QUEUE. It is one of the following:

STOPPED
The channel initiator is not running.

STARTING
The channel initiator is in the process of initializing and is not yet operational.

RUNNING
The channel initiator is fully initialized and is running.

STOPPING
The channel initiator is stopping.

CMDSERV
The status of the command server. It is one of the following:

STOPPED
The command server is not running.

STARTING
The command server is in the process of initializing and is not yet operational.

RUNNING
The command server is fully initialized and is running.

STOPPING
The command server is stopping.

CONNS
The current number of connections to the queue manager.

CURRLOG
The name of the log extent being written to at the time that the DISPLAY QMSTATUS command is processed. If the queue manager is using circular logging, and this parameter is explicitly requested, a blank string is displayed.

MEDIALOG
The name of the oldest log extent required by the queue manager to perform media recovery. If the queue manager is using circular logging, and this parameter is explicitly requested, a blank string is displayed.

QMNAME
The name of the queue manager. This is always returned.

RECLOG
The name of the oldest log extent required by the queue manager to perform restart recovery. If the queue manager is using circular logging, and this parameter is explicitly requested, a blank string is displayed.

STATUS
The status of the queue manager. It is one of the following:

STARTING
The queue manager is in the process of initializing.

RUNNING
The queue manager is fully initialized and is running.

QUIESCING
The queue manager is quiescing.

DISPLAY QSTATUS

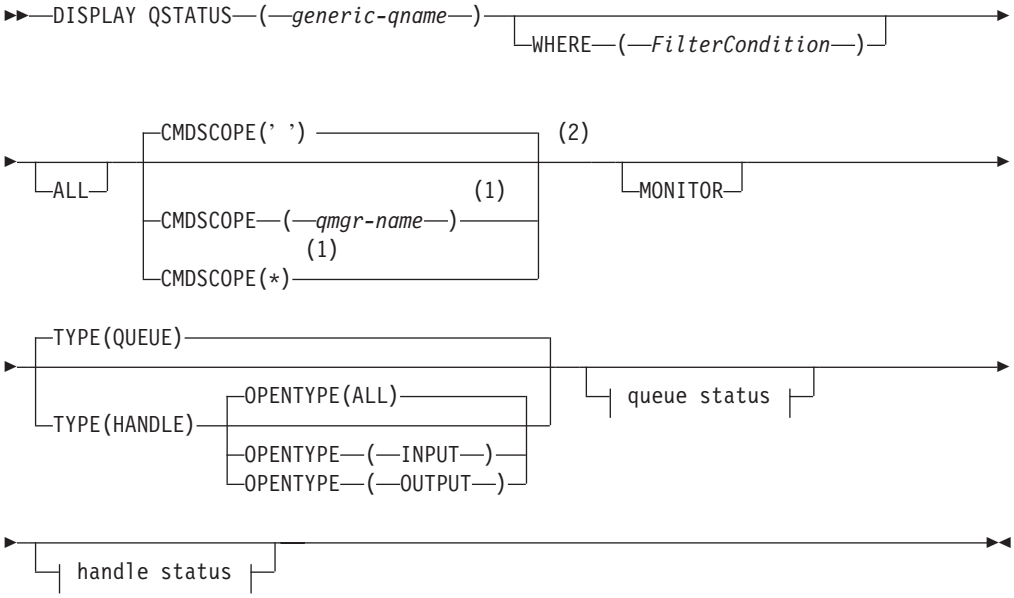
Use the MQSC command DISPLAY QSTATUS to display the status of one or more queues.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

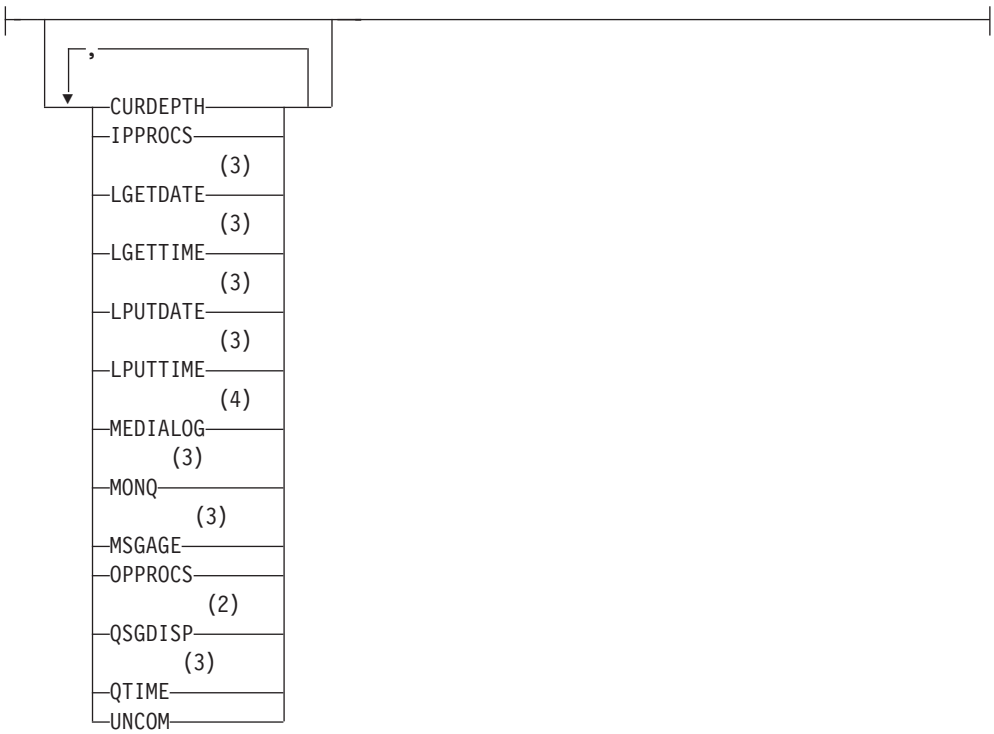
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS QS

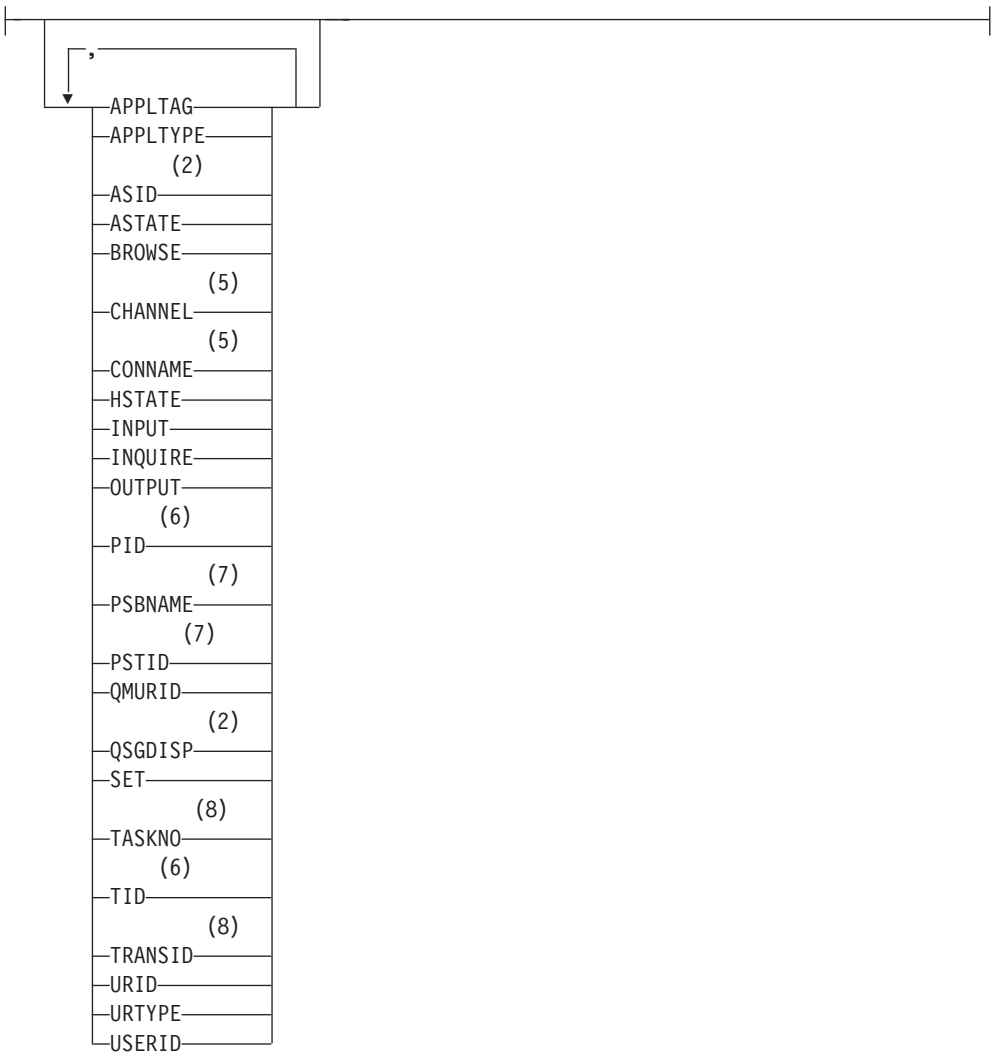
DISPLAY QSTATUS



Queue status:



Handle status:



Notes:

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

2 Valid on z/OS only.

3 Also displayed by selection of the MONITOR parameter.

4 Valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

5 Channel initiator only

6 Not valid on z/OS.

7 IMS only

8 CICS only

Parameter descriptions for DISPLAY QSTATUS

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.

WHERE
Specify a filter condition to display status information for queues that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Almost any parameter that can be used to display attributes for this DISPLAY command. However, you cannot use the CMDSCOPE, MONITOR, OPENTYPE, QSGDISP, QTIME, TYPE, or URID parameters as filter keywords.

operator
This is used to determine whether a queue satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

CT Contains a specified item. If the *filter-keyword* is a list, you can use this to display objects the attributes of which contain the specified item.

EX Does not contain a specified item. If the *filter-keyword* is a list, you can use this to display objects the attributes of which do not contain the specified item.

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:
- An explicit value, that is a valid value for the attribute being tested.

You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter (for example, the value NO on the UNCOM parameter), you can only use EQ or NE.

- A generic value. This is a character string (such as the character string in the APPLTAG parameter) with an asterisk at the end, for example ABC*. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

  You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.
- An item in a list of values. The operator must be CT or EX. If it is a character value, it can be explicit or generic. For example, if the value DEF is specified with the operator CT, all items where one of the attribute values is DEF are listed. If ABC* is specified, all items where one of the attribute values begins with ABC are listed.

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.

On z/OS this is also the default if you specify a filter condition using the WHERE parameter, but on other platforms only requested attributes are displayed.

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.

You cannot use CMDSCOPE as a filter keyword.

MONITOR

Specify this to return the set of online monitoring parameters. These are LGETDATE, LGETTIME, LPUTDATE, LPUTTIME, MONQ, MSGAGE, and QTIME. If you specify this parameter, any of the monitoring parameters that you request specifically have no effect; all monitoring parameters are still displayed.

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.

You cannot use OPENTYPE as a filter keyword.

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.

HANDLE
Status information relating to the handles that are accessing the queues is displayed.

You cannot use TYPE as a filter keyword.

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)
- On platforms other than z/OS, current queue depth (CURDEPTH parameter)
- On z/OS only, the queue-sharing group disposition (QSGDISP parameter)

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.

LGETDATE
The date on which the last message was retrieved from the queue since the queue manager started. A message being browsed does not count as a message being retrieved. When no get date is available, perhaps because no message has been retrieved from the queue since the queue manager

was started, the value is shown as a blank. For queues with QSGDISP(SHARED), the value shown is for measurements collected on this queue manager only.

This parameter is also displayed when you specify the MONITOR parameter.

A value is only displayed for this parameter if MONQ is set for this queue.

LGETTIME
The time at which the last message was retrieved from the queue since the queue manager started. A message being browsed does not count as a message being retrieved. When no get time is available, perhaps because no message has been retrieved from the queue since the queue manager was started, the value is shown as a blank. For queues with QSGDISP(SHARED), the value shown is for measurements collected on this queue manager only.

This parameter is also displayed when you specify the MONITOR parameter.

A value is only displayed for this parameter if MONQ is set for this queue.

LPUTDATE
The date on which the last message was put to the queue since the queue manager started. When no put date is available, perhaps because no message has been put to the queue since the queue manager was started, the value is shown as a blank. For queues with QSGDISP(SHARED), the value shown is for measurements collected on this queue manager only.

This parameter is also displayed when you specify the MONITOR parameter.

A value is only displayed for this parameter if MONQ is set for this queue.

LPUTTIME
The time at which the last message was put to the queue since the queue manager started. When no put time is available, perhaps because no message has been put to the queue since the queue manager was started, the value is shown as a blank. For queues with QSGDISP(SHARED), the value shown is for measurements collected on this queue manager only.

This parameter is also displayed when you specify the MONITOR parameter.

A value is only displayed for this parameter if MONQ is set for this queue.

MEDIALOG
The log extent or journal receiver needed for media recovery of the queue. On queue managers on which circular logging is in place, MEDIALOG is returned as a null string.

This is valid on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

MONQ
Current level of monitoring data collection for the queue.

This parameter is also displayed when you specify the MONITOR parameter.

MSGAGE
Age, in seconds, of the oldest message on the queue. The maximum displayable value is 999 999 999; if the age exceeds this value, 999 999 999 is displayed.

This parameter is also displayed when you specify the MONITOR parameter.

A value is only displayed for this parameter if MONQ is set for this queue.

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.

For shared queues, if the CF structure used by the queue is unavailable or has failed, the status information might be unreliable.

You cannot use QSGDISP as a filter keyword.

QTIME
Interval, in microseconds, between messages being put on the queue and then being destructively read. The maximum displayable value is 999 999 999; if the interval exceeds this value, 999 999 999 is displayed.

The interval is measured from the time that the message is placed on the queue until it is retrieved by an application and, therefore, includes any interval caused by a delay in committing by the putting application.

Two values are displayed:

- A value based on recent activity over a short period of time.
- A value based on activity over a longer period of time.

These values depend on the configuration and behavior of your system, as well as the levels of activity within it, and serve as an indicator that your system is performing normally. A significant variation in these values may indicate a problem with your system. For queues with QSGDISP(SHARED), the values shown are for measurements collected on this queue manager only.

This parameter is also displayed when you specify the MONITOR parameter.

A value is only displayed for this parameter if MONQ is set for this queue.

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)
- On platforms other than z/OS, user identifier (USERID parameter) – not returned for APPLTYPE(SYSTEM)
- On platforms other than z/OS, process ID (PID parameter)
- On platforms other than z/OS, thread ID (TID parameter)
- On platforms other than z/OS, application tag (APPLTAG parameter)
- Application type (APPLTYPE parameter)
- On platforms other than z/OS, whether handle is providing input access (INPUT parameter)
- On platforms other than z/OS, whether handle is providing output access (OUTPUT parameter)
- On platforms other than z/OS, whether handle is providing browse access (BROWSE parameter)
- On platforms other than z/OS, whether handle is providing inquire access (INQUIRE parameter)
- On platforms other than z/OS, 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
- i5/OS job name
- UNIX process
- Windows process

  Note: This consists of the full program path and executable file name. If it is more than 28 characters long, only the last 28 characters will be shown.
- Internal queue manager process name

Application name represents the name of the process or job that has connected to the queue manager. In the instance that this process or job is

connected via a channel, the application name represents the remote process or job rather than the local channel process or job name.

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

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.

ASTATE
The state of the asynchronous consumer on this queue.

Possible values are:

ACTIVE
An MQCB call has set up a function to call back to process messages asynchronously and the connection handle has been started so that asynchronous message consumption can proceed.

INACTIVE
An MQCB call has set up a function to call back to process messages asynchronously but the connection handle has not yet been started, or has been stopped or suspended, so that asynchronous message consumption cannot currently proceed.

SUSPENDED
The asynchronous consumption call back has been suspended so that asynchronous message consumption cannot currently proceed on this queue. This can be either because an MQCB call with Operation MQOP_SUSPEND has been issued against this object handle by the application, or because it has been suspended by the system. If it has been suspended by the system, as part of the process of suspending asynchronous message consumption the call back function will be called with the reason code that describes the problem resulting in suspension. This will be reported in the Reason field in the MQCBC structure that is passed to the call back function.

For asynchronous message consumption to proceed, the application must issue an MQCB call with the Operation parameter set to MQOP_RESUME.

SUSPTEMP
The asynchronous consumption call back has been temporarily suspended by the system so that asynchronous message consumption cannot currently proceed on this queue. As part of the process of suspending asynchronous message consumption, the call back function will be called with the reason code that describes the problem resulting in suspension. This will be reported in the Reason field in the MQCBC structure passed to the call back function.

The call back function will be called again when asynchronous message consumption is resumed by the system, when the temporary condition has been resolved.

NONE
An MQCB call has not been issued against this handle, so no asynchronous message consumption is configured on this handle.

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.

HSTATE
Whether an API call is in progress.

Possible values are:

ACTIVE
An API call from a connection is currently in progress for this object. For a queue, this condition can arise when an MQGET WAIT call is in progress.

If there is an MQGET SIGNAL outstanding, then this does not mean, by itself, that the handle is active.

INACTIVE
No API call from a connection is currently in progress for this object. For a queue, this condition can arise when no MQGET WAIT call is in progress.

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 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.

QMURID
The queue manager unit of recovery identifier. On z/OS, this is a 6–byte log RBA, displayed as 12 hexadecimal characters. On platforms other than z/OS, this is an 8–byte transaction identifier, displayed as `m.n` where `m` and `n` are the decimal representation of the first and last 4 bytes of the transaction identifier.

You can use QMURID as a filter keyword. On z/OS, you must specify the filter value as a hexadecimal string. On platforms other than z/OS, you must specify the filter value as a pair of decimal numbers separated by a period (.). You can only use the EQ, NE, GT, LT, GE, or LE filter operators.

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).

You cannot use QSGDISP as a filter keyword.

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 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 external unit of recovery identifier associated with the connection. It is the recovery identifier known in the external syncpoint coordinator. Its format is determined by the value of URTYPE.

You cannot use URID as a filter keyword.

URTYPE
The type of unit of recovery as seen by the queue manager. It is one of the following:
- CICS (valid only on z/OS)
- XA
- RRS (valid only on z/OS)
- IMS (valid only on z/OS)
- QMGR

URTYPE identifies the EXTURID type and not the type of the transaction coordinator. When URTYPE is QMGR, the associated identifier is in QMURID (and not URID).

USERID
The user identifier associated with the handle.

This parameter is not returned when APPLTYPE has the value SYSTEM.

Usage notes for DISPLAY QSTATUS

The state of asynchronous consumers, ASTATE, reflects that of the server-connection proxy on behalf of the client application; it does not reflect the client application state.

DISPLAY QUEUE

Use the MQSC command DISPLAY QUEUE to display the attributes of one or more queues of any type.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

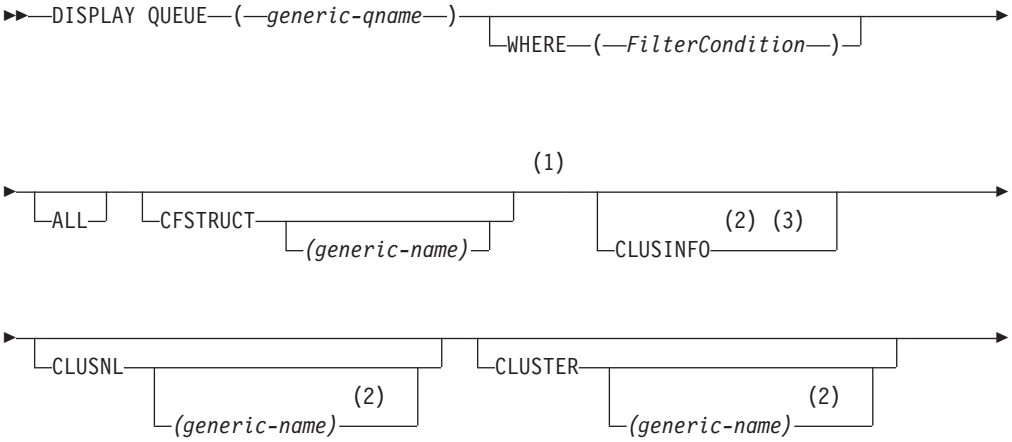
Note:

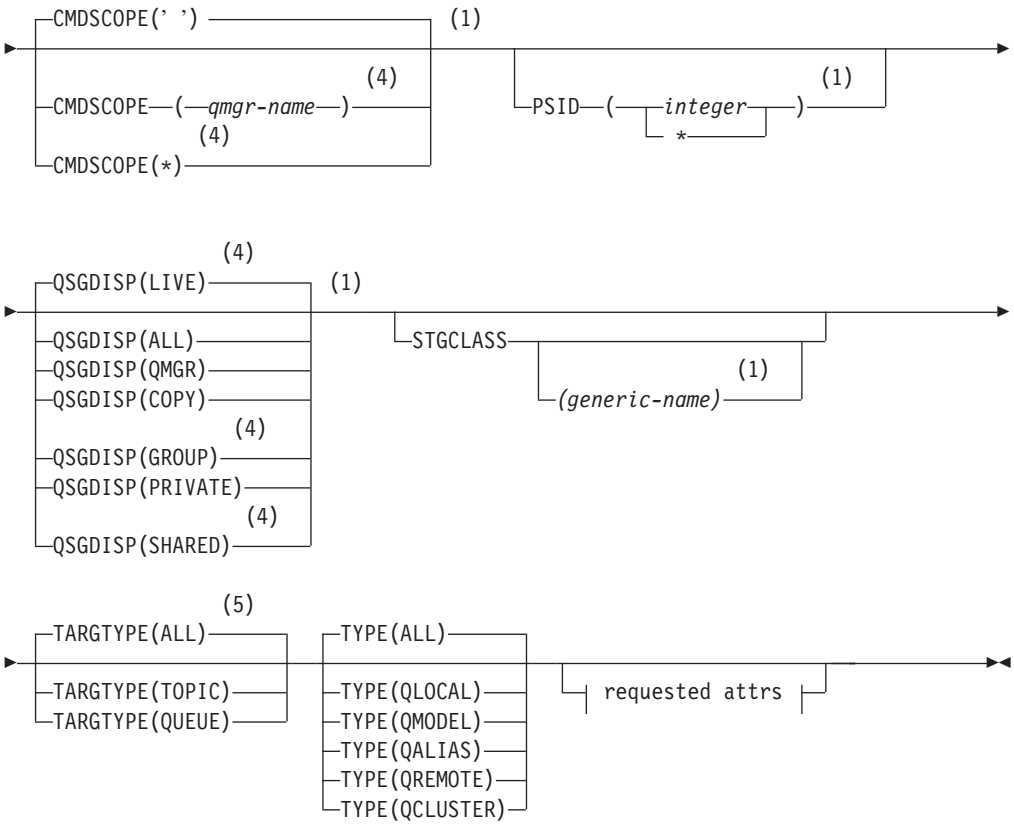
1. On AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS, you can use the following commands (or their synonyms) as an alternative way to display these attributes.
   - DISPLAY QALIAS
   - DISPLAY QCLUSTER
   - DISPLAY QLOCAL
   - DISPLAY QMODEL
   - DISPLAY QREMOTE

   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.
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.

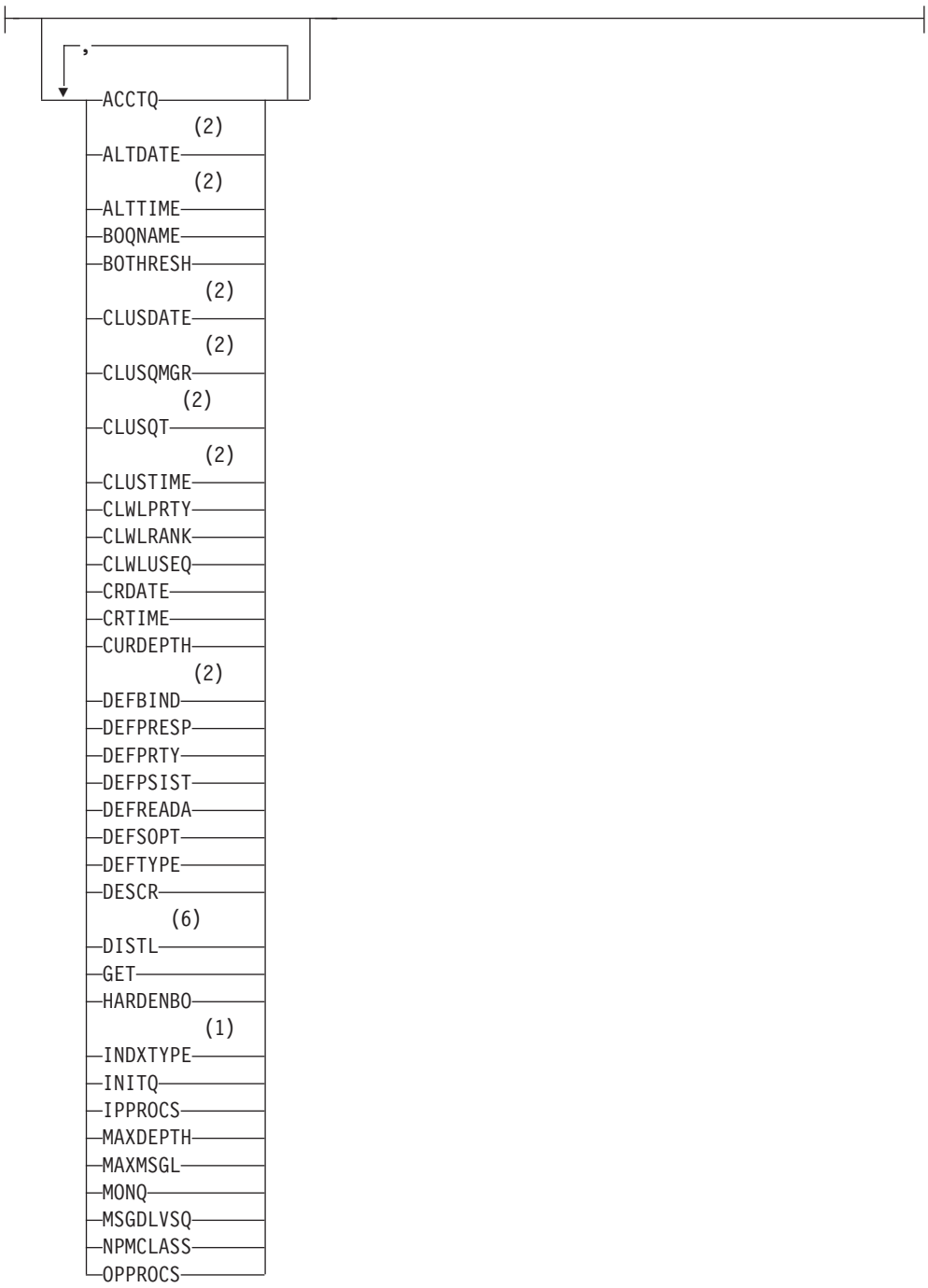
Synonym: DIS Q

DISPLAY QUEUE

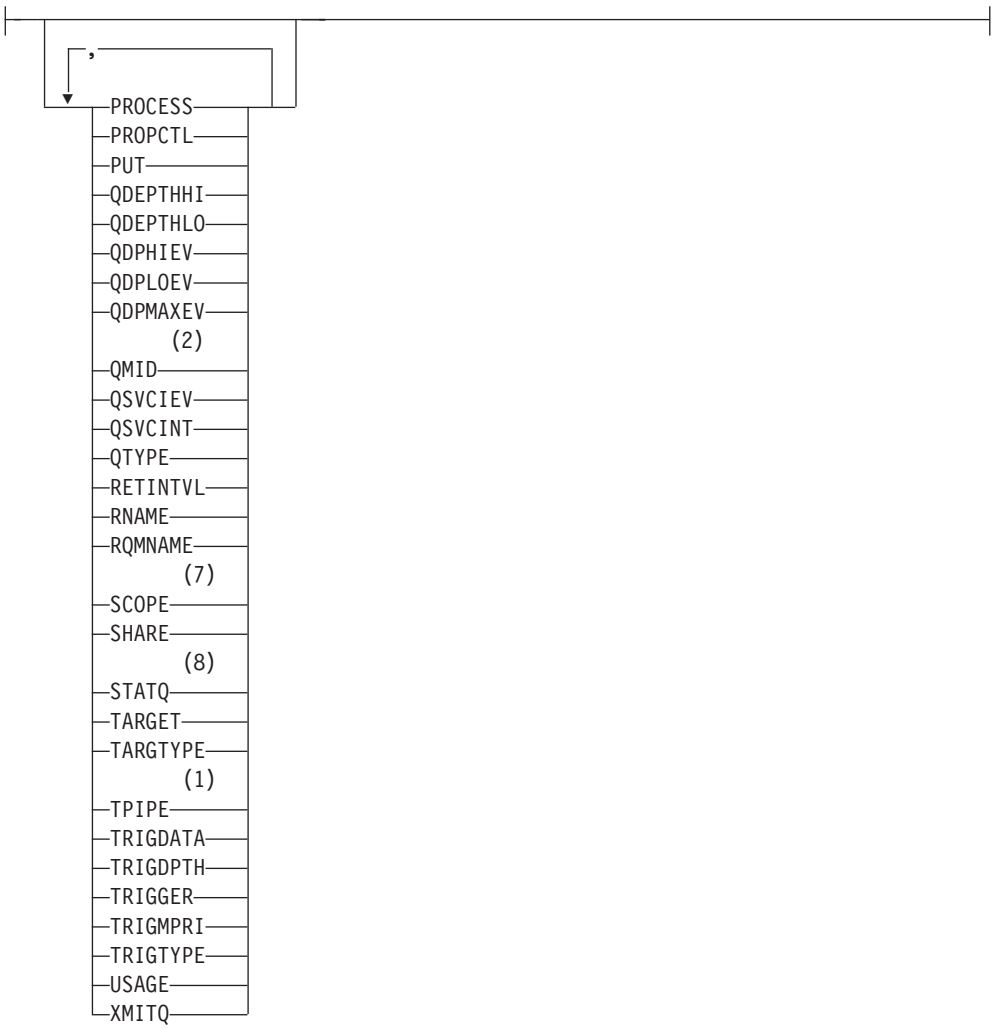




Requested attrs:



Requested attrs continued:



Notes:

1 Valid only on z/OS.

2 Valid only on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS.

3 On z/OS, you cannot issue this from CSQINP2.

4 Valid only on WebSphere MQ for z/OS when the queue manager is a member of a queue-sharing group.

5 Valid only on an alias queue.

6 Valid only on AIX, HP OpenVMS, , HP-UX, Linux, i5/OS, Solaris, and Windows.

7 Not valid on z/OS or i5/OS.

8 Valid only on AIX, HP-UX, Linux, i5/OS, Solaris, and Windows.

Parameter descriptions for DISPLAY QUEUE

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

(generic-q-name)
The local name of the queue definition to be displayed (see "Rules for naming WebSphere MQ objects" on page 5). A trailing asterisk (*) matches all queues with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all queues.

WHERE
Specify a filter condition to display only those queues that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Almost any parameter that can be used to display attributes for this DISPLAY command. However, you cannot use the CMDSCOPE, QDPHIEV, QDPLOEV, QDPMAXEV, QSGDISP, or QSVCIEV parameters as filter keywords. You cannot use CFSTRUCT, CLUSTER, CLUSNL, PSID, or STGCLASS if these are also used to select queues. Queues of a type for which the filter keyword is not a valid attribute are not displayed.

operator
This is used to determine whether a queue satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:
- An explicit value, that is a valid value for the attribute being tested.

  You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter (for example, the value QALIAS on the CLUSQT parameter), you can only use EQ or NE. For the parameters HARDENBO, SHARE, and TRIGGER, use either EQ YES or EQ NO.
- A generic value. This is a character string (such as the character string you supply for the DESCR parameter) with an asterisk at the end, for example ABC*. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

  You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.

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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS, this is the default if you do not specify a generic name and do not request any specific attributes.

On z/OS this is also the default if you specify a filter condition using the WHERE parameter, but on other platforms only requested attributes are displayed.

CFSTRUCT(*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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS. Note that, on z/OS, you cannot issue DISPLAY QUEUE CLUSINFO commands from CSQINP2.

CLUSNL(*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.
- 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, HP OpenVMS, HP-UX, Linux, i5/OS, 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, HP OpenVMS, HP-UX, Linux, i5/OS, 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.

You cannot use CMDSCOPE as a filter keyword.

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.

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:

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).

QSGDISP displays one of the following values:

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).

You cannot use QSGDISP as a filter keyword.

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.

TARGTYPE*(target-type)*
This is optional and specifies the target type of the alias queue you want to be displayed.

TYPE*(queue-type)*
This is optional, and specifies the type of queues you want to be displayed. If you specify ALL (which is the default value) all queue types are displayed; this includes cluster queues if CLUSINFO is also specified.

You can specify any, as well as ALL, of the queue types allowed for a DEFINE command (QLOCAL, QALIAS, QREMOTE, or their synonyms).

You can specify a queue type of QCLUSTER to display only cluster queue information on AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS. If QCLUSTER is specified, any selection criteria specified by the CFSTRUCT, STGCLASS, or PSID parameters are ignored. Note that you cannot issue DISPLAY QUEUE TYPE(QCLUSTER) commands from CSQINP2.

On AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows, QTYPE(*type*) can be used as a synonym for this parameter.

The queue name and queue type (and, on z/OS, the queue disposition) are always 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 13 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 13. Parameters that can be returned by the DISPLAY QUEUE command

	Local queue	Model queue	Alias queue	Remote queue	Cluster queue
ACCTQ	X	X			
ALTDATE	X	X	X	X	X
ALTTIME	X	X	X	X	X
BOQNAME	X	X			
BOTHRESH	X	X			
CFSTRUCT	X	X			
CLUSDATE					X
CLUSNL	X		X	X	
CLUSQMGR					X
CLUSQT					X
CLUSTER	X		X	X	X
CLUSTIME					X
CLWLPRTY	X		X	X	X
CLWLRANK	X		X	X	X
CLWLUSEQ	X				
CRDATE	X	X			
CRTIME	X	X			
CURDEPTH	X				
DEFBIND	X		X	X	X
DEFPRESP	X	X	X	X	X
DEFPRTY	X	X	X	X	X
DEFPSIST	X	X	X	X	X
DEFREADA	X	X	X		
DEFSOPT	X	X			
DEFTYPE	X	X			
DESCR	X	X	X	X	X
DISTL	X	X			
GET	X	X	X		
HARDENBO	X	X			
INDXTYPE	X	X			
INITQ	X	X			
IPPROCS	X				
MAXDEPTH	X	X			
MAXMSGL	X	X			
MONQ	X	X			
MSGDLVSQ	X	X			

Table 13. Parameters that can be returned by the DISPLAY QUEUE command (continued)

	Local queue	Model queue	Alias queue	Remote queue	Cluster queue
NPMCLASS	X	X			
OPPROCS	X				
PROCESS	X	X			
PROPCTL	X	X	X		
PSID	X				
PUT	X	X	X	X	X
QDEPTHHI	X	X			
QDEPTHLO	X	X			
QDPHIEV	X	X			
QDPLOEV	X	X			
QDPMAXEV	X	X			
QMID					X
QSGDISP	X	X	X	X	X
QSVCIEV	X	X			
QSVCINT	X	X			
QTYPE	X	X	X	X	X
RETINTVL	X	X			
RNAME				X	
RQMNAME				X	
SCOPE	X		X	X	
SHARE	X	X			
STATQ	X	X			
STGCLASS	X	X			
TARGET			X		
TARGTYPE			X		
TPIPE	X				
TRIGDATA	X	X			
TRIGDPTH	X	X			
TRIGGER	X	X			
TRIGMPRI	X	X			
TRIGTYPE	X	X			
USAGE	X	X			
XMITQ				X	

ACCTQ
Whether accounting (on z/OS, thread-level and queue-level accounting) data collection is to be enabled for the queue.

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.

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.

CLWLPRTY
The priority of the queue for the purposes of cluster workload distribution.

CLWLRANK
The rank of the queue for the purposes of cluster workload distribution.

CLWLUSEQ
Whether puts are allowed to other queue definitions apart from local ones.

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.

DEFPRESP
Default put response; defines the behavior that should be used by applications when the put response type in the MQPMO options has been set to MQPMO_RESPONSE_AS_Q_DEF.

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.

DEFREADA
This specifies the default read ahead behavior for non-persistent messages delivered to the client.

DEFSOPT
Default share option on a queue opened for input.

DEFTYPE
Queue definition type. This can be:

- PREDEFINED (Predefined)

  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, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, and Windows.)

GET Whether the queue is enabled for gets.

HARDENBO
Whether the back out count is hardened to ensure that the count of the number of times that a message has been backed out is accurate.

Note: This parameter affects only WebSphere MQ for z/OS. It can be set and displayed on other platforms but has no effect.

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.

MONQ
Online monitoring data collection.

MSGDLVSQ
Message delivery sequence.

NPMCLASS
Level of reliability assigned to non-persistent messages that are put to the queue.

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.

PROPCTL
Property control attribute.

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.

You cannot use QDPHIEV as a filter keyword.

QDPLOEV
Whether Queue Depth Low events are generated.

You cannot use QDPLOEV as a filter keyword.

QDPMAXEV
Whether Queue Full events are generated.

You cannot use QDPMAXEV as a filter keyword.

QMID
The internally generated unique name of the queue manager that hosts the queue.

QSVCIEV
Whether service interval events are generated.

You cannot use QSVCIEV as a filter keyword.

QSVCINT
Service interval event generation threshold.

QTYPE
Queue type.

On AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS, the queue type is always displayed.

On AIX, HP OpenVMS, HP-UX, Linux, i5/OS, 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.

STATQ
Whether statistics data information is to be collected.

STGCLASS
Storage class.

TARGET
This parameter requests that the base object name of an aliased queue is displayed.

TARGTYPE
This parameter requests that the target (base) type of an aliased queue is displayed.

TPIPE The TPIPE names used for communication with OTMA via the WebSphere MQ IMS bridge if the bridge is active. This parameter is supported only on z/OS.

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.

For more details of these parameters, see “DEFINE QUEUE” on page 229.

DISPLAY SBSTATUS

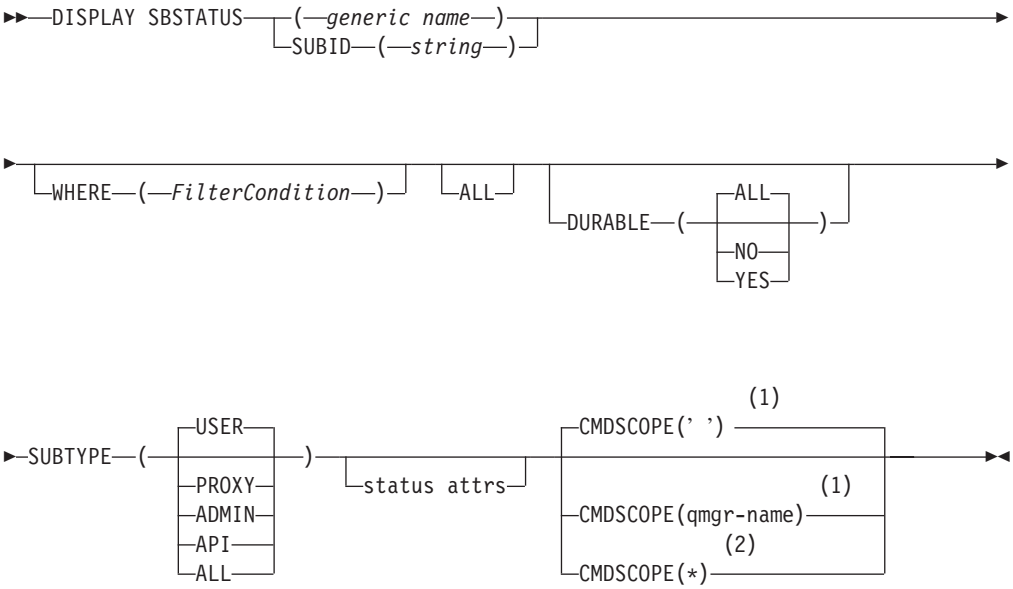
Use the MQSC command DISPLAY SBSTATUS to display the status of a subscription.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS SBSTATUS

DISPLAY SBSTATUS



Status attributes:

Notes:

1 Valid only on z/OS.

2 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for DISPLAY SBSTATUS

You must specify the name of the subscription definition for which you want to display status information. This can be a specific subscription name or a generic subscription name. By using a generic subscription name, you can display either:

- All subscription definitions
- One or more subscriptions that match the specified name

(generic-name)
The local name of the subscription definition to be displayed. A trailing asterisk (*) matches all subscriptions with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all subscriptions.

WHERE
Specify a filter condition to display only those subscriptions that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Almost any parameter that can be used to display attributes for this DISPLAY command. However, you cannot use the CMDSCOPE parameter as a filter keyword. Subscriptions of a type for which the filter keyword is not a valid attribute are not displayed.

operator
This is used to determine whether a subscription satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.

  You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter (for example, the value USER on the SUBTYPE parameter), you can only use EQ or NE.
- A generic value. This is a character string (such as the character string you supply for the SUBUSER parameter) with an asterisk at the end, for example ABC*. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.

ALL Display all the status information for each specified subscription definition. This is the default if you do not specify a generic name, and do not request any specific parameters.

On z/OS this is also the default if you specify a filter condition using the WHERE parameter, but on other platforms only, requested attributes are displayed.

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 processed on the queue manager on which it was entered. This is the default value.

qmgr-name

The command is processed 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 processed 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.

You cannot use CMDSCOPE as a filter keyword.

DURABLE

Specify this attribute to restrict the type of subscriptions which are displayed.

ALL Display all subscriptions.

NO Only information about nondurable subscriptions is displayed.

YES Only information about durable subscriptions is displayed.

SUBTYPE

Specify this attribute to restrict the type of subscriptions which are displayed.

USER Displays only API and ADMIN subscriptions.

PROXY

Only system created subscriptions relating to inter-queue-manager subscriptions are selected.

ADMIN

Only subscriptions that have been created by an administration interface or modified by an administration interface are selected.

API Only subscriptions created by applications using a WebSphere MQ API call are selected.

ALL All subscription types are displayed (no restriction).

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.

ACTCONN
Returns the *ConnId* of the *HConn* that currently has this subscription open.

DURABLE
A durable subscription is not deleted when the creating application closes its subscription handle.

NO The subscription is removed when the application that created it is closed or disconnected from the queue manager.

YES The subscription persists even when the creating application is no longer running or has been disconnected. The subscription is reinstated when the queue manager restarts.

LMSGDATE
The date on which a message was last published to the destination specified by this subscription.

LMSGTIME
The time on which a message was last published to the destination specified by this subscription.

NUMMSGS
The number of messages put to the destination specified by this subscription.

RESMDATE
The date of the most recent MQSUB API call that connected to the subscription.

RESMTIME
The time of the most recent MQSUB API call that connected to the subscription.

SUBID(*string***)**
The internal, unique key identifying a subscription.

SUBTYPE
Indicates how the subscription was created.

PROXY
An internally created subscription used for routing publications through a queue manager.

ADMIN
Created using the DEF SUB MQSC or PCF command. This SUBTYPE also indicates that a subscription has been modified using an administrative command.

API Created using an MQSUB API call.

For more details of these parameters, see “DEFINE SUB” on page 273

DISPLAY SECURITY

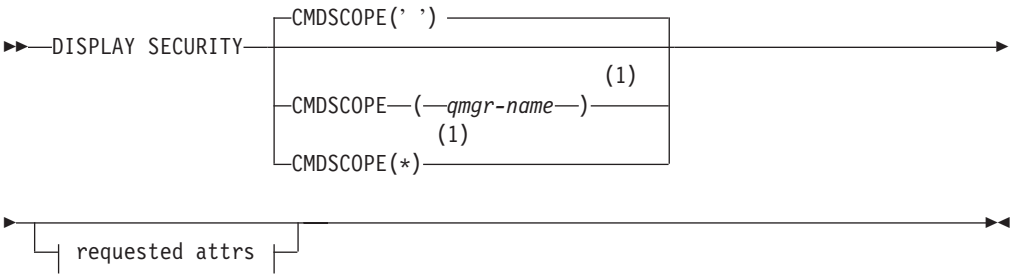
Use the MQSC command DISPLAY SECURITY to display the current settings for the security parameters.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS SEC

DISPLAY SECURITY



Requested attrs:

Notes:

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

Parameter descriptions for DISPLAY SECURITY

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.

ALL Display the TIMEOUT, INTERVAL, and SWITCHES parameters. This is the default if no requested parameters are specified.

The command now outputs an additional message, either `H037` or `H038`, stating whether security is currently using upper or mixed case security classes.

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 112 for details of the TIMEOUT and INTERVAL parameters.

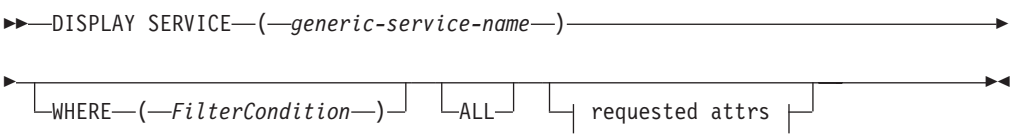
DISPLAY SERVICE

Use the MQSC command DISPLAY SERVICE to display information about a service.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	

Synonym:

DISPLAY SERVICE



Requested attrs:

Keyword and parameter descriptions for DISPLAY SERVICE

You must specify a service for which you want to display information. You can specify a service by using either a specific service name or a generic service name. By using a generic service name, you can display either:

- Information about all service definitions, by using a single asterisk (*), or
- Information about one or more service that match the specified name.

(*generic-listener-name***)**
The name of the service definition for which information is to be displayed. A single asterisk (*) specifies that information for all service identifiers is to be displayed. A character string with an asterisk at the end matches all services with the string followed by zero or more characters.

WHERE
Specify a filter condition to display information for those listeners that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Any parameter that can be used to display attributes for this DISPLAY command.

operator
This is used to determine whether a listener satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.

  You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter (for example, the value MANUAL on the CONTROL parameter), you can only use EQ or NE..
- A generic value. This is a character string. with an asterisk at the end, for example ABC*. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

  You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.

ALL Specify this to display all the service information for each specified service. 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 identifier, and do not request any specific parameters.

On z/OS this is also the default if you specify a filter condition using the WHERE parameter, but on other platforms only requested attributes are 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.

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`.

CONTROL
How the service is to be started and stopped:

MANUAL
The service is not to be started automatically or stopped automatically. It is to be controlled by use of the START SERVICE and STOP SERVICE commands.

QMGR
The service is to be started and stopped at the same time as the queue manager is started and stopped.

STARTONLY
The service is to be started at the same time as the queue manager is started, but is not requested to stop when the queue manager is stopped.

DESCR
Descriptive comment.

SERVTYPE
Specifies the mode in which the service is to run:

COMMAND
A command service object. Multiple instances of a command service object can be executed concurrently. You cannot monitor the status of command service objects.

SERVER
A server service object. Only one instance of a server service object can be executed at a time. The status of server service objects can be monitored using the DISPLAY SVSTATUS command.

STARTARG
Specifies the arguments to be passed to the user program at queue manager startup.

STARTCMD
Specifies the name of the program which is to run.

STDERR
Specifies the path to the file to which the standard error (stderr) of the service program is to be redirected.

STDOUT
Specifies the path to the file to which the standard output (stdout) of the service program is to be redirected.

STOPARG
Specifies the arguments to be passed to the stop program when instructed to stop the service.

STOPCMD
Specifies the name of the executable program to run when the service is requested to stop.

For more details of these parameters, see “DEFINE SERVICE” on page 265.

DISPLAY STGCLASS

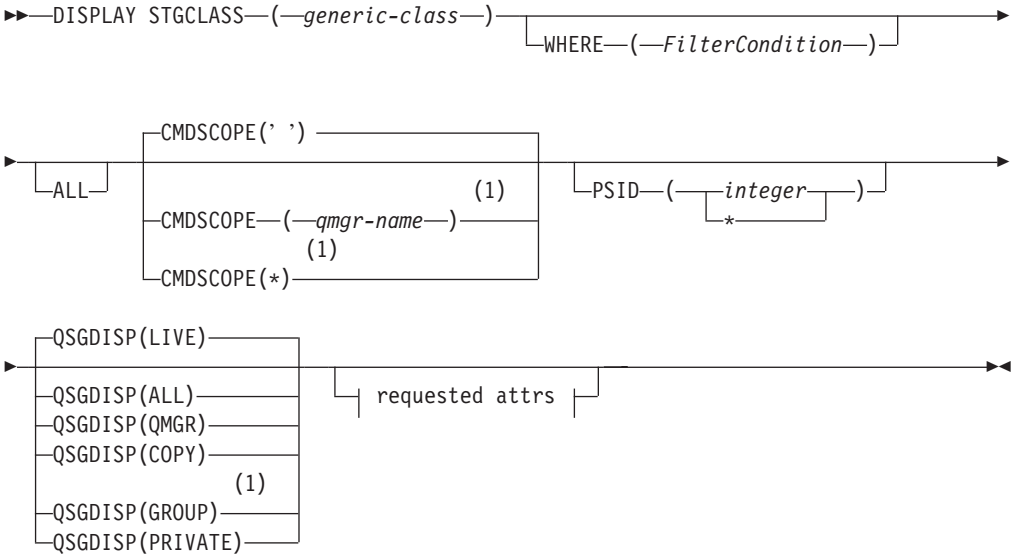
Use the MQSC command DISPLAY STGCLASS to display information about storage classes.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS STC

DISPLAY STGCLASS



Requested attrs:

Notes:

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

Parameter descriptions for DISPLAY STGCLASS

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.

WHERE
Specify a filter condition to display only those storage classes that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Almost any parameter that can be used to display attributes for this DISPLAY command. However, you cannot use the CMDSCOPE or QSGDISP parameters as filter keywords. You cannot use PSID as a filter keyword if you also use it to select storage classes.

operator
This is used to determine whether a connection satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.

  You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter, you can only use EQ or NE.
- A generic value. This is a character string (such as the character string in the DESCR parameter) with an asterisk at the end, for example ABC*. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string ABC are listed.

  You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.

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.

On z/OS this is also the default if you specify a filter condition using the WHERE parameter, but on other platforms only requested attributes are displayed.

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.

You cannot use CMDSCOPE as a filter keyword.

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 227.

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).

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:

QMGR
The object was defined with QSGDISP(QMGR).

GROUP
The object was defined with QSGDISP(GROUP).

COPY The object was defined with QSGDISP(COPY).

You cannot use QSGDISP as a filter keyword.

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, their page set identifiers and queue sharing group dispositions 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.

PASSTKTA
The application name used to authenticate IMS bridge passtickets. A blank value indicates that the default batch job profile name is to be used.

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.

For more details of these parameters, see “DEFINE STGCLASS” on page 268.

DISPLAY SUB

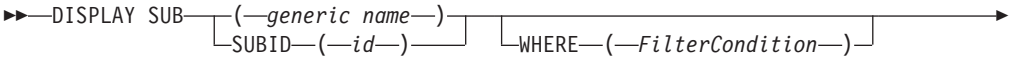
Use the MQSC command DISPLAY SUB to display the attributes associated with a subscription.

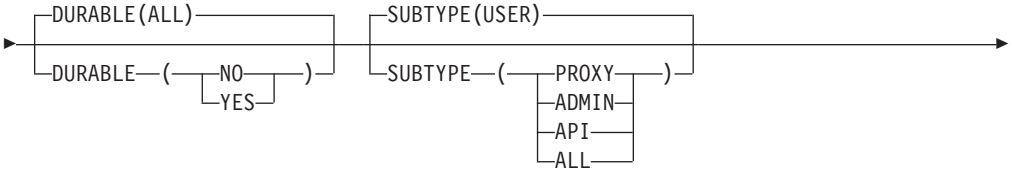
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

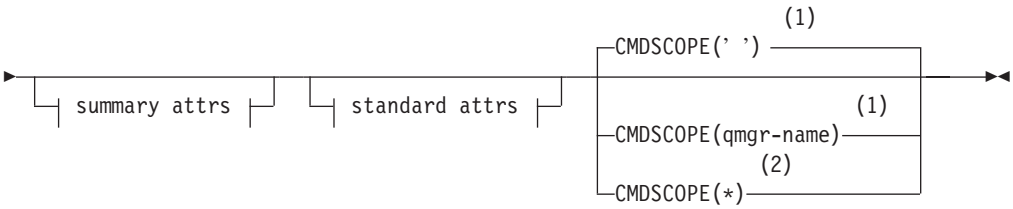
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS SUB

DISPLAY SUB

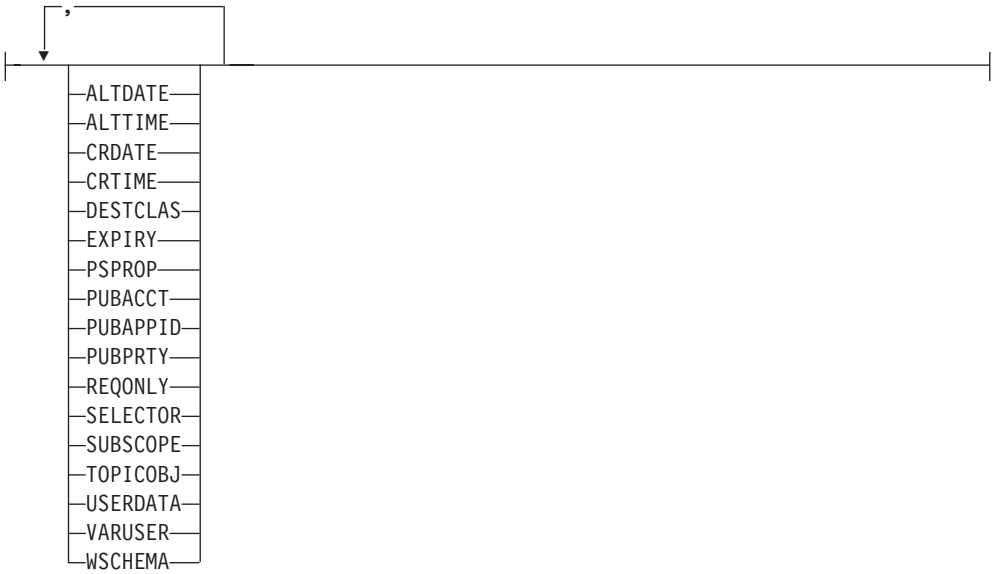






summary attributes:

standard attributes:



Notes:

1 Valid only on z/OS.

2 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for DISPLAY SUB

Describes the DISPLAY SUB (DIS SUB) parameters.

You must specify either the name or the identifier of subscription you want to display. This can be a specific subscription name, or SUBID, or a generic subscription name. By using a generic subscription name, you can display either:

- All subscription definitions
- One or more subscriptions that match the specified name

The following are valid forms:

```
DIS SUB(xyz)
DIS SUB SUBID(123)
DIS SUB(xyz*)
```

(generic-name)
The local name of the subscription definition to be displayed. A trailing asterisk (*) matches all subscriptions with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all subscriptions.

WHERE
Specify a filter condition to display only those subscriptions that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Almost any parameter that can be used to display attributes for this DISPLAY command. However, you cannot use the CMDSCOPE, QDPHIEV, QDPLOEV, QDPMAXEV, QSGDISP, or QSVCIEV parameters as filter keywords. You cannot use CFSTRUCT, CLUSTER,

CLUSNL, PSID, or STGCLASS if these are also used to select subscriptions. Subscriptions of a type for which the filter keyword is not a valid attribute are not displayed.

operator
This is used to determine whether a subscription satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.

  You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter (for example, the value QALIAS on the CLUSQT parameter), you can only use EQ or NE. For the parameters HARDENBO, SHARE, and TRIGGER, use either EQ YES or EQ NO.
- A generic value. This is a character string (such as the character string you supply for the DESCR parameter) with an asterisk at the end, for example ABC*. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

  You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.

SUMMARY
Specify this to display the set of summary attributes; this is the default value.

On AIX, HP OpenVMS, HP-UX, Linux, i5/OS, Solaris, Windows, and z/OS, this is the default if you do not specify a generic name and do not request any specific attributes.

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.

ALTDATE(*string***)**
The date of the most recent MQSUB or ALTER SUB command that modified the properties of the subscription.

ALTTIME(*string***)**
The time of the most recent MQSUB or ALTER SUB command that modified the properties of the subscription.

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

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

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

qmgr-name
The command is processed 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 processed 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.

You cannot use CMDSCOPE as a filter keyword.

CRDATE(*string***)**
The date of the first MQSUB or DEF SUB command that created this subscription.

CRTIME(*string***)**
The time of the first MQSUB or DEF SUB command that created this subscription.

DEST(*string***)**
The destination queue for messages published to this subscription.

DESTCLAS
System managed destination.

PROVIDED
The destination is a queue.

MANAGED
The destination is managed.

DESTCORL(*string***)**
The *CorrelId* used for messages published to this subscription.

DESTQMGR(*string***)**
The destination queue manager for messages published to this subscription.

DURABLE
A durable subscription is not deleted when the creating application closes its subscription handle.

ALL Display all subscriptions.

NO The subscription is removed when the application that created it, is closed or disconnected from the queue manager.

YES The subscription persists even when the creating application is no

longer running or has been disconnected. The subscription is reinstated when the queue manager restarts.

EXPIRY(*integer***)**
The time to expiry, in tenths of a second, from the creation date and time.

PSPROP
The manner in which publish subscribe related message properties are added to messages published to this subscription.

NONE
Do not add publish subscribe properties to the message.

COMPAT
Publish subscribe properties are added as PCF attributes.

MSGPROP
Publish subscribe properties are added within an RFH version 1 header.

RFH2 Publish subscribe properties are added within an RFH version 2 header.

PUBACCT(*string***)**
Accounting token passed by subscriber, for propagation into messages published to this subscription in the *AccountingToken* field of the MQMD.

PUBAPPID(*string***)**
Identity data passed by the subscriber for propagation into messages published to this subscription in the *ApplIdentityData* field of the MQMD.

PUBPRTY
The priority of the message published to this subscription.

ASPUB
Priority of the message published to this subscription is taken from that supplied in the published message.

ASQDEF
Priority of the message published to this subscription is taken from the default priority of the queue defined as a destination.

0-9 An integer providing an explicit priority for messages published to this subscription.

REQONLY
Indicates whether the subscriber polls for updates using the MQSUBRQ API call, or whether all publications are delivered to this subscription.

NO All publications on the topic are delivered to this subscription.

YES Publications are only delivered to this subscription in response to an MQSUBRQ API call.

SELECTOR(*string***)**
A selector that is applied to messages published to the topic.

SUB(*string***)**
The application's unique identifier for a subscription.

SUBID(*string***)**
The internal, unique key identifying a subscription.

SUBSCOPE
Determines whether this subscription should be forwarded to other brokers so that the subscriber receives messages published at those other brokers.

ALL The subscription is forwarded to all queue managers directly connected through a publish subscribe collective or hierarchy.

QMGR
The subscription forwards messages published only on the topic within this queue manager.

SUBTYPE
Indicates how the subscription was created.

USER Displays only API and ADMIN subscriptions.

PROXY
An internally created subscription used for routing publications through a queue manager.

ADMIN
Created using DEF SUB MQSC or PCF command. This SUBTYPE also indicates that a subscription has been modified using an administrative command.

API Created using an MQSUB API request.

ALL All.

TOPICOBJ(*string***)**
The name of a topic object used by this subscription.

TOPICSTR(*string*
Specifies a fully qualified topic name, or topic set using wildcard characters for the subscription.

USERDATA(*string***)**
Specifies the user data associated with the subscription. The string is a variable length value that can be retrieved by the application on an MQSUB API call and passed in a message sent to this subscription as a message property.

VARUSER
Specifies whether a user other than the subscription creator can connect to and take over ownership of the subscription.

ANY Any user can connect to and takeover ownership of the subscription.

FIXED
Takeover by another **USERID** is not permitted.

WSCHEMA
The schema to be used when interpreting any wildcard characters in the topic string.

CHAR
Wildcard characters represent portions of strings.

TOPIC
Wildcard characters represent portions of the topic hierarchy.

Usage notes for DISPLAY SUB

1. The TOPICSTR parameter might contain characters that cannot be translated into printable characters when the command output is displayed. On z/OS, these non-printable characters will be displayed as blanks. On distributed platforms using runmqsc, these non-printable characters will be displayed as dots.

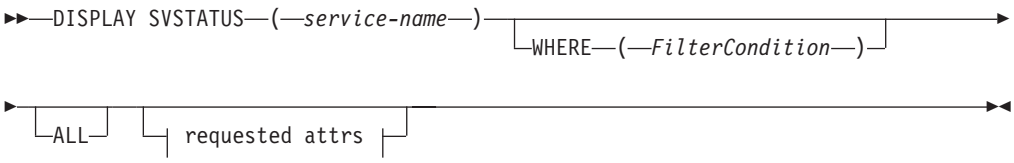
DISPLAY SVSTATUS

Use the MQSC command DISPLAY SVSTATUS to display status information for one or more services.

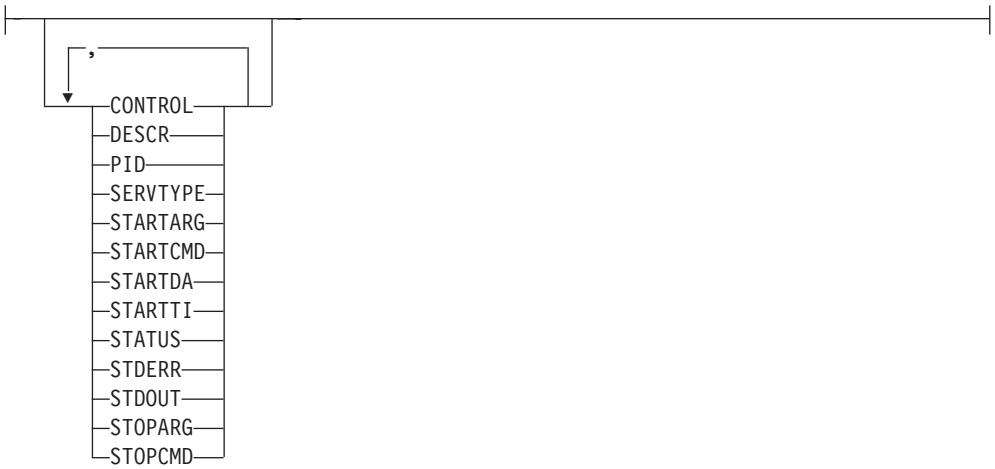
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	

Synonym:

DISPLAY SVSTATUS



Requested attrs:



Keyword and parameter descriptions for DISPLAY SVSTATUS

You must specify a service for which you want to display status information. You can specify a service by using either a specific service name or a generic service name. By using a generic service name, you can display either:

- Status information for all service definitions, by using a single asterisk (*), or
- Status information for one or more services that match the specified name.

(*generic-service-name***)**
The name of the service definition for which status information is to be displayed. A single asterisk (*) specifies that information for all connection identifiers is to be displayed. A character string with an asterisk at the end matches all services with the string followed by zero or more characters.

WHERE
Specify a filter condition to display status information for those services

that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Any parameter that can be used to display attributes for this DISPLAY command.

operator
This is used to determine whether a service satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.

  You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter (for example, the value MANUAL on the CONTROL parameter), you can only use EQ or NE.
- A generic value. This is a character string. with an asterisk at the end, for example ABC*. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

  You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.

ALL Display all the status information for each specified service. 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 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.

CONTROL
How the service is to be started and stopped:

MANUAL
The service is not to be started automatically or stopped automatically. It is to be controlled by use of the START SERVICE and STOP SERVICE commands.

QMGR
The service is to be started and stopped at the same time as the queue manager is started and stopped.

STARTONLY
The service is to be started at the same time as the queue manager is started, but is not requested to stop when the queue manager is stopped.

DESCR
Descriptive comment.

PID The operating system process identifier associated with the service.

STARTARG
The arguments passed to the user program at startup.

STARTCMD
The name of the program being run.

STARTDA
The date on which the service was started.

STARTTI
The time at which the service was started.

STATUS
The current status of the process:

RUNNING
The service is running.

STARTING
The service is in the process of initializing.

STOPPING
The service is stopping.

STDERR
Destination of the standard error (stderr) of the service program.

STDOUT
Destination of the standard output (stdout) of the service program.

STOPARG
The arguments to be passed to the stop program when instructed to stop the service.

STOPCMD
The name of the executable program to run when the service is requested to stop.

For more details of these parameters, see "DEFINE SERVICE" on page 265.

DISPLAY SYSTEM

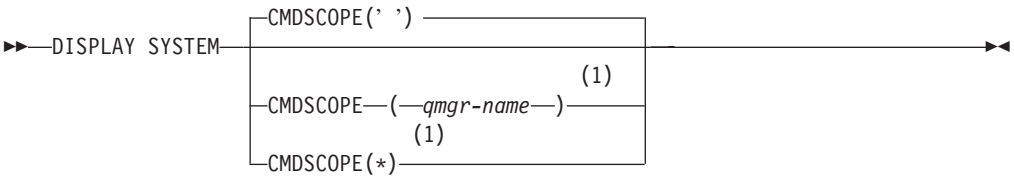
Use the MQSC command DISPLAY SYSTEM to display general system parameters and information.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS SYSTEM

DISPLAY SYSTEM



Notes:

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

Parameter descriptions for DISPLAY SYSTEM

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 for DISPLAY SYSTEM

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).
- 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 (QINDXBLD).
- 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 between scanning the queue index for WLM-managed queues (WLMTIME).
- WLMTIMU indicates whether WLMTIME is given in seconds or minutes.
- It may also return a report about system status.

2. This command is issued internally by WebSphere MQ at the end of queue manager startup.

DISPLAY THREAD

Use the MQSC command DISPLAY THREAD to display information about active and in-doubt threads.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

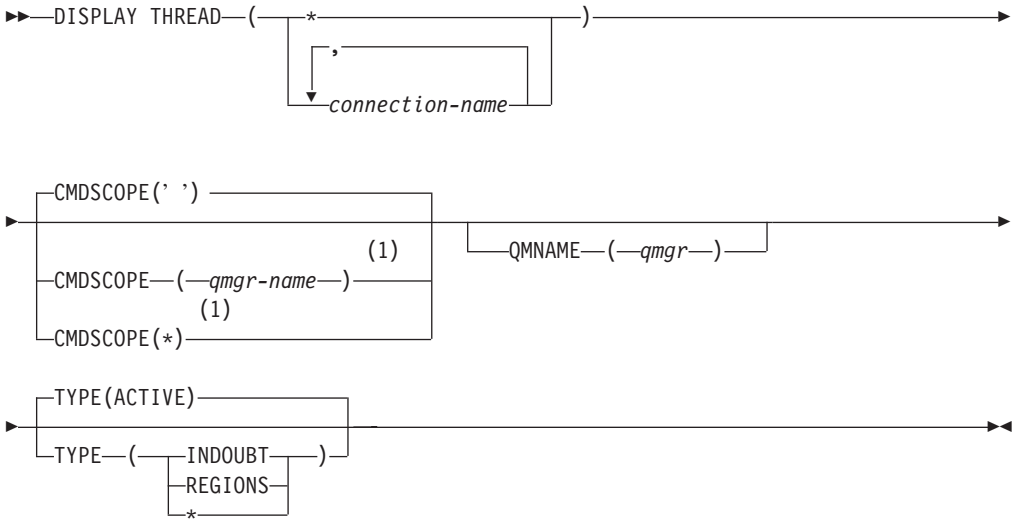
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Threads shown as in doubt on one invocation of this command will probably be resolved for subsequent invocations.

This command is retained for compatibility with earlier release of WebSphere MQ. It has been superseded by the DISPLAY CONN command which is preferable to use.

Synonym: DIS THD

DISPLAY THREAD



Notes:

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

Parameter descriptions for DISPLAY THREAD

(connection-name)

List of one or more *connection-name*s (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 for all RRSBATCH-type connections, or the batch job name

Threads are selected from the address spaces associated with these connections only.

(*) 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.

' ' 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.

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.

DISPLAY TOPIC

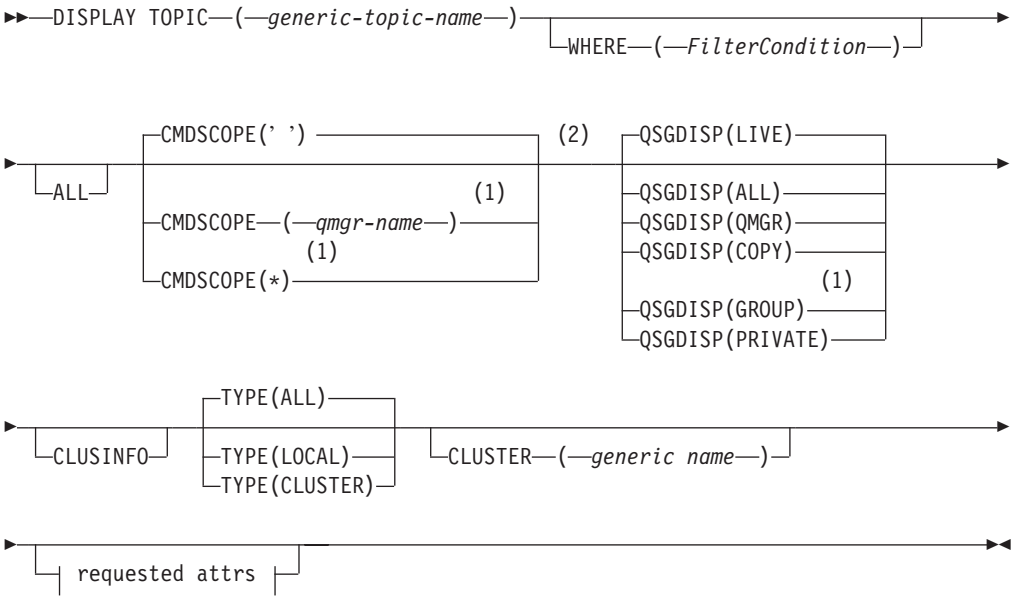
Use the MQSC command DISPLAY TOPIC to display the attributes of one or more WebSphere MQ topic objects of any type.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

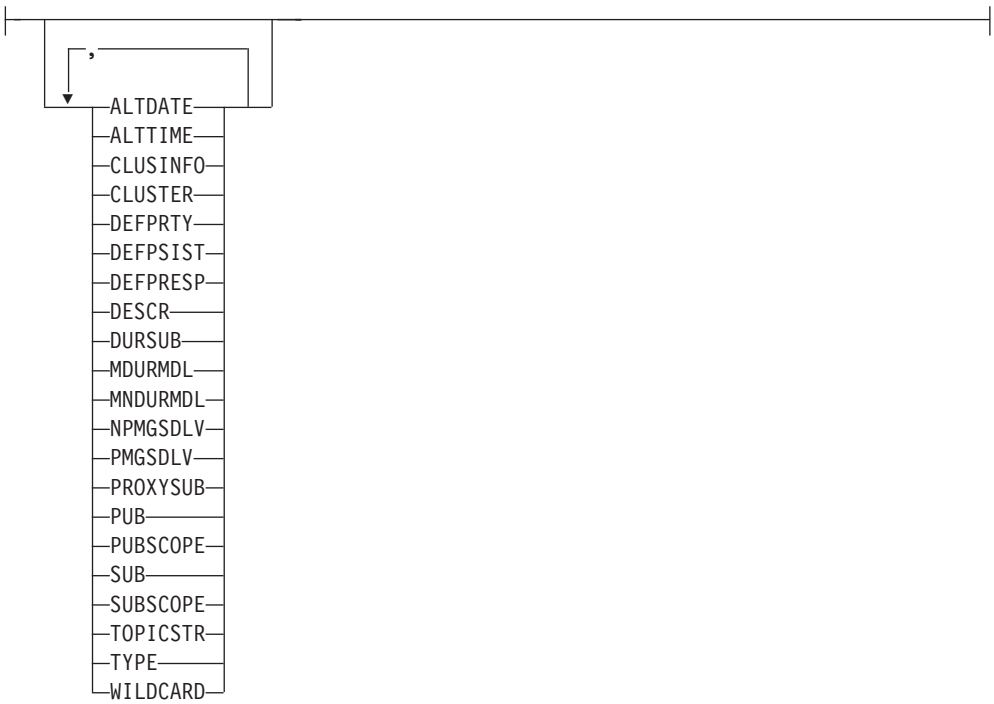
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS TOPIC

DISPLAY TOPIC



Requested attrs:



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.Valid only on i5/OS, UNIX systems, and Windows.

Parameter descriptions for DISPLAY TOPIC

You must specify the name of the topic definition you want to display. This can be a specific topic name or a generic topic name. By using a generic topic name, you can display either:

- All topic definitions
- One or more topic definitions that match the specified name

(generic-topic-name)
The name of the administrative topic definition to be displayed (see "Rules for naming WebSphere MQ objects" on page 5). A trailing asterisk (*) matches all administrative topic objects with the specified stem followed by zero or more characters. An asterisk (*) on its own specifies all administrative topic objects.

WHERE
Specify a filter condition to display only those administrative topic object definitions that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword
Almost any parameter that can be used to display attributes for this DISPLAY command. However, you cannot use the CMDSCOPE, or QSGDISP parameters as filter keywords.

operator
This is used to determine whether a topic object satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *filter-value*

NL Does not match a generic string that you provide as a *filter-value*

filter-value
The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value, that is a valid value for the attribute being tested.

  You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter, you can only use EQ or NE.
- A generic value. This is a character string (such as the character string you supply for the DESCR parameter) with an asterisk at the end, for example ABC*. If the operator is LK, all items where the attribute value begins with the string (ABC in the example) are listed. If the operator is NL, all items where the attribute value does not begin with the string are listed.

  You cannot use a generic filter-value for parameters with numeric values or with one of a set of values.

ALL Specify this to display all the attributes. If this parameter is specified, any attributes that are requested specifically have no effect; all attributes are still displayed.

This is the default if you do not specify a generic name, and do not request any specific attributes.

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.

You cannot use CMDSCOPE as a filter keyword.

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 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).

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 TOPIC(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
QSGDISP displays one of the following values:

QMGR
The object was defined with QSGDISP(QMGR).

GROUP
The object was defined with QSGDISP(GROUP).

COPY The object was defined with QSGDISP(COPY).

You cannot use QSGDISP as a filter keyword.

CLUSINFO
Requests that, in addition to information about attributes of topics defined on this queue manager, information about these and other topics in the

cluster, that match the selection criteria, is displayed. In this case, there might be multiple topics with the same topic string displayed. The cluster information is obtained from the repository on this queue manager.

On z/OS, the channel initiator must be running before you can use the CLUSINFO parameter to display information about cluster topics.

CLUSTER
Limits the information displayed to topics with the specified cluster name 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 cluster name information is returned about all the topics displayed.

On z/OS, the channel initiator must be running before you can use the CLUSINFO parameter to display information about cluster topics.

TYPE Specifies the type of topics that you want to be displayed. Values are:

ALL Display all topic types; this includes cluster topics if you also specify CLUSINFO.

LOCAL
Display locally defined topics.

CLUSTER
Display topics that are defined in publish/subscribe clusters.

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.

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`.

CLUSDATE
The date on which the information became available at the local queue manager.

CLUSQMGR
The name of the queue manager that hosts the topic.

CLUSTIME
The time at which the information became available at the local queue manager.

DEFPRTY
Default priority of the messages published to this topic.

DEFPSIST
Default persistence of messages published to this topic.

DEFPRESP
Default put response for this topic. This attribute defines the behavior that should be used by applications when the put response type in the MQPMO options has been set to MQPMO_RESPONSE_AS_TOPIC_DEF.

DESCR
Description of this administrative topic object.

DURSUB
Whether the topic permits durable subscriptions to be made.

MDURMDL
The name of the model queue for durable managed subscriptions.

MNDURMDL
The name of the model queue for non-durable managed subscriptions.

NPMSGDLV
The delivery mechanism for non-persistent messages.

PMSGDLV
The delivery mechanism for persistent messages.

PROXYSUB
Whether a proxy subscription is forced for this subscription, even if no local subscriptions exist.

PUB Whether the topic is enabled for publication.

PUBSCOPE
Whether this queue manager will propagate publications to queue managers as part of a hierarchy or as part of a publish/subscribe cluster.

QMID
The internally generated unique name of the queue manager that hosts the topic.

SUB Whether the topic is enabled for subscription.

SUBSCOPE
Whether this queue manager will propagate subscriptions to queue managers as part of a hierarchy or as part of a publish/subscribe cluster.

TOPICSTR
The topic string.

WILDCARD
The behavior of wildcard subscriptions with respect to this topic.

For more details of these parameters, see "DEFINE TOPIC" on page 278.

Usage notes for DISPLAY TOPIC

1. You can use the DISPLAY TCLUSTER command (or its synonym, DIS TCLUSTER) as an alternative way to display these attributes. This produces the same output as the DISPLAY TOPIC TYPE(CLUSTER). If you enter the command this way, do not use the TYPE parameter.
2. On z/OS, the channel initiator must be running before you can display information about cluster topics, using TYPE(CLUSTER) or the CLUSINFO parameter.
3. The TOPICSTR parameter might contain characters that cannot be translated into printable characters when the command output is displayed. On z/OS, these non-printable characters will be displayed as blanks. On distributed platforms using runmqsc, these non-printable characters will be displayed as dots.

DISPLAY TPSTATUS

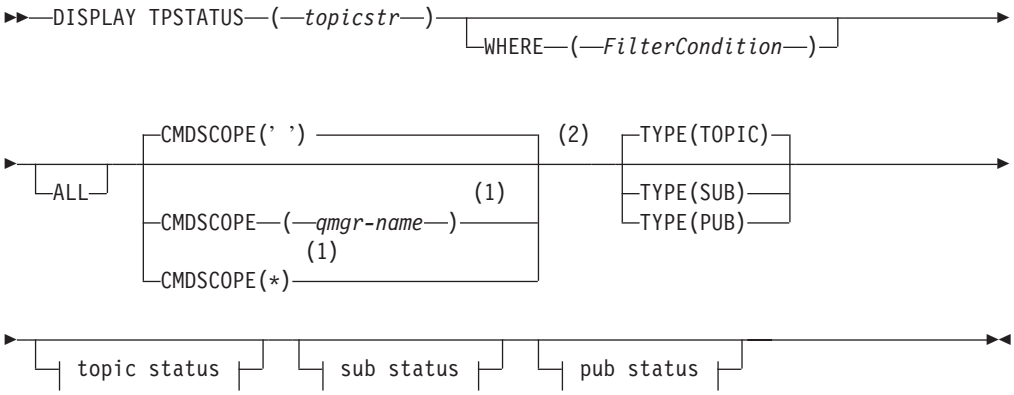
Use the MQSC command DISPLAY TPSTATUS to display the status of one or more topic nodes in a topic tree.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

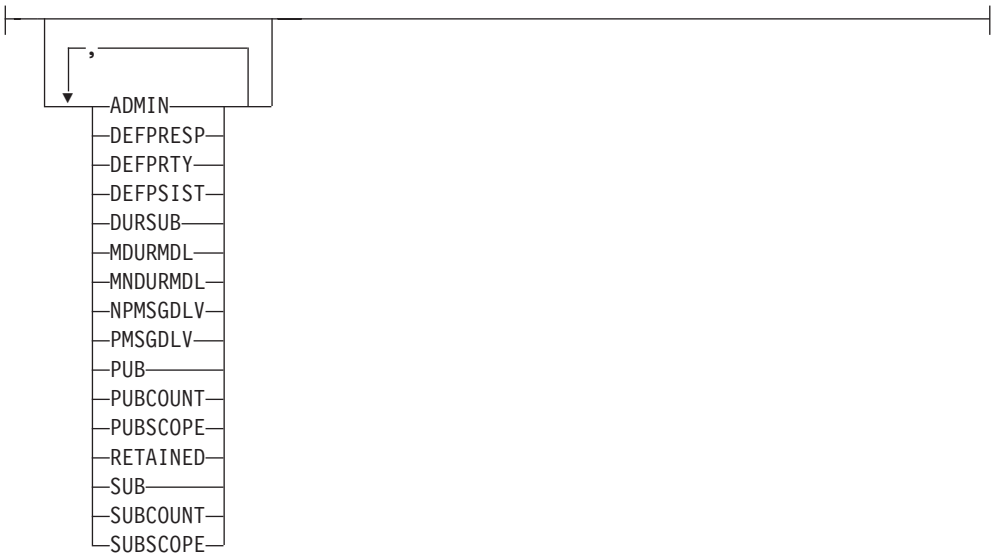
The DISPLAY TPSTATUS command requires a topic string value to determine which topic nodes the command returns.

Synonym: DIS TPS

DISPLAY TPSTATUS



Topic status:



Sub status:

Pub status:

Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for DISPLAY TPSTATUS

The DISPLAY TPSTATUS (DIS TPS) command displays the status of one or more topic nodes in a topic tree, as specified by the parameters supplied.

The DISPLAY TPSTATUS command requires a topic string value to determine which topic nodes the command returns.

(topicstr)

The value of the topic string for which you want to display status information. Note that you cannot specify the name of a WebSphere MQ topic object.

The topic string can have one of the following values:

- A specific topic string value. For example, `DIS TPS('Sports/Football')` returns just the 'Sports/Football' node.
- A topic string containing a "+" wildcard. For example, `DIS TPS('Sports/Football/+')` returns all direct child nodes of the 'Sports/Football' node.
- A topic string containing a "#" wildcard. For example, `DIS TPS('Sports/Football/#')` returns the 'Sports/Football' node and all its descendant nodes.
- A topic string containing more than one wildcard. For example, `DIS TPS('Sports/+/Teams/#')` returns any direct child node of 'Sports' that also has a 'teams' child, with all descendants of the latter nodes.

The DISPLAY TPSTATUS command does not support the '*' wildcard. For more information about using wildcards, refer to the related topic.

- To return a list of all root-level topics, use `DIS TPS('+')`
- To return a list of all topics in the topic tree, use `DIS TPS('#')`, but note that this command might return a large amount of data.
- To filter the list of topics returned, use the **WHERE** parameter. For example, `DIS TPS('Sports/Football/+') WHERE(TOPICSTR LK 'Sports/Football/L*')` returns all direct child nodes of the 'Sports/Football' node that begin with the letter "L".

WHERE

Specifies a filter condition to display only those administrative topic definitions that satisfy the selection criterion of the filter condition. The filter condition is in three parts: *filter-keyword*, *operator*, and *filter-value*:

filter-keyword

With the exception of the CMDSCOPE parameter, any parameter that you can use with this DISPLAY command.

operator

Determines whether a topic string satisfies the filter value on the given filter keyword. The operators are:

LT Less than

GT Greater than

EQ Equal to

NE Not equal to

LE Less than or equal to

GE Greater than or equal to

LK Matches a generic string that you provide as a *topicstr*

NL Does not match a generic string that you provide as a *topicstr*

filter-value

The value that the attribute value must be tested against using the operator. Depending on the filter-keyword, this can be:

- An explicit value that is a valid value for the attribute being tested.

  You can use operators LT, GT, EQ, NE, LE or GE only. However, if the attribute value is one from a possible set of values on a parameter, you can use only EQ or NE.
- A generic value. This is a character string with an asterisk at the end, for example ABC*. If the operator is LK, the command lists all topic nodes that begin with the string (ABC in the example). If the operator is NL, the command lists all topic nodes that do not begin with the string.

  You cannot use a generic *filter-value* for parameters with numeric values or with one of a set of values.

ALL Use this parameter to display all attributes.

If this parameter is specified, any attributes that you request specifically have no effect; the command displays all attributes.

This is the default parameter if you do not specify a generic name, and do not request any specific attributes.

CMDSCOPE

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

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

qmgr-name

The command runs on the named queue manager, if the queue manager is active within the queue-sharing group.

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

* The command runs on the local queue manager and on every active queue manager in the queue-sharing group. The effect of this option is equivalent to entering the command on every queue manager in the queue-sharing group.

TYPE

TOPIC

The command displays status information relating to each topic node, which is the default if you do not provide a TYPE parameter.

PUB The command displays status information relating to applications that have topic nodes open for publish.

SUB The command displays status information relating to applications that subscribe to the topic node or nodes. Note that the subscribers that the command returns are not necessarily the subscribers that would receive a message published to this topic node. The value of SelectionString or SubLevel determines which subscribers receive such messages.

Topic status parameters

Topic status parameters define the data that the command displays. You can specify these parameters in any order but must not specify the same parameter more than once.

ADMIN

If the topic node is an admin-node, the command displays the associated topic object name containing the node configuration. If the field is not an admin-node the command displays a blank.

DEFPRESP

Displays the resolved default put response of messages published to the topic, if it has no *ASPARENT* response value. The value can be *SYNC* or *ASYNC*

DEFPRTY

Displays the resolved default priority of messages published to the topic, if it has no *ASPARENT* response value.

DEFPSIST

Displays the resolved default persistence for this topic string, if it has no *ASPARENT* response value. The value can be *YES* or *NO*.

DURSUB
Displays the resolved value that shows whether applications can make durable subscriptions, if there is no *ASPARENT* response value. The value can be *YES* or *NO*.

MDURMDL
Displays the resolved value of the name of the model queue to be used for durable subscriptions. The name cannot be blank, because that is the equivalent of *ASPARENT* for this parameter.

MNDURMDL
Displays the resolved value of the name of the model queue used for non-durable subscriptions. The name cannot be blank, because that is the equivalent of *ASPARENT* for this parameter.

NPMSGDLV
Displays the resolved value for the delivery mechanism for non-persistent messages published to this topic. The value can be *ALL*, *ALLDUR*, or *ALLAVAIL*, but not ASPARENT.

PMSGDLV
Displays the resolved value for the delivery mechanism for persistent messages published to this topic. The value can be *ALL*, *ALLDUR*, or *ALLAVAIL*, but not ASPARENT.

PUB Displays the resolved value that shows whether publications are allowed for this topic, if there is no *ASPARENT* response value. The values can be *ENABLED* or *DISABLED*.

PUBCOUNT
Displays the number of handles that are open for publish on this topic node.

PUBSCOPE
Determines whether this queue manager will propagate publications, for this topic node, to queue managers as part of a hierarchy or as part of a pub/sub operation. The value can be *QMGR* or *ALL*.

RETAINED
Displays whether there is a retained publication associated with this topic. The value can be *YES* or *NO*.

SUB Displays the resolved value that shows whether subscriptions are allowed for this topic, if there is no ASPARENT response value. The values can be *ENABLED* or *DISABLED*.

SUBCOUNT
Displays the number of subscribers to this topic node, including durable subscribers that are not currently connected.

SUBSCOPE
Determines whether this queue manager will propagate subscriptions, for this topic node, to queue managers as part of a hierarchy or as part of a pub/sub operation. The value can be *QMGR* or *ALL*.

Sub status parameters

Sub status parameters define the data that the command displays. You can specify these parameters in any order but must not specify the same parameter more than once.

ACTCONN
Detects local publications, returning the currently active ConnectionId (CONNID) that opened this subscription. This parameter is returned only when the value of **ASSOC** is *YES*, indicating that. the subscription that is identified by the SUBID has an active subscriber associated with it.

DURABLE
Indicates whether a durable subscription is not deleted when the creating application closes its subscription handle, and persists over queue manager restart. The value can be *YES* or *NO*.

LMSGDATE
The date on which an MQPUT call last sent a message to this subscription. The MQPUT call updates the date field only when the call successfully puts a message to the destination specified by this subscription. Note that an MQSUBRQ call causes an update to this value.

LMSGTIME
The time at which an MQPUT call last sent a message to this subscription. The MQPUT call updates the time field only when the call successfully puts a message to the destination specified by this subscription. Note that an MQSUBRQ call causes an update to this value.

NUMMSGS
Number of messages put to the destination specified by this subscription. Note that an MQSUBRQ call causes an update to this value.

RESMDATE
Date of the most recent MQSUB call that connected to this subscription.

RESMTIME
Time of the most recent MQSUB call that connected to this subscription.

SUBID
An all time unique identifier for this subscription, assigned by the queue manager. The format of **SUBID** matches that of a CorrelId. For durable subscriptions, the command returns the **SUBID** even if the subscriber is not currently connected to the queue manager.

SUBTYPE
The type of subscription, indicating how it was created. The value can be *ADMIN*, *API*, or *PROXY*.

SUBUSER
The user ID that owns this subscription, which can be either the user ID associated with the creator of the subscription or, if subscription takeover is permitted, the user ID that last took over the subscription.

Pub status parameters

Pub status parameters define the data that the command displays. You can specify these parameters in any order but must not specify the same parameter more than once.

ACTCONN
The currently active ConnectionId (CONNID) associated with the handle that has this topic node open for publish.

LPUBDATE
The date on which this publisher last sent a message.

LPUBTIME
The time at which this publisher last sent a message.

NUMPUBS
Number of publishes by this publisher. Note that this value records the actual number of publishes, not the total number of messages published to all subscribers.

Usage notes for DISPLAY TPSTATUS

1. The TOPICSTR parameter might contain characters that cannot be translated into printable characters when the command output is displayed. On z/OS, these non-printable characters will be displayed as blanks. On distributed platforms using runmqsc, these non-printable characters will be displayed as dots.
2. The topic-string input parameter on this command must match the topic you wish to act upon. Keep the character strings in your topic strings as characters that can be used from the location issuing the command. If you issue commands using MQSC, you will have fewer characters available to you than if you are using an application that submits PCF messages, such as the WebSphere MQ Explorer.

DISPLAY TRACE

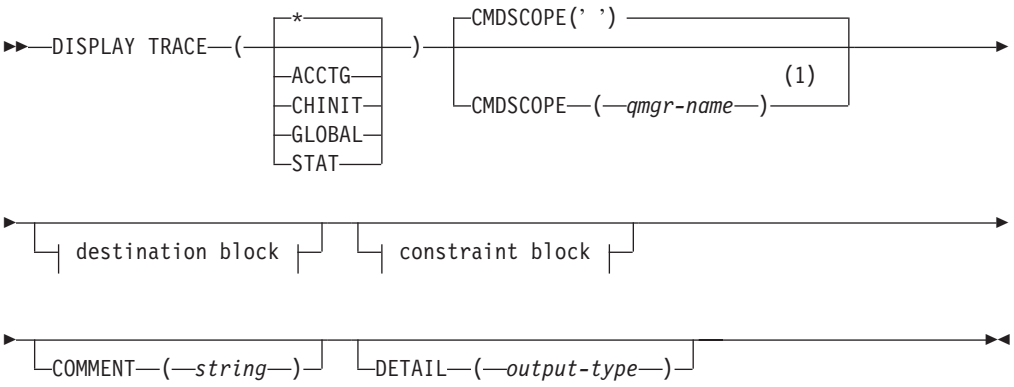
Use the MQSC command DISPLAY TRACE to display a list of active traces.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

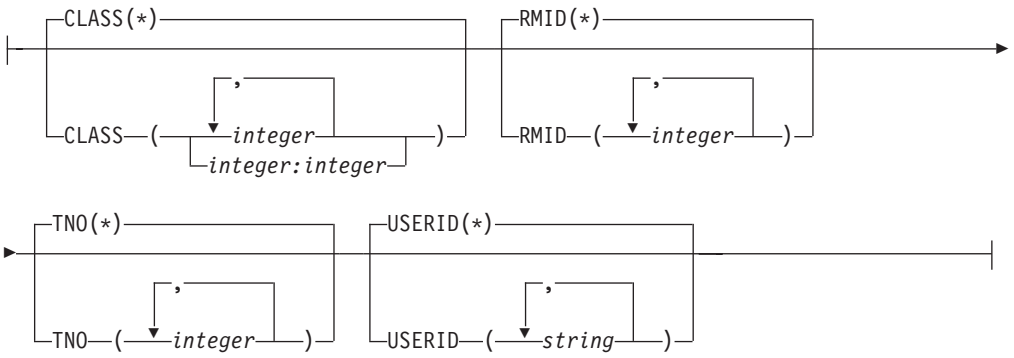
Synonym: DIS TRACE

DISPLAY TRACE



Destination block:

Constraint block:



Notes:

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

Parameter descriptions for DISPLAY TRACE

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.

* 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)

CHINIT
Service data from the channel initiator. The synonym is CHI or DQM.

GLOBAL
Service data from the entire queue manager except the channel initiator. 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***)**
This parameter is ignored; it is retained only for compatibility with earlier releases.

Possible values for *output-type* are *, 1, or 2.

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

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

Constraint block

CLASS(*integer***)**
Limits the list to traces started for particular classes. See "START TRACE" on page 573 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 573 for a list of allowed resource manager identifiers. Do not use this option with the STAT or CHINIT trace type.

The default is RMID(*), which does not limit the list.

TNO(*integer***)**
Limits the list to particular traces, identified by their trace number (0 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.

0 is the trace that the channel initiator can start automatically. Traces 1 to 32 are those for queue manager or the channel initiator that can be started automatically by the queue manager, or manually, using the START TRACE command.

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.

DISPLAY USAGE

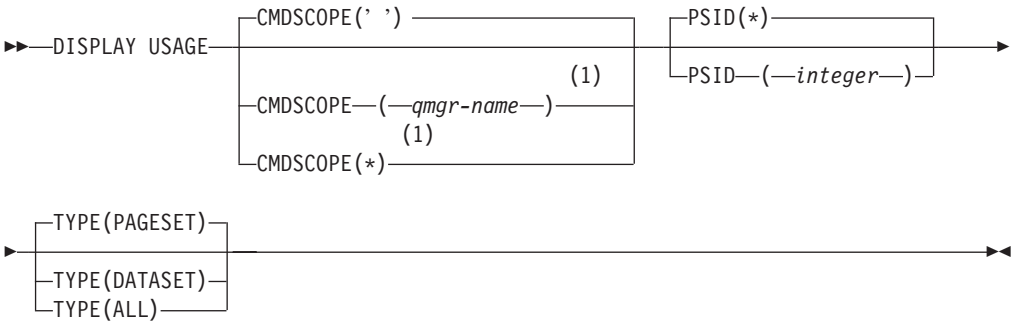
Use the MQSC command DISPLAY USAGE to display information about the current state of a page set, or to display information about the log data sets.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: DIS USAGE

DISPLAY USAGE



Notes:

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

Parameter descriptions for DISPLAY USAGE

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.

PSID(*integer*)

The page-set identifier. 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).

TYPE Defines the type of information to be displayed. Values are:

PAGESET
Display page set and buffer pool information. This is the default.

DATASET
Display data set information for log data sets. This returns messages containing 44-character data set names for the following:

- 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.
- The log data set containing the oldest restart_RBA of any pageset owned by this queue manager.
- 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.

ALL
Display both page set and data set information.

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.
- At queue manager startup so that page set information can be recorded.

MOVE QLOCAL

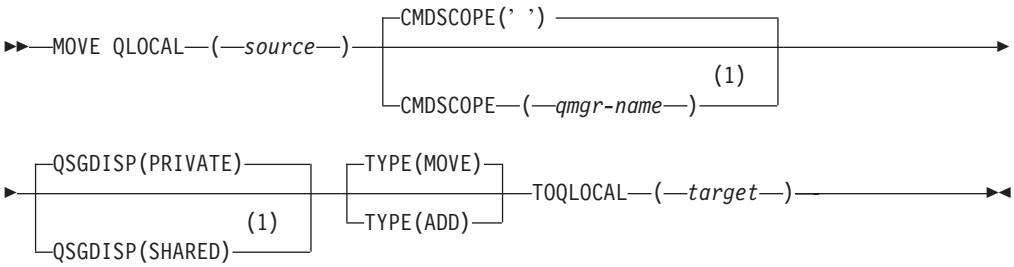
Use the MQSC command MOVE QLOCAL to move all the messages from one local queue to another.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: MOVE QL

MOVE QLOCAL



Notes:

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

Parameter descriptions for MOVE QLOCAL

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.

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, and USAGE parameters of the target queue must be the same as those of the source queue.

Usage notes for MOVE QLOCAL

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.
    - The page set is full.

PING CHANNEL

Use the MQSC command 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.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

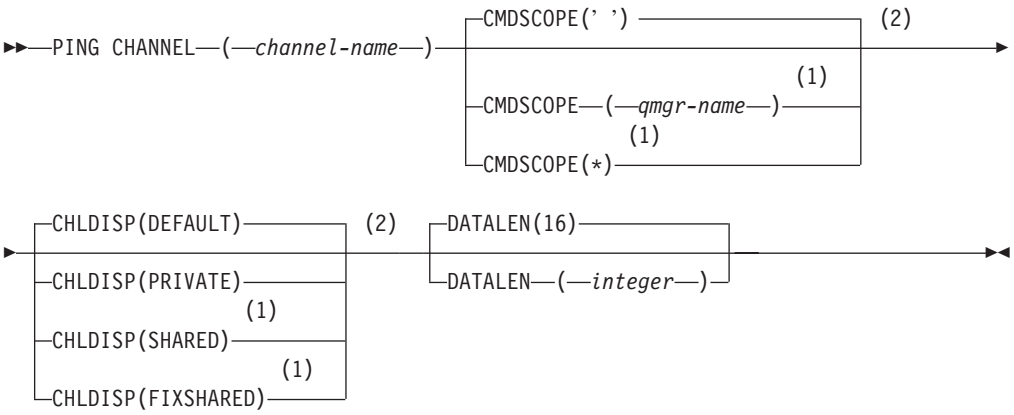
Note:

1. On z/OS, 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



Notes:

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

2 Valid only on z/OS.

Parameter descriptions for PING CHANNEL

(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:

- DEFAULT
- PRIVATE
- SHARED
- FIXSHARED

If this parameter is omitted, then the DEFAULT value applies. This is the value of the default channel disposition attribute, DEFCDISP, of the channel object.

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.

The various combinations of CHLDISP and CMDSCOPE are summarized in Table 14

Table 14. CHLDISP and CMDSCOPE for PING CHANNEL

CHLDISP	CMDSCOPE( ) or CMDSCOPE (local-qmgr)	CMDSCOPE (qmgr-name)	CMDSCOPE(*)
PRIVATE	Ping private channel on the local queue manager	Ping private channel on the named queue manager	Ping private channel on all active queue managers
SHARED	Ping a shared channel on the most suitable queue manager in the group 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.	Not permitted	Not permitted
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.

PING QMGR

Use the MQSC command PING QMGR to test whether the queue manager is responsive to commands.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	

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

RECOVER BSDS

Use the MQSC command RECOVER BSDS to reestablish a dual bootstrap data set (BSDS) after one has been disabled by a data set error.

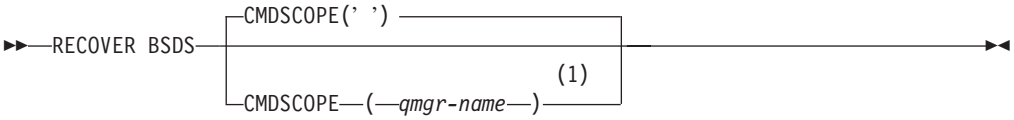
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

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



Notes:

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

Parameter descriptions for RECOVER BSDS

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.

RECOVER CFSTRUCT

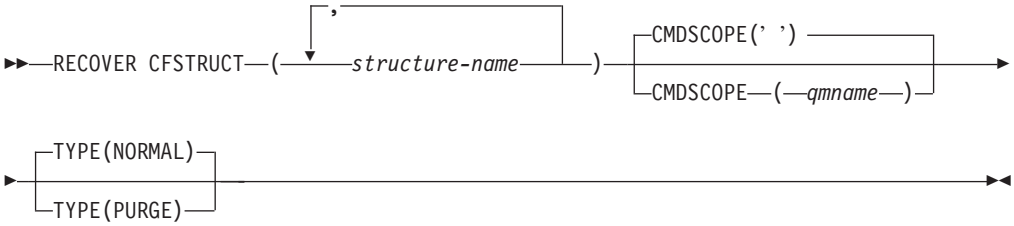
Use the MQSC command RECOVER CFSTRUCT to initiate recovery of CF application structures. This command is valid only when the queue manager is a member of a queue-sharing group.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: REC CFSTRUCT

RECOVER CFSTRUCT



Keyword and parameter descriptions for RECOVER CFSTRUCT

The keywords and parameters for RECOVER CFSTRUCT are described.

CFSTRUCT(*structure-names ...***)**
A list of names of up to 256 CF application structures to be recovered.

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.

TYPE Specifies which variant of the RECOVER command is to be issued. Values are:

NORMAL
Perform a true recovery on the CF structures from backups previously taken using the BACKUP CFSTRUCT command. This is the default.

PURGE
Recover to empty CF structures. Any messages in the CF structures are lost.

Usage notes for RECOVER CFSTRUCT

Notes on using RECOVER CFSTRUCT.

- The command fails if the application structure is not flagged as being in a FAILED state.
- The command fails if any one of the specified structure names is not defined in the CFRM policy data set.
- 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.
- 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.
- 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.
- To use TYPE(NORMAL), you must have taken a backup of the CF structures, using the BACKUP CFSTRUCT command.
- If backups of the requested CF structures have not been taken recently, using TYPE(NORMAL) may take a considerable amount of time.
- If a backup of the CF structure, or a required archive log, is not available, you can recover to an empty CF structure using TYPE(PURGE).

REFRESH CLUSTER

Use the MQSC command 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.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	CR

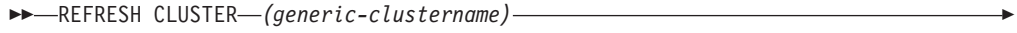
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

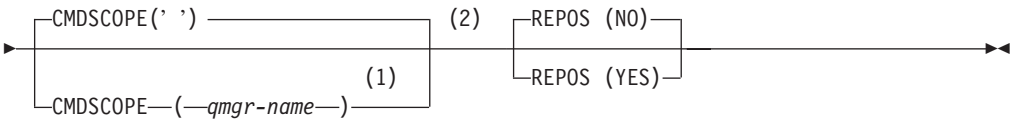
Note:

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





Notes:

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

2 Valid only on z/OS.

Parameter descriptions for REFRESH CLUSTER

(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 for REFRESH CLUSTER

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. Also, if an application is publishing or subscribing on a cluster topic, that topic might become temporarily unavailable, resulting in a pause in the publication stream until REFRESH completes. 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 a 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.

REFRESH QMGR

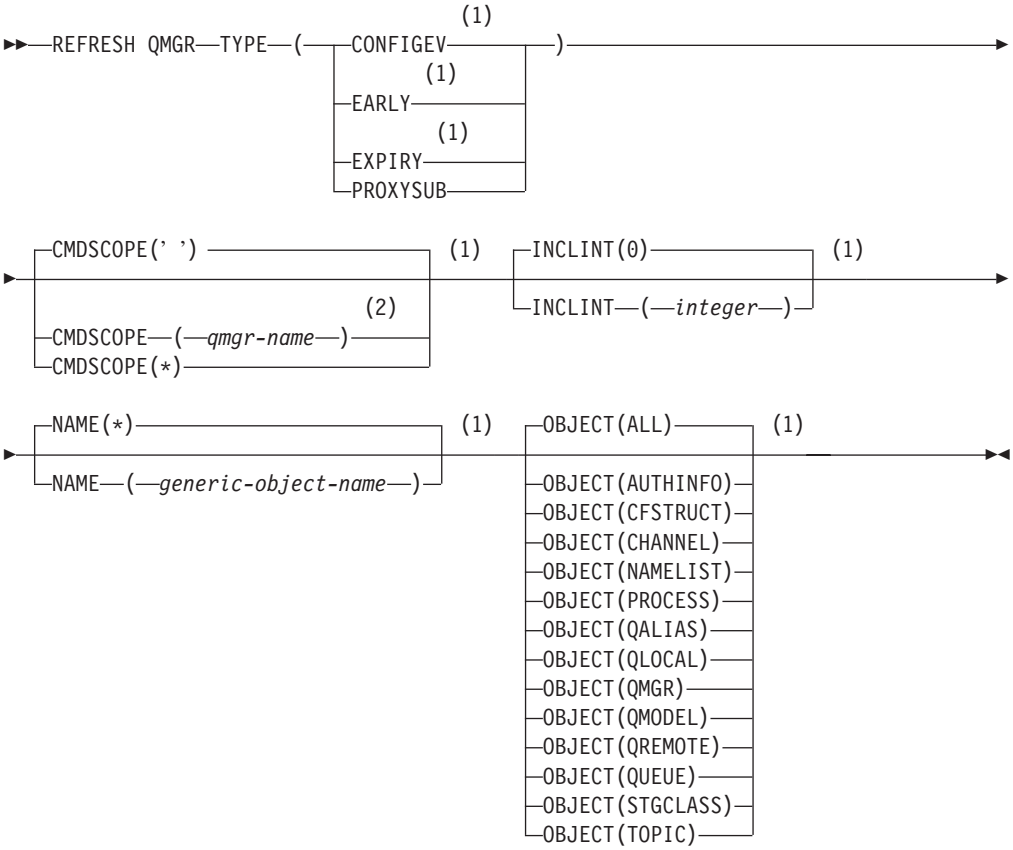
Use the MQSC command REFRESH QMGR to perform special operations on queue managers.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: None

REFRESH QMGR



Notes:

1 Valid only on z/OS.

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

Parameter descriptions for REFRESH QMGR

These are the parameter descriptions for the MQSC command REFRESH QMGR.

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.

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 need to use this command only 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 a scan to discard expired messages 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.)

PROXYSUB
Requests that the queue manager resynchronizes the proxy subscriptions that are held with, and on behalf of, queue managers that are connected in a hierarchy or publish/subscribe cluster.

Note: If TYPE(EARLY) is specified, no other keywords are allowed and the command can be issued only from the z/OS console and only if the queue manager is not active.

Usage Notes for REFRESH QMGR

1. 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.
2. 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.
3. Issue the command with TYPE(EXPIRY) at any time when you believe that a queue could contain numbers of expired messages.
4. You are unlikely to use REFRESH QMGR TYPE(PROXYSUB) other than in exceptional circumstances. Typically, a queue manager revalidates proxy subscriptions with affected directly-connected queue managers as follows:
   - When forming a hierarchical connection
   - When modifying the PUBSCOPE or SUBSCOPE or CLUSTER attributes on a topic object
   - When restarting the queue manager

REFRESH SECURITY

Use the MQSC command REFRESH SECURITY to perform a security refresh.

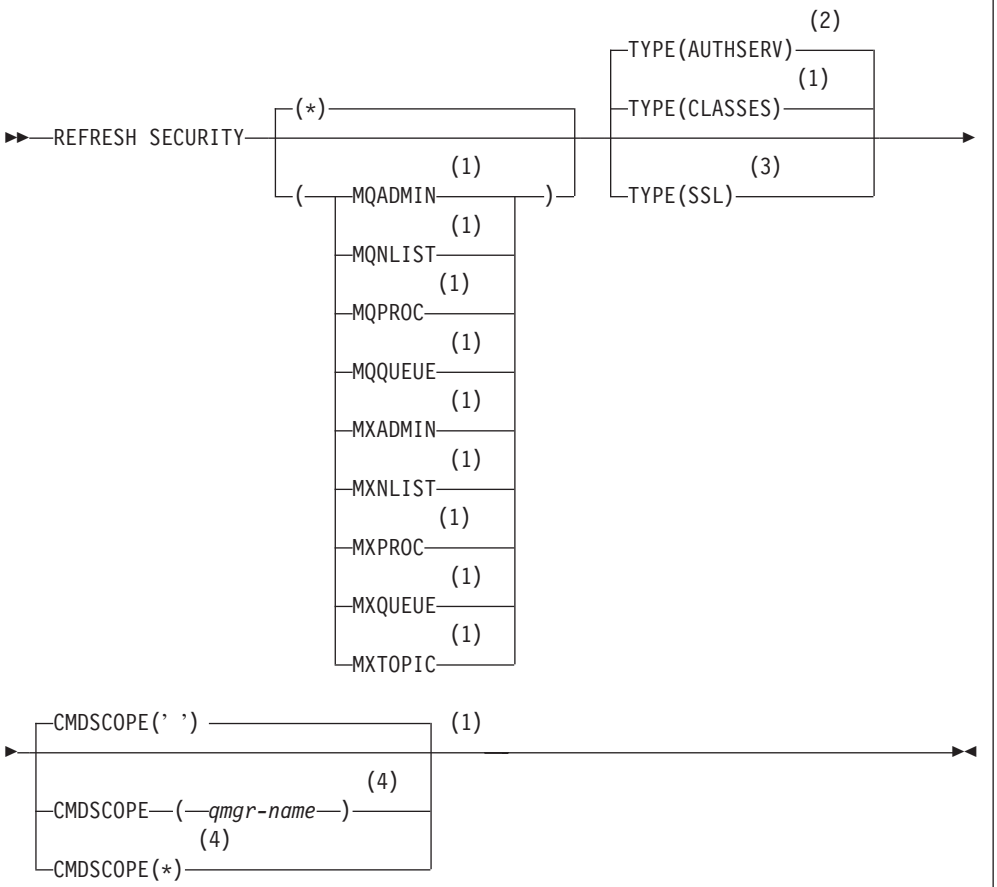
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
	X	X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: REF SEC

REBUILD SECURITY is another synonym for REFRESH SECURITY.

REFRESH SECURITY



Notes:

1 Valid only on z/OS.

2 Not valid on z/OS.

3 On WebSphere MQ for z/OS, you cannot issue this from CSQINP2.

4 Valid only on WebSphere MQ for z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for REFRESH SECURITY

The command qualifier allows you to indicate more precise behavior for a specific TYPE value. Select from:

* A full refresh of the type specified is performed. This is the default value.

MQADMIN
Valid only if TYPE is CLASSES. Specifies that Administration type resources are to be refreshed. Valid on z/OS only.

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.

MQNLIST
Valid only if TYPE is CLASSES. Specifies that Namelist resources are to be refreshed. Valid on z/OS only.

MQPROC
Valid only if TYPE is CLASSES. Specifies that Process resources are to be refreshed. Valid on z/OS only.

MQQUEUE
Valid only if TYPE is CLASSES. Specifies that Queue resources are to be refreshed. Valid on z/OS only.

MXADMIN
Valid only if TYPE is CLASSES. Specifies that administration type resources are to be refreshed. Valid on z/OS only.

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.

MXNLIST
Valid only if TYPE is CLASSES. Specifies that namelist resources are to be refreshed. Valid on z/OS only.

MXPROC
Valid only if TYPE is CLASSES. Specifies that process resources are to be refreshed. Valid on z/OS only.

MXQUEUE
Valid only if TYPE is CLASSES. Specifies that queue resources are to be refreshed. Valid on z/OS only.

MXTOPIC
Valid only if TYPE is CLASSES. Specifies that topic resources are to be refreshed. 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.

' '
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.

TYPE Specifies the type of refresh that is to be performed.

AUTHSERV
The list of authorizations held internally by the authorization services component is refreshed.

This is valid only on non-z/OS platforms where it is the default.

CLASSES

WebSphere MQ in-storage ESM (external security manager, for example RACF) profiles are refreshed. 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.

You can select specific resource classes for which to perform the security refresh.

This is valid only on z/OS where it is the default. See the WebSphere MQ for z/OS System Setup Guide for more information about RACF commands you have to issue when you use this parameter.

SSL Refreshes the cached view of the Secure Sockets Layer key repository and allows updates to become effective on successful completion of the command. Also refreshed are the locations of:

- the LDAP servers to be used for Certified Revocation Lists
- the key repository

as well as any cryptographic hardware parameters specified through WebSphere MQ.

Usage notes for REFRESH SECURITY

When you issue the REFRESH SECURITY TYPE(SSL) MQSC command, all running SSL channels are stopped and restarted. Sometimes SSL channels can take a long time to shut down and this means that the refresh operation takes some time to complete. There is a time limit of 10 minutes for an SSL refresh to complete (or 1 minute on z/OS), so it can potentially take 10 minutes for the command to finish. This can give the appearance that the refresh operation has "frozen". The refresh operation will fail with an MQSC error message of AMQ9710 or PCF error MQRCCF_COMMAND_FAILED if the timeout is exceeded before all channels have stopped. This is likely to happen if the following conditions are true:

- The queue manager has many SSL channels running simultaneously when the refresh command is invoked
- The channels are handling large numbers of messages

If a refresh fails under these conditions, retry the command later when the queue manager is less busy. In the case where many channels are running, you can choose to stop some of the channels manually before invoking the REFRESH command.

When using TYPE(SSL):

1. On z/OS, the command server and channel initiator must be running.
2. On z/OS, WebSphere MQ determines whether a refresh is needed due to one, or more, of the following reasons:
   - The contents of the key repository have changed

- The location of the LDAP server to be used for Certification Revocation Lists has changed
- The location of the key repository has changed

If no refresh is needed, the command completes successfully and the channels are unaffected.

3. On platforms other than z/OS, the command updates all SSL channels regardless of whether a security refresh is needed.
4. If a refresh is to be performed, the command updates all SSL channels currently running, as follows:
   - Sender, server and cluster-sender channels using SSL are allowed to complete the current batch. In general they then run the SSL handshake again with the refreshed view of the SSL key repository. However, you must manually restart a requester-server channel on which the server definition has no CONNAME parameter.
   - All other channel types using SSL are stopped with a STOP CHANNEL MODE(FORCE) STATUS(INACTIVE) command. If the partner end of the stopped MCA channel has retry values defined, the channel retries and the new SSL handshake uses the refreshed view of the contents of the SSL key repository, the location of the LDAP server to be used for Certification Revocation Lists, and the location of the key repository. In the case of a server-connection channel, the client application loses its connection to the queue manager and has to reconnect in order to continue.

When using TYPE(CLASSES):

- Classes MQADMIN, MQNLIST, MQPROC, and MQQUEUE can only hold profiles defined in uppercase.
- Classes MXADMIN, MXNLIST, MXPROC, and MQXUEUE can hold profiles defined in mixed case.
- Class MXTOPIC can be refreshed whether using uppercase or mixed case classes. Although it is a mixed case class, it is the only mixed case class that can be active with either group of classes.

Notes:

1. Performing a `REFRESH SECURITY(*) TYPE(CLASSES)` operation is the only way to change the classes being used by your system from uppercase-only support to mixed case support.

   Do this by checking the queue manager attribute SCYCASE to see if it is set to UPPER or MIXED
2. It is your responsibility to ensure that you have copied, or defined, all the profiles you need in the appropriate classes before you carry out a `REFRESH SECURITY(*) TYPE(CLASSES)` operation.
3. A refresh of an individual class is allowed only if the classes currently being used are of the same type. For example, if MQPROC is in use, you can issue a refresh for MQPROC but not MXPROC.

RESET CHANNEL

Use the MQSC command 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.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	CR

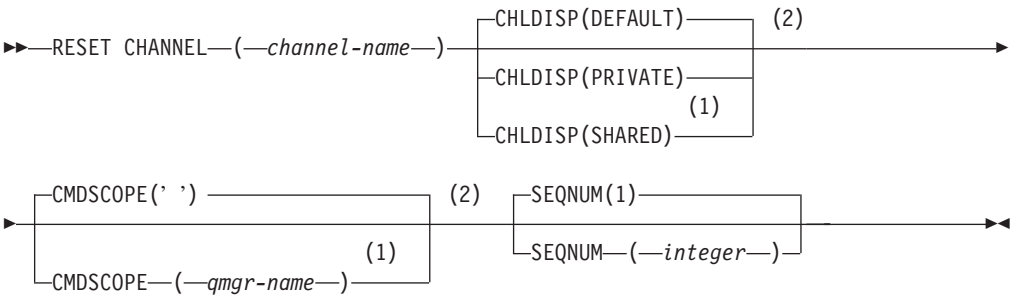
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

1. On z/OS, 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 or server channel, then in addition to resetting the value at the end at which the command is issued, the value at the other (receiver or requester) end is also reset to the same value the next time this channel is initiated (and resynchronized if necessary). Issuing this command on a cluster-sender channel may reset the message sequence number at either end of the channel. However, this is not significant because the sequence numbers are not checked on clustering channels.
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



Notes:

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

2 Valid only on z/OS.

Parameter descriptions for RESET CHANNEL

(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:

- DEFAULT
- PRIVATE
- SHARED

If this parameter is omitted, then the DEFAULT value applies. This is taken from the default channel disposition attribute, DEFCDISP, of the channel object.

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 15

Table 15. CHLDISP and CMDSCOPE for RESET CHANNEL

CHLDISP	CMDSCOPE( ) or CMDSCOPE (local-qmgr)	CMDSCOPE (qmgr-name)
PRIVATE	Reset private channel on the local queue manager	Reset private channel on the named queue manager

Table 15. CHLDISP and CMDSCOPE for RESET CHANNEL (continued)

CHLDISP	CMDSCOPE( ) or CMDSCOPE (local-qmgr)	CMDSCOPE (qmgr-name)
SHARED	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.	Not permitted

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.

RESET CLUSTER

Use the MQSC command RESET CLUSTER to perform special operations on clusters.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	CR

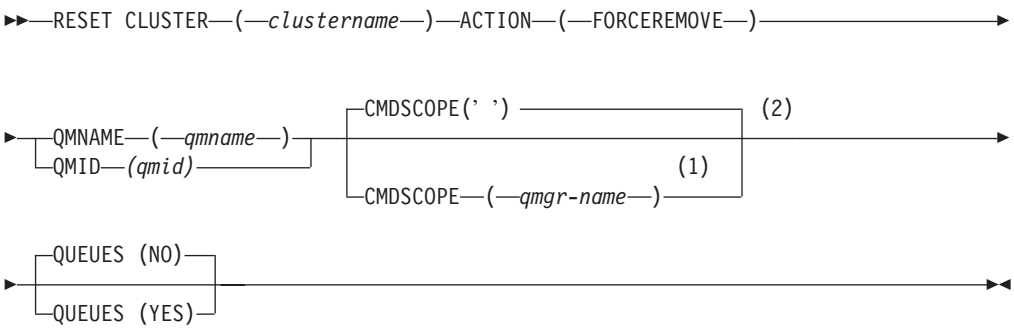
For an explanation of the symbols in the z/OS column, see"Using commands in z/OS" on page 4.

Note:

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

RESET CLUSTER



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for RESET CLUSTER

(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.

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 for RESET CLUSTER

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 5 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.

RESET QMGR

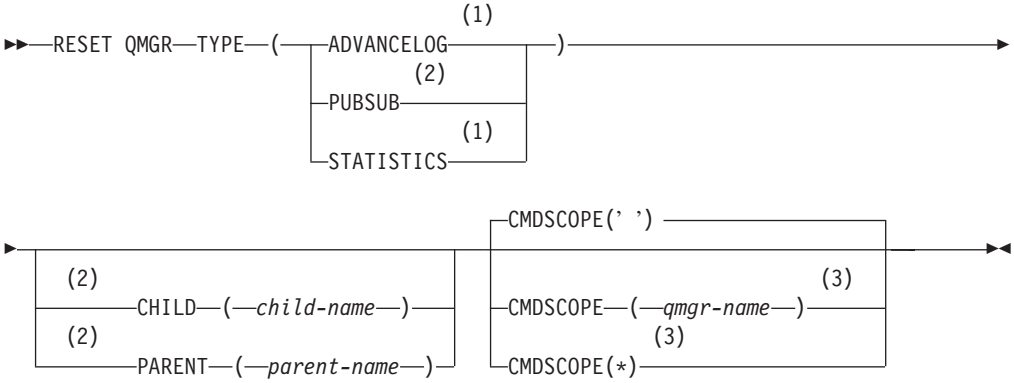
Use the MQSC command RESET QMGR as part of your backup and recovery procedures.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	2CR

You can use this command to request that the queue manager starts writing to a new log extent, making the previous log extent available for backup. See Updating a backup queue manager in the WebSphere MQ System Administration Guide. Alternatively, you can use this command to request that the queue manager ends the current statistics collection period and writes the collected statistics. You can also use this command to forcibly remove a publish/subscribe hierarchical connection for which this queue manager is nominated as either the parent or the child in the hierarchical connection.

Synonym: None

RESET QMGR



Notes:

1 Not valid on Compaq NSK, HP OpenVMS or z/OS.

2 Not valid on Compaq NSK, HP OpenVMS.

3 Valid only on z/OS when the queue manager is a member of a queue-sharing group

Parameter descriptions for RESET QMGR

TYPE

ADVANCELOG
Requests that the queue manager starts writing to a new log extent, making the previous log extent available for backup. See Updating a backup queue manager in the WebSphere MQ System Administration Guide. This command is accepted only if the queue manager is configured to use linear logging.

PUBSUB
Requests that the queue manager cancels the indicated

publish/subscribe hierarchical connection. This value requires that one of the optional CHILD or PARENT attributes is specified.

STATISTICS
Requests that the queue manager ends the current statistics collection period and writes the collected statistics.

CHILD
The name of the child queue manager for which the hierarchical connection is to be forcibly cancelled. This attribute is used only with TYPE(PUBSUB).

PARENT
The name of a parent queue manager for which the hierarchical connection is to be forcibly cancelled. This attribute is used only with TYPE(PUBSUB).

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.

Usage notes for RESET QMGR

1. The queue manager might refuse a request to advance the recovery log, if advancing the recovery log would cause the queue manager to become short of space in the active log.
2. You are unlikely to use RESET QMGR TYPE(PUBSUB) other than in exceptional circumstances. Typically the child queue manager uses ALTER QMGR PARENT(' ') to remove the hierarchical connection.

   When you need to disconnect from a child or parent queue manager with which the queue manager has become unable to communicate, you must issue the RESET QMGR TYPE (PUBSUB) command from a queue manager. When using this command, the remote queue manager is not informed of the cancelled connection. It might, therefore, be necessary to issue the ALTER QMGR PARENT(' ') command at the remote queue manager. If the child queue manager is not manually disconnected, it is forcibly disconnected and the parent status is set to REFUSED.

   If you are resetting the parent relationship, you should issue the ALTER QMGR PARENT(' ') command, otherwise the queue manager will attempt to re-establish the connection when the publish/subscribe capability of the queue manager is subsequently enabled.

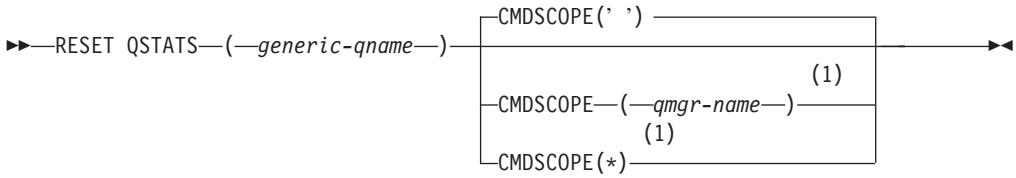
RESET QSTATS

Use the MQSC command RESET QSTATS to report performance data for a queue and then to reset that data.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: None



Notes:

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

Parameter descriptions for RESET QSTATS

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 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. The maximum displayable value is 999 999 999; if the number exceeds this value, 999 999 999 is displayed.

MSGSOUT
The number of messages removed from the queue by destructive (non-browse) MQGET calls since the statistics were last reset.

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. The maximum displayable value is 999 999 999; if the number exceeds this value, 999 999 999 is displayed.

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 for RESET QSTATS

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.

RESET TPIPE

Use the MQSC command RESET TPIPE to reset the recoverable sequence numbers for an IMS Tpipe used by the WebSphere MQ-IMS bridge.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					CR

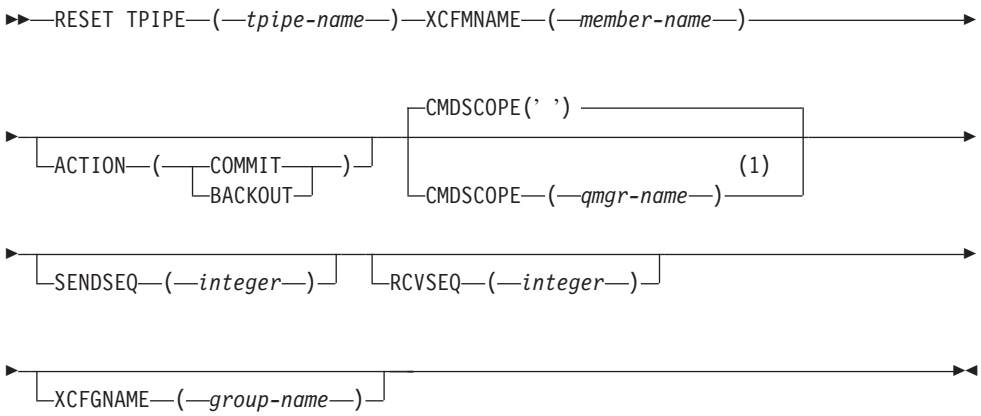
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

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.

Synonym: There is no synonym for this command.

RESET TPIPE



Notes:

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

Parameter descriptions for RESET TPIPE

(*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.

RESOLVE CHANNEL

Use the MQSC command RESOLVE CHANNEL to request a channel to commit or back out in-doubt messages.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	CR

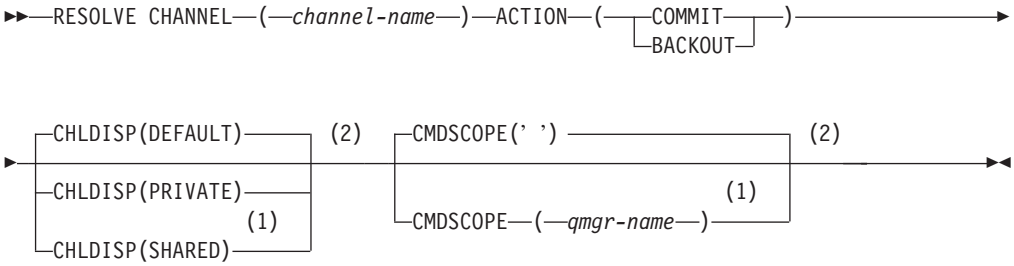
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

1. On z/OS, 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)

RESOLVE CHANNEL



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for RESOLVE CHANNEL

(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):

COMMIT
The messages are committed, that is, they are deleted from the transmission queue

BACKOUT
The messages are backed out, that is, they are restored to the transmission queue

CHLDISP
This parameter applies to z/OS only and can take the values of:

- DEFAULT
- PRIVATE
- SHARED

If this parameter is omitted, then the DEFAULT value applies. This is taken from the default channel disposition attribute, DEFCDISP, of the channel object.

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 16

Table 16. CHLDISP and CMDSCOPE for RESOLVE CHANNEL

CHLDISP	CMDSCOPE( ) or CMDSCOPE (local-qmgr)	CMDSCOPE (qmgr-name)
PRIVATE	Resolve private channel on the local queue manager	Resolve private channel on the named queue manager

Table 16. CHLDISP and CMDSCOPE for RESOLVE CHANNEL (continued)

CHLDISP	CMDSCOPE( ) or CMDSCOPE (local-qmgr)	CMDSCOPE (qmgr-name)
SHARED	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.	Not permitted

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 for RESOLVE CHANNEL

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.

RESOLVE INDOUBT

Use the MQSC command RESOLVE INDOUBT to resolve threads left in doubt because WebSphere MQ or a transaction manager could not resolve them automatically.

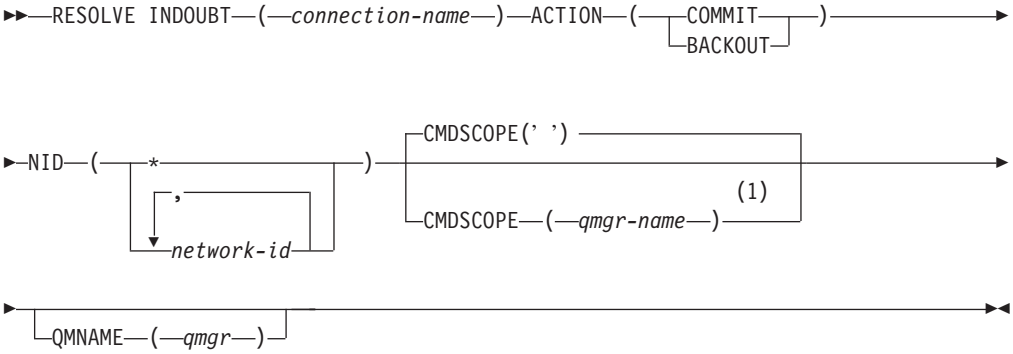
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

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



Notes:

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

Parameter descriptions for RESOLVE INDOUBT

(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.

' '
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.

NID Origin identifier. Specifies the thread or threads to be resolved.

(*origin-id***)**
This is as returned by the DISPLAY CONN command, and is of the form *origin-node*.*origin-urid*, where:

- *origin-node* identifies the originator of the thread, except RRSBATCH where it is omitted.
- *origin-urid* is the hexadecimal number assigned to the unit of recovery by the originating system for the specific thread to be resolved.

When *origin-node* is present there must be a period (.) between it and *origin-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 origin 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(*)
```

RESUME QMGR

Use the MQSC command RESUME QMGR to inform other queue mangers 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.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	CR

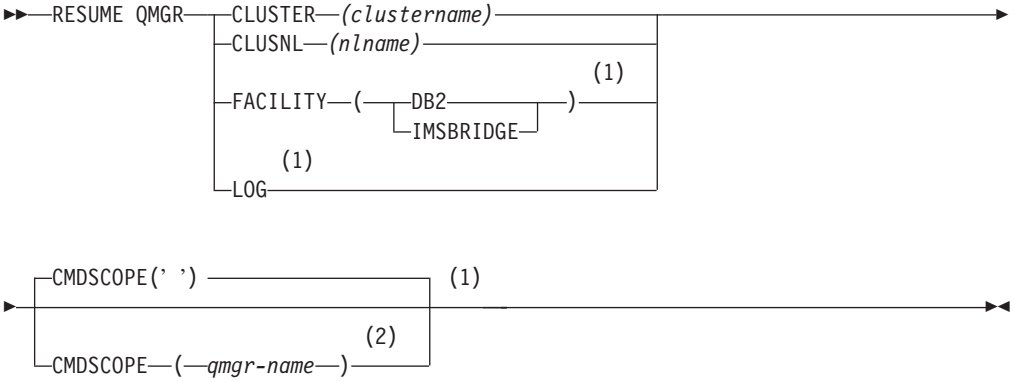
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

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:
   a. The command fails if the channel initiator has not been started.
   b. 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.
3. On z/OS, you cannot issue RESUME QMGR CLUSTER*(clustername)* or RESUME QMGR FACILITY commands from CSQINP2.
4. This command, with the CLUSTER and CLUSNL parameters, is **not** available on the reduced function form of WebSphere MQ for z/OS supplied with WebSphere Application Server.

Synonym: None

RESUME QMGR



Notes:

1 Valid only on z/OS.

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

Parameter descriptions for RESUME QMGR

CLUSTER*(clustername)*
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.

FACILITY
Specifies the facility to which connection is to be re-established.

DB2 Re-establishes connection to DB2.

IMSBRIDGE
Resumes normal IMS Bridge activity.

This parameter is only valid on z/OS.

LOG Resumes logging and update activity for the queue manager that was suspended by a previous SUSPEND QMGR command. Valid on z/OS only. If LOG is specified, the command can be issued only from the z/OS console.

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.

RVERIFY SECURITY

Use the MQSC command 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.

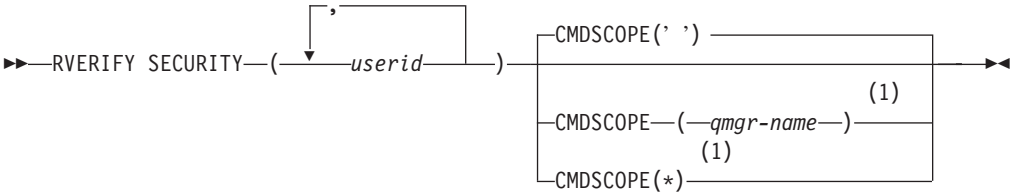
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: REV SEC

Note: REVERIFY SECURITY is another synonym for RVERIFY SECURITY.

RVERIFY SECURITY



Notes:

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

Parameter descriptions for RVERIFY SECURITY

(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.

SET ARCHIVE

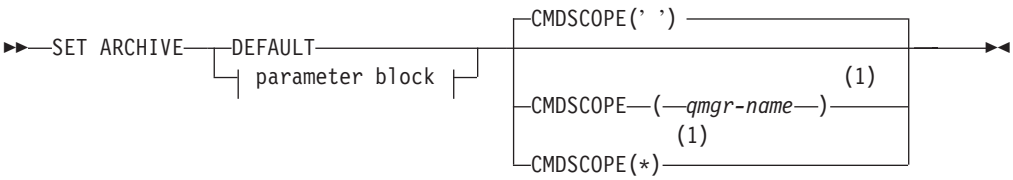
Use the MQSC command SET ARCHIVE to dynamically change certain archive system parameter values initially set by your system parameter module at queue manager startup.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

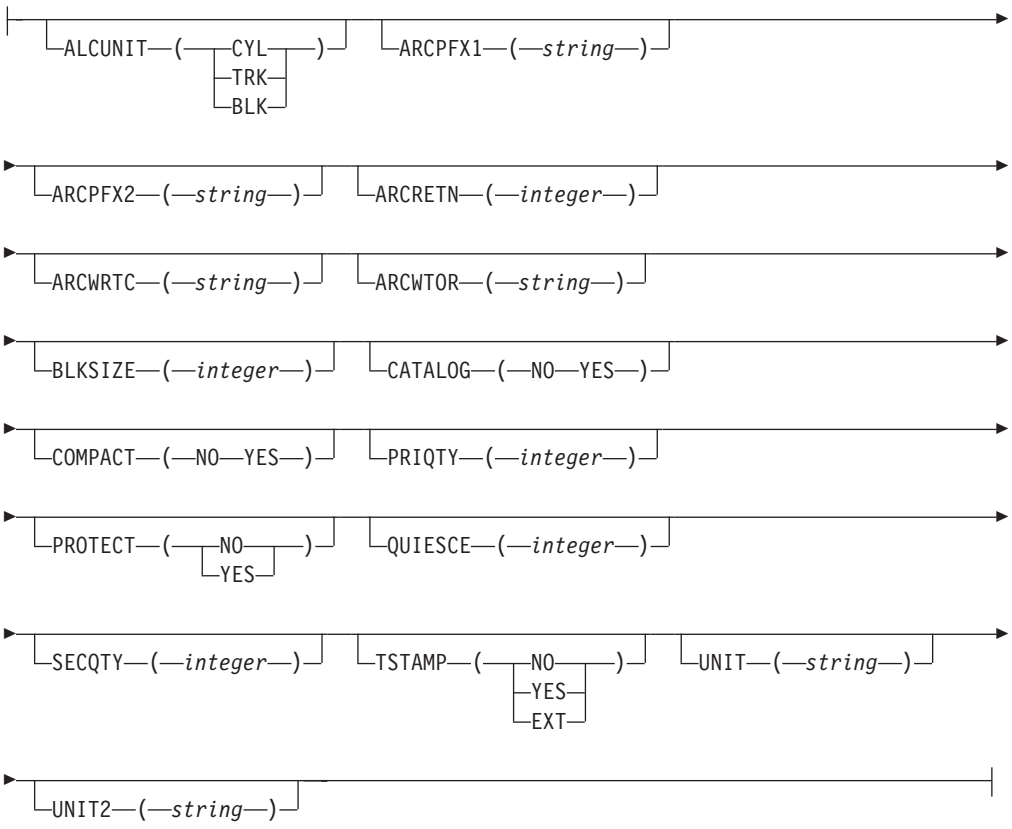
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: SET ARC

SET ARCHIVE



Parameter Block:



Notes:

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

Parameter descriptions for SET ARCHIVE

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 it was entered, only if you are using a queue-sharing group environment and if the command server is enabled.

You cannot use CMDSCOPE(*qmgr-name)* for commands issued from the first initialization input data set, CSQINP1.

* 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.

You cannot use CMDSCOPE(*) for commands issued from CSQINP1.

DEFAULT

Resets all the archive system parameters to the values set at queue manager startup.

Parameter block

For a full description of these parameters, see 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 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 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 - 9999.

Discarding archive log data sets is discussed in 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.

ARCWTOR
Specifies whether a message is to be sent to the operator and a reply 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.

SECQTY
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.Thhmmsst.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.Dyyyyddd.Thhmmsst.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 for SET ARCHIVE

1. The new values will be used at the next archive log off-load.

SET LOG

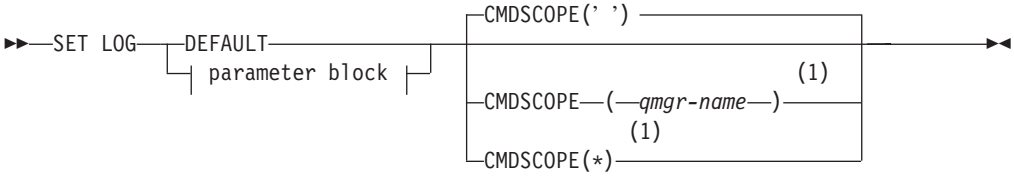
Use the MQSC command SET LOG to dynamically change certain log system parameter values that were initially set by your system parameter module at queue manager startup.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

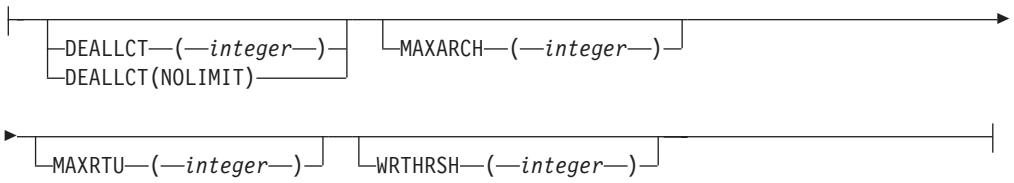
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: SET LOG

SET LOG



Parameter Block:



Notes:

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

Parameter descriptions for SET LOG

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.You cannot use CMDSCOPE(*qmgr-name)* for commands issued from the first initialization input data set, CSQINP1.

* 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.

   You cannot use CMDSCOPE(*) for commands issued from CSQINP1.

DEFAULT
Reset all the log system parameters to the values specified at queue manager startup.

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.

Note:

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.

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 for SET LOG

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).

SET SYSTEM

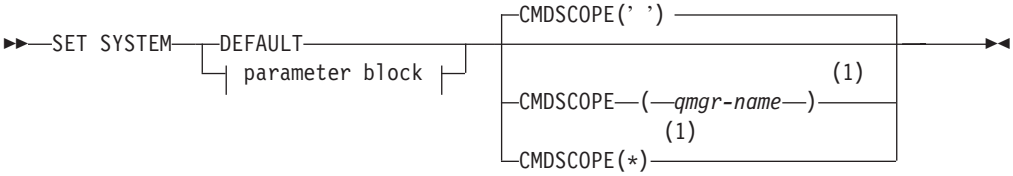
Use the MQSC command SET SYSTEM to dynamically change certain general system parameter values that were initially set from your system parameter module at queue manager startup.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

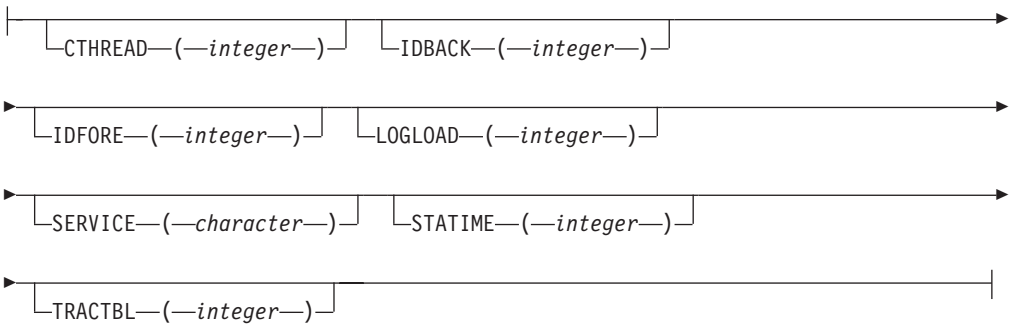
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: None

SET SYSTEM



Parameter Block:



Notes:

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

Parameter descriptions for SET SYSTEM

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.You cannot use CMDSCOPE(*qmgr-name)* for commands issued from the first initialization input data set, CSQINP1.

* 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.

You cannot use CMDSCOPE(*) for commands issued from CSQINP1.

DEFAULT
Resets all the general system parameters to the values set at queue manager startup.

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:

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 for SET SYSTEM

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.

START CHANNEL

Use the MQSC command START CHANNEL to start a channel.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	CR

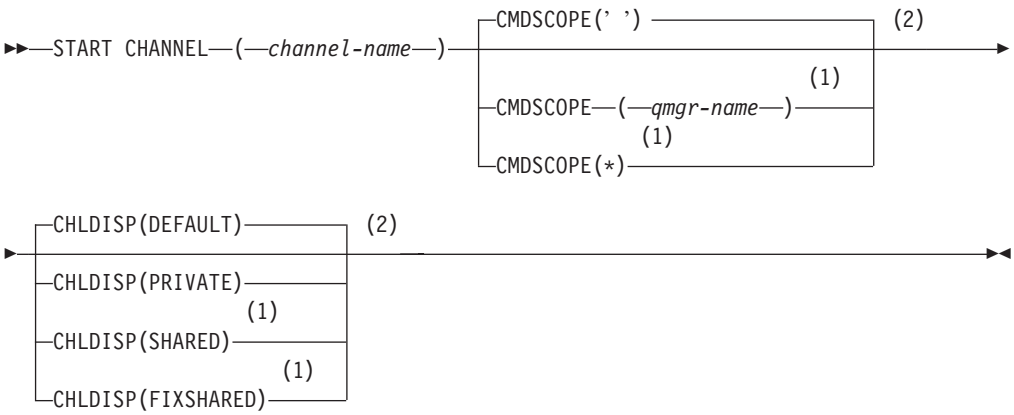
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

1. On z/OS, 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

START CHANNEL



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for START CHANNEL

(channel-name)
The name of the channel definition to be started. This is required. The name must be that of an existing channel.

CHLDISP

This parameter applies to z/OS only and can take the values of:

- DEFAULT
- PRIVATE
- SHARED
- FIXSHARED

If this parameter is omitted, then the DEFAULT value applies. This is taken from the default channel disposition attribute, DEFCDISP, of the channel object.

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 17

Table 17. CHLDISP and CMDSCOPE for START CHANNEL

CHLDISP	CMDSCOPE( ) or CMDSCOPE (local-qmgr)	CMDSCOPE (qmgr-name)	CMDSCOPE(*)
PRIVATE	Start as a private channel on the local queue manager	Start as a private channel on the named queue manager	Start as a private channel on all active queue managers

Table 17. CHLDISP and CMDSCOPE for START CHANNEL (continued)

CHLDISP	CMDSCOPE( ) or CMDSCOPE (local-qmgr)	CMDSCOPE (qmgr-name)	CMDSCOPE(*)
SHARED	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.	Not permitted	Not permitted
FIXSHARED	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.	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.	Not permitted

Channels started with CHLDISP(FIXSHARED) 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 FIXSHARED 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.

START CHINIT

Use the MQSC command START CHINIT to start a channel initiator.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	2CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

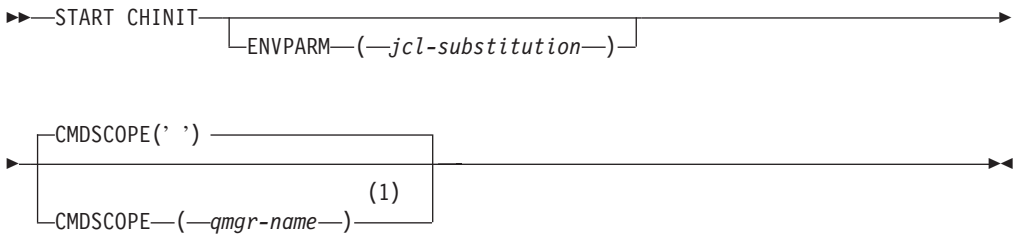
Note:

1. On z/OS, the command server must be running.
2. Although START CHINIT is permitted from CSQINP2, its processing is not complete (and the channel initiator is not available) until after CSQINP2 processing has finished. Therefore, certain other commands in CSQINP2 may fail because the channel initiator is not available.

Synonym: STA CHI

WebSphere MQ for z/OS

START CHINIT

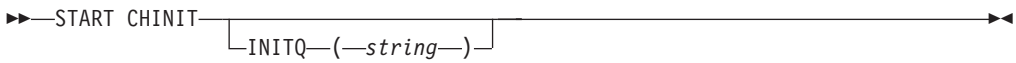


Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

WebSphere MQ on other platforms

START CHINIT



Parameter descriptions for START CHINIT

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, HP OpenVMS, HP-UX, Linux, OS/2 Warp, i5/OS, 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.

START CMDSERV

Use the MQSC command START CMDSERV to initialize the command server.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12C

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: STA CS

START CMDSERV

Usage notes for START CMDSERV

1. START CMDSERV 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 CMDSERV 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 CMDSERV 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 the MQSC command START LISTENER to start a channel listener.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

1. On UNIX systems, the command is valid only for AIX, HP-UX, Linux, and Solaris.
2. On z/OS:
   a. The command server and the channel initiator must be running.
   b. If IPADDR is not specified, the listener listens on all available IPv4 and IPv6 addresses.
   c. For TCP/IP, it is possible to listen on multiple addresses and port combinations.
   d. 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.
   e. 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.
   f. 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.

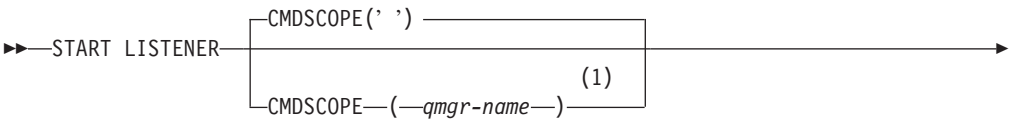
      You should disable IP forwarding and routing from other non-secure interfaces so that packets arriving at the other interface do not get passed to this specific address.

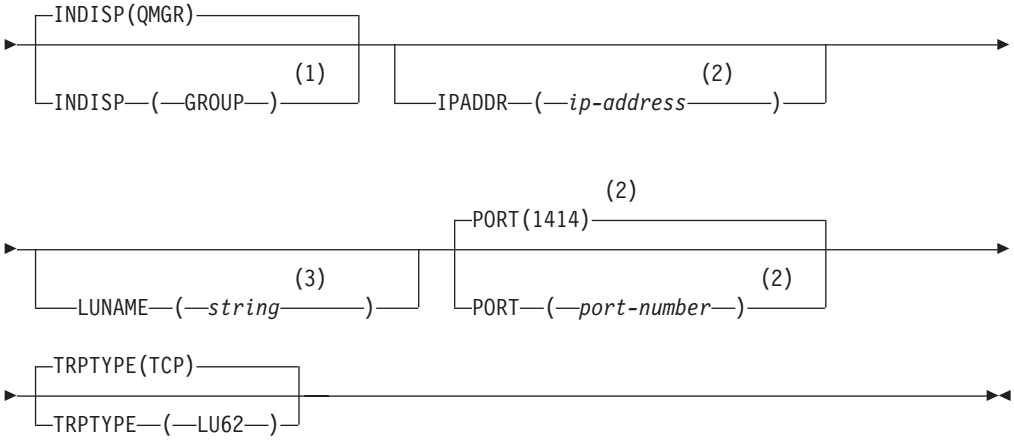
      Consult the appropriate TCP/IP documentation for information on how to do this.
3. On i5/OS, OS/2 Warp, UNIX systems, and Windows, this command is valid only for channels for which the transmission protocol (TRPTYPE) is TCP.

Synonym: STA LSTR

WebSphere MQ for z/OS

START LISTENER





Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only for TRPTYPE(TCP).

3 Valid only for TRPTYPE(LU62).

WebSphere MQ on other platforms

START LISTENER

Parameter descriptions for START LISTENER

(*name***)** Name of the listener to be started. If you specify this parameter, you cannot specify any other parameters.

If you do not specify a name (on platforms other than z/OS), the SYSTEM.DEFAULT.LISTENER.TCP is started.

This parameter is not valid on 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.

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.

This parameter is valid only on z/OS.

IPADDR
IP address for TCP/IP specified in IPv4 dotted decimal, IPv6 hexadecimal notation, or alphanumeric form. This is valid only if the transmission protocol (TRPTYPE) is TCP/IP.

This parameter is valid only on z/OS.

LUNAME(*string***)**
The symbolic destination name for the logical unit as specified in the APPC side information data set. (This must be the same LU that was specified for the queue manager, using the LUNAME parameter of the ALTER QMGR command.)

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 valid 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 valid 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 valid only on z/OS.

START QMGR

Use the MQSC command START QMGR to initialize the queue manager.

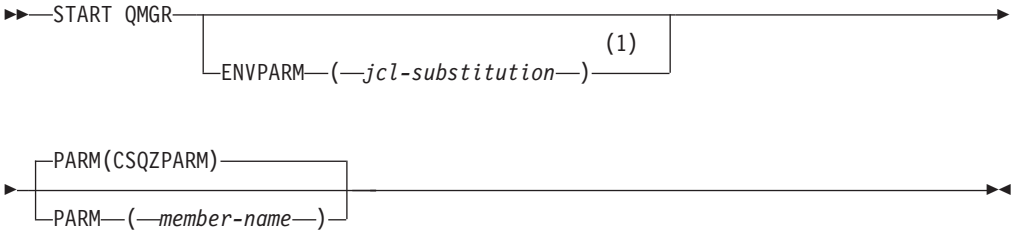
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					C

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

When the command has been completed, the queue manager is active and available to CICS, IMS, batch, and TSO applications.

Synonym: STA QMGR

START QMGR



Notes:

1 MSTR is accepted as a synonym for ENVPARM

Parameter descriptions for START QMGR

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.

START SERVICE

Use the MQSC command START SERVICE to start a service. The identified service definition is started within the queue manager and inherits the environment and security variables of the queue manager.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	

Synonym:

START SERVICE

Parameter descriptions for START SERVICE

(*service-name***)**
The name of the service definition to be started. This is required. The name must that of an existing service on this queue manager.

If the service is already running, and the operating system task is active, an error is returned.

START TRACE

Use the MQSC command START TRACE to start traces.

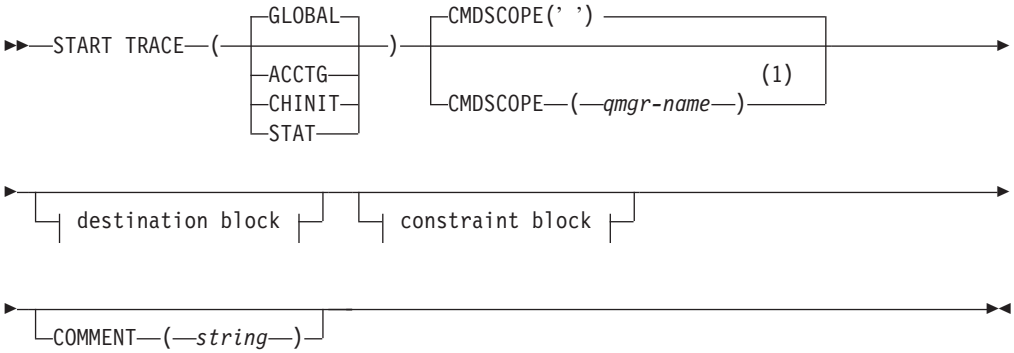
HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

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.

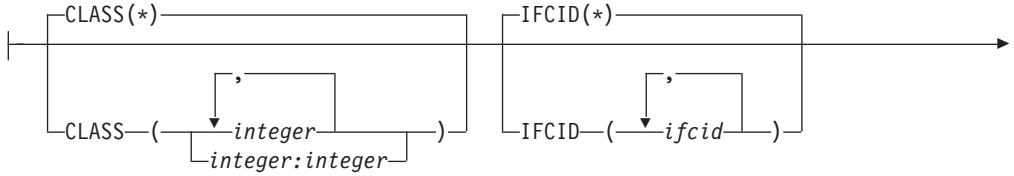
Synonym: STA TRACE

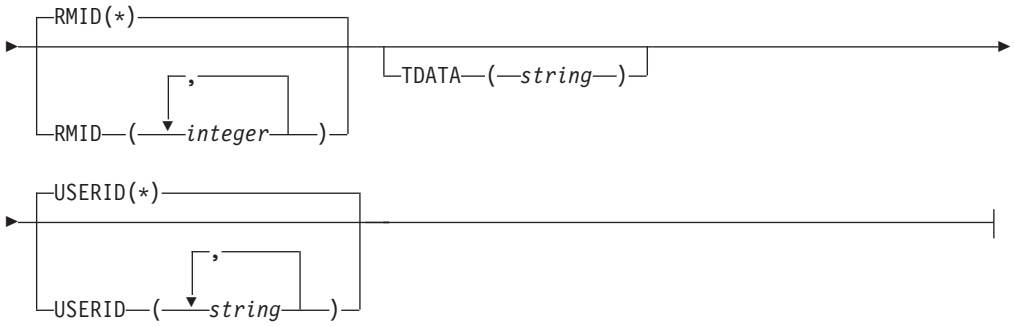
START TRACE



Destination block:

Constraint block:





Notes:

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

Parameter descriptions for START TRACE

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.

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 .

CHINIT
This includes data from the channel initiator. The synonym is CHI or DQM. If tracing for the channel initiator is started, it stops if the channel initiator stops.

Note that you cannot issue START TRACE(CHINIT) if the command server or the channel initiator is not running.

GLOBAL
This includes data from the entire queue manager except the channel initiator. 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 CHINIT.

SMF The System Management Facility (SMF). If used, the SMF must be functioning before the START TRACE command is issued. The SMF record numbers used by WebSphere MQ are 115 and 116.

SRV A serviceability routine reserved for IBM use only; not for general use.

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 18. Destinations allowed for each trace type

Type	GTF	RES	SMF	SRV
GLOBAL	Allowed	Default	No	Allowed
STAT	No	No	Default	Allowed
ACCTG	Allowed	No	Default	Allowed
CHINIT	No	Default	No	Allowed

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 19. Constraints allowed for each trace type

Type	CLASS	IFCID	RMID	USERID
GLOBAL	Allowed	Allowed	Allowed	Allowed
STAT	Allowed	No	No	No
ACCTG	Allowed	No	No	No
CHINIT	Allowed	Allowed	No	No

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 20. Descriptions of trace events and classes

Class	IFCID	Description
		Global trace
01	0000	Reserved for IBM service
02	0018	User parameter error detected in a control block
03	0016	User parameter error detected on entry to MQI
	0017	User parameter error detected on exit from MQI
	0018	User parameter error detected in a control block
04	Various	Reserved for IBM service
		Statistics trace
01	0001	Subsystem statistics
	0002	Queue manager statistics
		Accounting trace
01	0003	The CPU time spent processing MQI calls and a count of MQPUT and MQGET calls
03	0025	Enhanced accounting and statistical data
		CHINIT trace
01	0199	Reserved for IBM service
04	Various	Reserved for IBM service

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, ACCTG, or CHINIT traces.

(*) Starts a trace for all resource managers.

This is the default.

(*integer***)**

The identifying number of any resource manager in Table 21. You can use up to 8 of the allowed resource manager identifiers; do not use the same one twice.

Table 21. Resource Manager identifiers that are allowed

RMID	Resource manager
1	Initialization procedures
2	Agent services management
3	Recovery management
4	Recovery log management
6	Storage management
7	Subsystem support for allied memories
8	Subsystem support for subsystem interface (SSI) functions
12	System parameter management
16	Instrumentation commands, trace, and dump services
23	General command processing
24	Message generator
26	Instrumentation accounting and statistics
148	Connection manager
197	CF manager
199	Functional recovery
200	Security management
201	Data management
211	Lock management
212	Message management
213	Command server
215	Buffer management
242	WebSphere MQ-IMS bridge
245	DB2 manager

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, ACCTG, or CHINIT 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.

STOP CHANNEL

Use the MQSC command STOP CHANNEL to stop a channel.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	CR

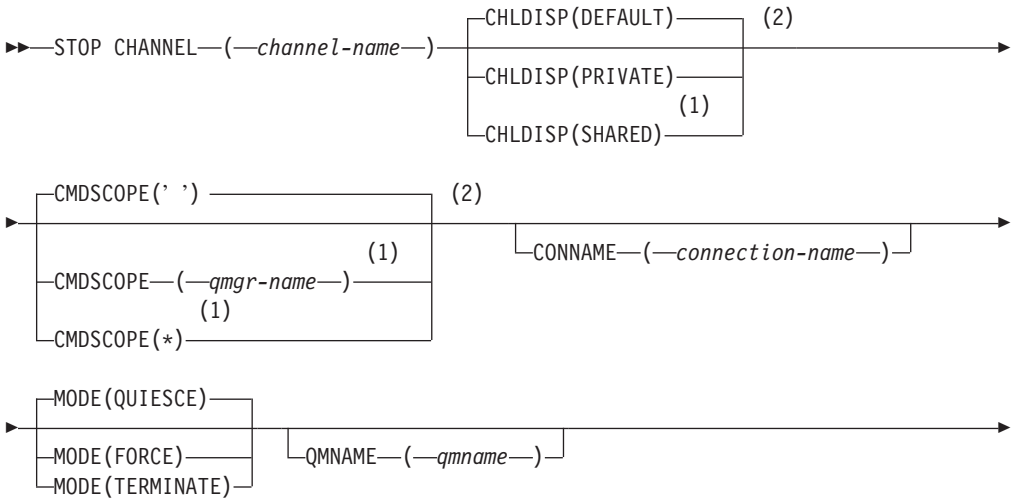
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

1. On z/OS, 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.
5. If you issue a `STOP CHANNEL(<channelname>) MODE QUIESCE` command on a server-connection channel with the sharing conversations feature enabled, the WebSphere MQ client infrastructure becomes aware of the stop request in a timely manner; this time is dependent upon the speed of the network. The client application becomes aware of the stop request as a result of issuing a subsequent call to WebSphere MQ.

Synonym: STOP CHL

STOP CHANNEL



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

2 Valid only on z/OS.

Parameter descriptions for STOP CHANNEL

(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:

- DEFAULT
- PRIVATE
- SHARED

If this parameter is omitted, then the DEFAULT value applies. This is taken from the default channel disposition attribute, DEFCDISP, of the channel object.

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 22

Table 22. CHLDISP and CMDSCOPE for STOP CHANNEL

CHLDISP	CMDSCOPE( ) or CMDSCOPE (local-qmgr)	CMDSCOPE (qmgr-name)	CMDSCOPE(*)
PRIVATE	Stop as a private channel on the local queue manager.	Stop as a private channel on the named queue manager	Stop as a private channel on all active queue managers

Table 22. CHLDISP and CMDSCOPE for STOP CHANNEL (continued)

CHLDISP	CMDSCOPE( ) or CMDSCOPE (local-qmgr)	CMDSCOPE (qmgr-name)	CMDSCOPE(*)
SHARED	For RCVR and SVRCONN channels, stop as shared channel on all active queue managers. 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.	Not permitted	Not permitted

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
- The next heartbeat (if heartbeats are being used)

before it stops.

For server-connection channels, allows the current connection to end.

If you issue a `STOP CHANNEL <channelname> MODE (QUIESCE)` command on a server-connection channel with the sharing conversations feature enabled, the WebSphere MQ client infrastructure becomes aware of the stop request in a timely manner; this time is dependent upon the speed of the network. The client application becomes aware of the stop request as a result of issuing a subsequent call to WebSphere MQ.

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 for STOP CHANNEL

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

STOP CHINIT

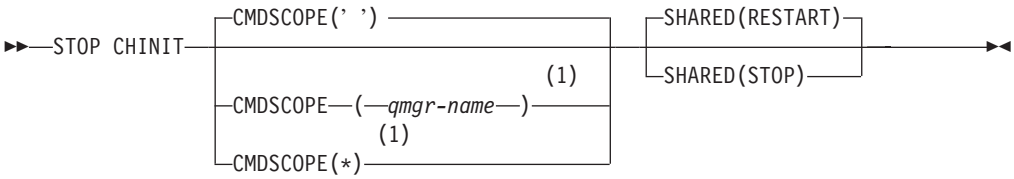
Use the MQSC command STOP CHINIT to stop a channel initiator. The command server must be running.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: STOP CHI

STOP CHINIT



Notes:

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

Parameter descriptions for STOP CHINIT

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.

SHARED

Specifies whether the channel initiator should attempt to restart any active sending channels, started with CHLDISP(SHARED), that it owns on another queue manager. The possible values are:

RESTART

Shared sending channels are to be restarted. This is the default.

STOP Shared sending channels are not to be restarted, so will become inactive.

(Active channels started with CHLDISP(FIXSHARED) are not restarted, and always become inactive.)

Usage notes for STOP CHINIT

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) STATUS(INACTIVE)
   - All other channels are stopped using STOP CHANNEL MODE(FORCE)

   See "STOP CHANNEL" on page 578 for information about what this involves.
2. You might receive communications-error messages as a result of issuing the STOP CHINIT command.

STOP CMDSERV

Use the MQSC command STOP CMDSERV to stop the command server.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12C

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: STOP CS

STOP CMDSERV

Usage notes for STOP CMDSERV

1. STOP CMDSERV 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 CMDSERV 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 CONN

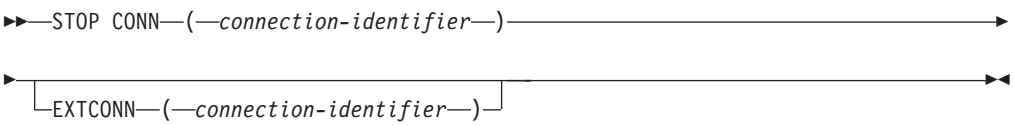
Use the MQSC command STOP CONN to break a connection between an application and the queue manager.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	

There may be circumstances in which the queue manager cannot implement this command when the success of this command cannot be guaranteed.

Synonym: STOP CONN

STOP CONN



Parameter descriptions for STOP CONN

(*connection-identifier***)**
The identifier of the connection definition for the connection to be broken.

When an application connects to WebSphere MQ, it is given a unique 24–byte connection identifier (ConnectionId). The value of CONN is formed by converting the last eight bytes of the ConnectionId to its 16–character hexadecimal equivalent.

EXTCONN
The value of EXTCONN is based on the first sixteen bytes of the ConnectionId converted to its 32–character hexadecimal equivalent.

Connections are identified by a 24-byte connection identifier. The connection identifier comprises a prefix, which identifies the queue manager, and a suffix which identifies the connection to that queue manager. By default, the prefix is for the queue manager currently being administered, but you can specify a prefix explicitly by using the EXTCONN parameter. Use the CONN parameter to specify the suffix.

When connection identifiers are obtained from other sources, specify the fully qualified connection identifier (both EXTCONN and CONN) to avoid possible problems related to non-unique CONN values.

STOP LISTENER

Use the MQSC command STOP LISTENER to stop a channel listener.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	CR

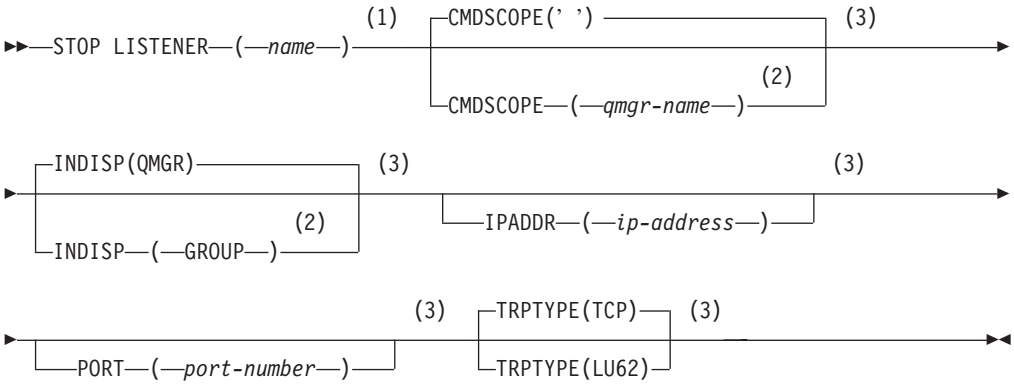
For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Note:

1. On z/OS:
   - The command server and the channel initiator must be running.
   - If a listener is listening on multiple addresses or ports, only the address and port combinations with the address, or port, specified are stopped.
   - If a listener is listening on all addresses for a particular port, a stop request for a specific IPADDR with the same port fails.
   - If neither an address nor a port is specified, all addresses and ports are stopped and the listener task ends.

Synonym: STOP LSTR

STOP LISTENER



Notes:

1 Not valid on z/OS.

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

3 Valid only on z/OS.

Parameter descriptions for STOP LISTENER

(*name***)** Name of the listener to be stopped. If you specify this parameter, you cannot specify any other parameters.

This parameter is required on all platforms other than z/OS where it is not a supported parameter.

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.

This parameter is valid only on z/OS.

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.

This parameter is valid only on z/OS.

IPADDR
IP address for TCP/IP specified in IPv4 dotted decimal, IPv6 hexadecimal notation, or alphanumeric form. This is valid only if the transmission protocol (TRPTYPE) is TCP/IP.

This parameter is valid only on z/OS.

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.

This parameter is valid only on z/OS.

TRPTYPE
Transmission protocol used. This is optional.

TCP TCP. This is the default if TRPTYPE is not specified.

LU62 SNA LU 6.2.

This parameter is valid only on z/OS.

The listener stops in quiesce mode (it disregards any further requests).

STOP QMGR

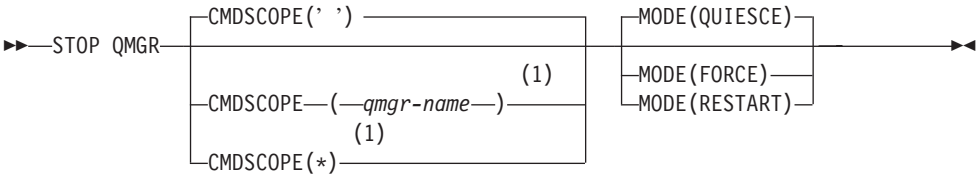
Use the MQSC command STOP QMGR to stop the queue manager.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					CR

For an explanation of the symbols in the z/OS column, see "Using commands in z/OS" on page 4.

Synonym: There is no synonym for this command.

STOP QMGR



Notes:

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

Parameter descriptions for STOP QMGR

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.

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.

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 the DISPLAY CONN command, and can cancel remaining connections using z/OS commands.

This option deregisters WebSphere MQ from the z/OS 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 z/OS 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 z/OS command CANCEL to terminate.

This option does not deregister WebSphere MQ from ARM, so the queue manager is eligible for immediate automatic restart.

STOP SERVICE

Use the MQSC command STOP SERVICE to stop a service.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
		X	X	X	

If the service is running, it is requested to stop. This command is processed asynchronously so may return before the service has stopped.

If the service that is requested to stop has no STOP command defined, an error is returned.

Synonym:

STOP SERVICE

Parameter descriptions for STOP SERVICE

(*service-name***)**
The name of the service definition to be stopped. This is required. The name must that of an existing service on this queue manager.

STOP TRACE

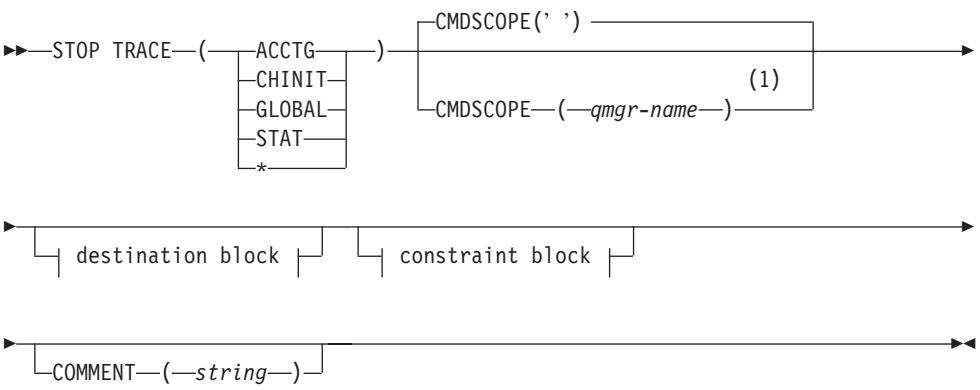
Use the MQSC command STOP TRACE to stop tracing.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
					12CR

For an explanation of the symbols in the z/OS column, see “Using commands in z/OS” on page 4.

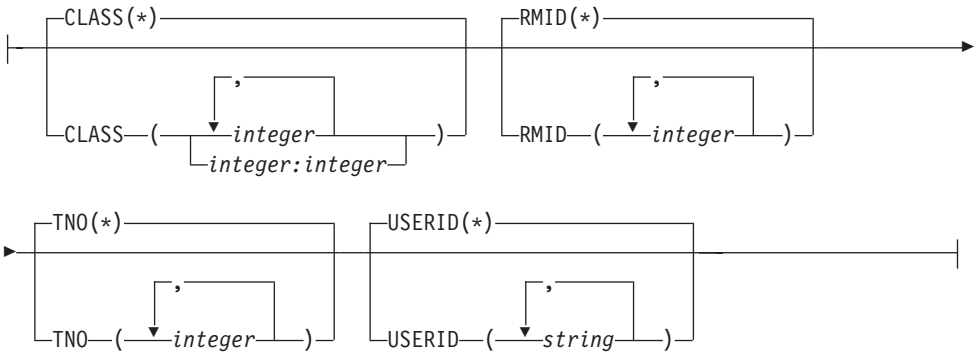
Synonym: There is no synonym for this command.

STOP TRACE



Destination block:

Constraint block:



Notes:

1 Valid only on z/OS when the queue manager is a member of a queue-sharing group.

Parameter descriptions for STOP TRACE

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.

CHINIT
Service data from the channel initiator. The synonym is CHI or DQM.

GLOBAL
Service data from the entire queue manager except for the channel initiator. 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.

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 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

RES A wrap-around table residing in the ECSA

SMF The System Management Facility

SRV A serviceability routine designed for problem diagnosis

Constraint block

CLASS(*integer***)**
Limits the command 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 command 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, ACCTG, or CHINIT trace type.

The default is RMID(*), which does not limit the command.

TNO(*integer***)**
Limits the command to particular traces, identified by their trace number (0 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.

0 is the trace that the channel initiator can start automatically. Traces 1 to 32 are those for queue manager or the channel initiator that can be started automatically by the queue manager, or manually, using the START TRACE command.

The default is TNO(*), which applies the command to all active traces with numbers 1 to 32, but **not** to the 0 trace. You can stop trace number 0 only be specifying it explicitly.

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 the STAT, ACCTG, or CHINIT trace type.

The default is USERID(*), which does not limit the command.

SUSPEND QMGR

Use the MQSC command SUSPEND QMGR to inform other queue mangers 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.

HP NSS	HP Open VMS	i5/OS	UNIX systems	Windows	z/OS
X	X	X	X	X	CR

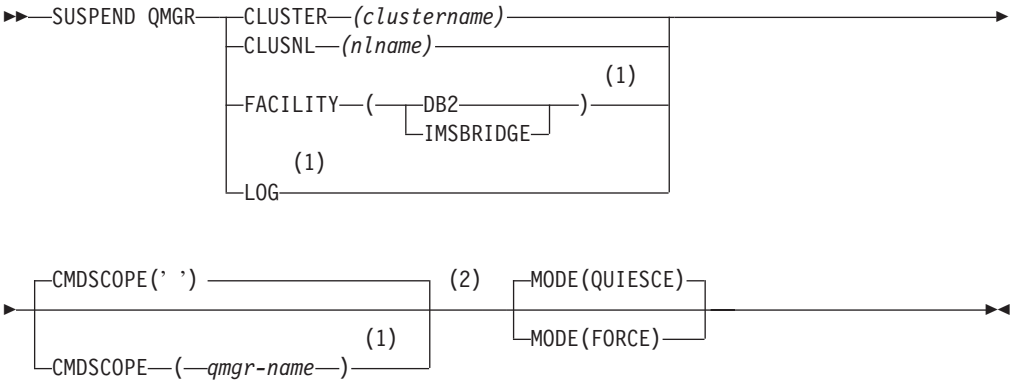
For an explanation of the symbols in the z/OS column, see"Using commands in z/OS" on page 4.

Note: On z/OS, if you define CLUSTER or CLUSNL:

1. The command fails if the channel initiator has not been started.
2. 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

SUSPEND QMGR



Notes:

1 Valid only on z/OS.

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

Parameter descriptions for SUSPEND QMGR

CLUSTER*(clustername)*
The name of the cluster for which availability is to be suspended.

CLUSNL*(nlname)*
The name of the namelist specifying a list of clusters to suspend availability for.

FACILITY
Specifies the facility to which connection is to be terminated.

DB2 Causes the existing connection to DB2 to be terminated. The connection is re-established when the RESUME QMGR FACILITY(DB2) command is executed. Any current or subsequent MQGET and MQPUT API requests are suspended until the DB2 connection is re-established by a RESUME QMGR FACILITY(DB2) command, or if the queue manager is stopped.

IMSBRIDGE
Stops the sending of messages from IMS Bridge queues to OTMA. When the tasks which transmit messages to IMS have been terminated, no further messages are sent to IMS until one of the following actions happens:

- OTMA or IMS is stopped and restarted
- WebSphere MQ is stopped and restarted
- A RESUME QMGR FACILITY(IMSBRIDGE) command is processed

Return messages from IMS OTMA to the queue manager are unaffected.

If you want to monitor progress of the command, issue the following command:

```
DIS Q(*) CMDSCOPE(qmgr) STGCLASS(bridge_stgclass) IPPROCS
```

and ensure that none of the queues are open. If any queue is open, use DISPLAY QSTATUS to verify that the MQ-IMS bridge does not have it open.

This parameter is only valid on z/OS.

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. If LOG is specified, the command can be issued only from the z/OS console.

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.

' ' 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, or FACILITY.

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-k,u
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.

© Copyright IBM Corp. 1999, 2008

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.

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX
CICS
DB2
i5/OS
IBM
IBMLink
IMS
MQSeries
MVS
OS/2

RACF
VTAM
z/OS

Java™ and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft®, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Index

A

ACCTCONO parameter
  ALTER QMGR 49
  DISPLAY QMGR 421
ACCTG parameter
  START TRACE 574
  STOP TRACE 594
ACCTINT parameter
  ALTER QMGR 49
  DISPLAY QMGR 421
ACCTMQI parameter
  ALTER QMGR 49
  DISPLAY QMGR 421
ACCTQ parameter
  ALTER QMGR 50
  DEFINE queues 91, 243
  DISPLAY QMGR 421
  DISPLAY QUEUE 456
ACTCHL parameter
  ALTER QMGR 50
  DISPLAY QMGR 421
ACTION parameter
  RESET CLUSTER 533
  RESET TPIPE 540
  RESOLVE CHANNEL 541
  RESOLVE INDOUBT 544
active thread, display 488
active trace, display list of 504
ACTIVREC parameter
  ALTER QMGR 50
  DISPLAY QMGR 421
ADAPTER parameter
  DEFINE LISTENER 210
  DISPLAY LISTENER 392
  DISPLAY LSSTATUS 398
administrator commands 11
ADOPTCHK parameter
  ALTER QMGR 51
  DISPLAY QMGR 421
ADOPTMCA parameter
  ALTER QMGR 51
  DISPLAY QMGR 421
AFFINITY parameter
  DISPLAY CHANNEL 336
  DISPLAY CLUSQMGR 370
ALCUNIT parameter, SET ARCHIVE 551
alias queue
  alter parameters 79
  define 230
  delete definition 301
  display attributes 446
ALL parameter
  DISPLAY AUTHINFO 316
  DISPLAY CFSTRUCT 326
  DISPLAY CHANNEL 332
  DISPLAY CHSTATUS 349
  DISPLAY CLUSQMGR 368
  DISPLAY CONN 379
  DISPLAY LISTENER 392
  DISPLAY LSSTATUS 397
ALL parameter *(continued)*
  DISPLAY NAMELIST 403
  DISPLAY PROCESS 408
  DISPLAY QMGR 420
  DISPLAY QMSTATUS 431
  DISPLAY QSTATUS 437
  DISPLAY QUEUE 451
  DISPLAY SBSTATUS 464
  DISPLAY SECURITY 467
  DISPLAY SERVICE 469
  DISPLAY STGCLASS 473
  DISPLAY SUB 478
  DISPLAY SVSTATUS 484
  DISPLAY TOPIC 493
  DISPLAY TPSTATUS 499
ALTDATE parameter
  DISPLAY AUTHINFO 318
  DISPLAY CFSTRUCT 326
  DISPLAY CHANNEL 336
  DISPLAY CLUSQMGR 370
  DISPLAY LISTENER 392
  DISPLAY NAMELIST 405
  DISPLAY PROCESS 409
  DISPLAY QMGR 421
  DISPLAY QUEUE 456
  DISPLAY SERVICE 469
  DISPLAY STGCLASS 474
  DISPLAY SUB 478
  DISPLAY TOPIC 495
ALTER AUTHINFO command 12
ALTER BUFFPOOLcommand 13
ALTER CFSTRUCT command 14
ALTER CHANNEL command 15
ALTER LISTENER command 38
ALTER NAMELIST command 40
ALTER PROCESS command 41
ALTER PSID command 43
ALTER QALIAS command 79
ALTER QLOCAL command 81
ALTER QMGR command 44
ALTER QMODEL command 85
ALTER QREMOTE command 88
ALTER SECURITY command 112
ALTER SERVICE command 114
ALTER STGCLASS command 115
alter sub
  alter 116
ALTER SUB command 116
ALTER TOPIC command 121
ALTER TRACE command 123
ALTTIME parameter
  DISPLAY AUTHINFO 318
  DISPLAY CFSTRUCT 326
  DISPLAY CHANNEL 336
  DISPLAY CLUSQMGR 371
  DISPLAY LISTENER 392
  DISPLAY NAMELIST 405
  DISPLAY PROCESS 409
  DISPLAY QMGR 422
  DISPLAY QUEUE 457
  DISPLAY SERVICE 469
ALTTIME parameter *(continued)*
  DISPLAY STGCLASS 474
  DISPLAY SUB 478
  DISPLAY TOPIC 495
APPLICID parameter
  DEFINE PROCESS 222
  DISPLAY PROCESS 409
APPLTAG parameter, DISPLAY CONN 380
APPLTAG parameter, DISPLAY QSTATUS 441
APPLTYPE parameter
  DEFINE PROCESS 222
  DISPLAY CONN 380
  DISPLAY PROCESS 409
  DISPLAY QSTATUS 442
archive
  display 313
  set 550
ARCHIVE LOG command 125
ARCPFX1 parameter, SET ARCHIVE 551
ARCPFX2 parameter, SET ARCHIVE 552
ARCRETN parameter, SET ARCHIVE 552
ARCWRTC parameter, SET ARCHIVE 552
ARCWTOR parameter, SET ARCHIVE 552
ASID parameter, DISPLAY CONN 381
ASID parameter, DISPLAY QSTATUS 442
ASTATE parameter
  DISPLAY CONN 381, 384, 442
authentication information
  alter 12
  define 134
  delete 288
  display 315
AUTHINFO parameter
  DEFINE AUTHINFO 135
  DELETE AUTHINFO 288
  DISPLAY AUTHINFO 315
AUTHOREV parameter
  ALTER QMGR 51
  DISPLAY QMGR 422
AUTHTYPE parameter
  DEFINE AUTHINFO 135
  DISPLAY AUTHINFO 318
auto-definition exit program 52, 422
auto-definition of channels 52, 422
AUTOSTART parameter
  DEFINE CHANNEL 176
  DISPLAY CHANNEL 336

B

backing up the log 125
BACKLOG parameter
  DEFINE LISTENER 210

© Copyright IBM Corp. 1999, 2008

BACKLOG parameter *(continued)*
DISPLAY LISTENER 392
DISPLAY LSSTATUS 398
BACKUP CFSTRUCT
keywords and parameters 128
BACKUP CFSTRUCT command 128
BATCHES parameter, DISPLAY CHSTATUS 354
BATCHHB parameter
DEFINE CHANNEL 177
DISPLAY CHANNEL 336
DISPLAY CLUSQMGR 371
BATCHINT parameter
DEFINE CHANNEL 177
DISPLAY CHANNEL 336
DISPLAY CLUSQMGR 371
BATCHSZ parameter
DEFINE CHANNEL 177
DISPLAY CHANNEL 336
DISPLAY CHSTATUS 355
DISPLAY CLUSQMGR 371
BLKSIZE parameter, SET ARCHIVE 552
bootstrap data set, recover 516
BOQNAME parameter
DEFINE queues 91, 244
DISPLAY QUEUE 457
BOTHRESH parameter
DEFINE queues 91, 244
DISPLAY QUEUE 457
BRIDGEEV parameter
ALTER QMGR 51
DISPLAY QMGR 422
BROWSE parameter, DISPLAY QSTATUS 443
BSDS, recover 516
buffer pool
alter 13
buffer pool, defining 139
buffer pool, deleting 290
BUFFERS parameter, DEFINE BUFFPOOL, ALTER BUFFPOOL 13, 139
buffpool
alter 13
BUFFPOOL parameter, ALTER BUFFPOOL 13
BUFFPOOL parameter, DEFINE BUFFPOOL 139
BUFFPOOL parameter, DEFINE PSID 227
BUFFPOOL parameter, DELETE BUFFPOOL 290
BUFSRCVD parameter, DISPLAY CHSTATUS 355
BUFSSENT parameter, DISPLAY CHSTATUS 355
building command scripts 3
building commands
characters with special meanings 2
rules for 1
BYTSRCVD parameter, DISPLAY CHSTATUS 355
BYTSSENT parameter, DISPLAY CHSTATUS 355

C

CANCEL OFFLOAD parameter, ARCHIVE LOG 126
CATALOG parameter, SET ARCHIVE 552
CCSID parameter
ALTER QMGR 52
DISPLAY QMGR 422
CFLEVEL parameter
DEFINE CFSTRUCT 141
DISPLAY CFSTRUCT 327
RECOVER CFSTRUCT 517
cfstruct
alter 14
backup 128
define 141
delete 291
display 325
display status 319
recover 517
CFSTRUCT parameter
BACKUP CFSTRUCT 128
DEFINE CFSTRUCT 141
DEFINE queues 92, 244
DELETE CFSTRUCT 291
DISPLAY CFSTATUS 319
DISPLAY CFSTRUCT 325
DISPLAY QUEUE 451
CHAD parameter
ALTER QMGR 52
DISPLAY QMGR 422
CHADEV parameter
ALTER QMGR 52
DISPLAY QMGR 422
CHADEXIT parameter
ALTER QMGR 52
DISPLAY QMGR 422
channel
alter parameters 15
auto-definition 52, 422
define parameters 144
delete definition 292
display 328
ping 512
reset 530
resolve 541
start 561
start initiator 565
start listener 568
stop 578
channel initiator
start 565
stop 583
CHANNEL parameter
DEFINE CHANNEL 175
DELETE CHANNEL 292
DISPLAY CHANNEL 330
DISPLAY CLUSQMGR 368
DISPLAY CONN 381
DISPLAY QSTATUS 443
PING CHANNEL 512
RESET CHANNEL 531
RESOLVE CHANNEL 541
START CHANNEL 561
STOP CHANNEL 579
channel status, displaying 343
channels, rules for names of 8
CHIADAPS parameter
ALTER QMGR 53
DISPLAY QMGR 422
CHIDISPS parameter
ALTER QMGR 53
DISPLAY QMGR 422
CHINIT parameter
DISPLAY QMGR 420
DISPLAY QMSTATUS 431
START TRACE 574
STOP TRACE 594
CHISERVP parameter
ALTER QMGR 54
DISPLAY QMGR 422
CHLDISP parameter
DISPLAY CHSTATUS 350
PING CHANNEL 513
RESET CHANNEL 531
RESOLVE CHANNEL 542
START CHANNEL 562
STOP CHANNEL 579
CHLEV parameter
ALTER QMGR 54
DISPLAY QMGR 422
CHLTYPE parameter
DEFINE CHANNEL 178
DISPLAY CHANNEL 336
CHLTYPE parameter, DISPLAY CHSTATUS 352
CHSTADA parameter, DISPLAY CHSTATUS 355
CHSTATI parameter, DISPLAY CHSTATUS 355
CHSTATUS parameter, DISPLAY CHSTATUS 348
CLASS parameter
ALTER TRACE 124
DISPLAY TRACE 506
START TRACE 576
STOP TRACE 595
CLEAR QLOCAL command 130
CLEAR TOPIC command 131
CLEAR TOPICSTR 132
CLNTWGHT parameter
DISPLAY CHANNEL 337
DISPLAY CLUSQMGR 371
CLRTYPE parameter
CLEAR TOPICSTR 132
CLUSDATE parameter
DISPLAY CLUSQMGR 369
DISPLAY QUEUE 457
DISPLAY TOPIC 495
CLUSINFO parameter
DISPLAY TOPIC 494
CLUSINFO parameter, DISPLAY QUEUE 451
CLUSNL parameter
DEFINE CHANNEL 179
DEFINE queues 92, 245
DISPLAY CHANNEL 337
DISPLAY QUEUE 451, 457
RESUME QMGR 547
SUSPEND QMGR 596
CLUSQMGR parameter
DISPLAY CLUSQMGR 367
DISPLAY TOPIC 495

CLUSQMGR parameter, DISPLAY QUEUE 457
CLUSQT parameter, DISPLAY QUEUE 457
cluster
  refresh 519
  reset 533
CLUSTER parameter
  DEFINE CHANNEL 179
  DEFINE queues 93, 245
  DEFINE TOPIC 280
  DISPLAY CHANNEL 337
  DISPLAY CLUSQMGR 368
  DISPLAY QMGR 420
  DISPLAY QUEUE 451, 457
  DISPLAY TOPIC 495
  REFRESH CLUSTER 520
  RESET CLUSTER 533
  RESUME QMGR 547
  SUSPEND QMGR 596
cluster queue manager, display 364
cluster queue, display attributes 446
clusters, rules for names of 8
CLUSTIME parameter
  DISPLAY CLUSQMGR 369
  DISPLAY QUEUE 457
  DISPLAY TOPIC 495
CLWLDATA parameter
  ALTER QMGR 54
  DISPLAY QMGR 422
CLWLEXIT parameter
  ALTER QMGR 54
  DISPLAY QMGR 422
CLWLLEN parameter
  ALTER QMGR 54
  DISPLAY QMGR 422
CLWLMRUC parameter
  ALTER QMGR 55
  DISPLAY QMGR 423
CLWLPRTY parameter
  DEFINE CHANNEL 179
  DEFINE queues 93, 245
  DISPLAY CHANNEL 337
  DISPLAY CLUSQMGR 371
  DISPLAY QUEUE 457
CLWLRANK parameter
  DEFINE CHANNEL 179
  DEFINE queues 93, 245
  DISPLAY CHANNEL 337
  DISPLAY CLUSQMGR 371
  DISPLAY QUEUE 457
CLWLUSEQ parameter
  ALTER QMGR 55
  DEFINE queues 93, 246
  DISPLAY QMGR 423
  DISPLAY QUEUE 457
CLWLWGHT parameter
  DEFINE CHANNEL 180
  DISPLAY CHANNEL 337
  DISPLAY CLUSQMGR 371
CMDEV parameter
  ALTER QMGR 55
  DISPLAY QMGR 423
CMDLEVEL parameter, DISPLAY QMGR 423
CMDSCOPE parameter
  ALTER queue manager 55
CMDSCOPE parameter *(continued)*
  ALTER SECURITY 112
  ALTER SUB 117
  ALTER TRACE 124
  ARCHIVE LOG 126
  BACKUP CFSTRUCT 128
  CLEAR QLOCAL 130
  CLEAR TOPICSTR 132
  DEFINE AUTHINFO 135
  DEFINE CHANNEL 179
  DEFINE LOG 212
  DEFINE NAMELIST 217
  DEFINE PROCESS 223
  DEFINE queues 93, 245
  DEFINE STGCLASS 269
  DEFINE SUB 274
  DEFINE TOPIC 280, 536
  DELETE AUTHINFO 288
  DELETE CHANNEL 293
  DELETE NAMELIST 296
  DELETE PROCESS 298
  DELETE queues 303
  DELETE STGCLASS 309
  DELETE SUB 307
  DELETE TOPIC 311
  DISPLAY ARCHIVE 313
  DISPLAY AUTHINFO 317
  DISPLAY CHANNEL 332
  DISPLAY CHINIT 341
  DISPLAY CHSTATUS 350
  DISPLAY CLUSQMGR 368
  DISPLAY CONN 379
  DISPLAY LOG 394
  DISPLAY MAXSMSGS 400
  DISPLAY NAMELIST 403
  DISPLAY PROCESS 408
  DISPLAY PUBSUB 411
  DISPLAY QMGR 420
  DISPLAY QSTATUS 437
  DISPLAY QUEUE 452
  DISPLAY SECURITY 466
  DISPLAY STGCLASS 473
  DISPLAY SUB 464, 479
  DISPLAY SYSTEM 486
  DISPLAY THREAD 489
  DISPLAY TOPIC 493
  DISPLAY TPSTATUS 500
  DISPLAY USAGE 507
  MOVE QLOCAL 509
  PING CHANNEL 513
  RECOVER BSDS 516
  RECOVER CFSTRUCT 517
  REFRESH CLUSTER 520
  REFRESH QMGR 522
  REFRESH SECURITY 527
  RESET CHANNEL 532
  RESET CLUSTER 533
  RESET QSTATS 538
  RESET TPIPE 539
  RESOLVE CHANNEL 543
  RESOLVE INDOUBT 545
  RESUME QMGR 547
  RVERIFY SECURITY 548
  SET ARCHIVE 551
  SET LOG 555
  SET SYSTEM 558
  START CHANNEL 563
CMDSCOPE parameter *(continued)*
  START CHINIT 565
  START TRACE 574
  STOP CHANNEL 579
  STOP CHINIT 583
  STOP LISTENER 588
  STOP QMGR 589
  STOP TRACE 594
  SUSPEND QMGR 597
CMDSERV parameter
  DISPLAY QMSTATUS 432
coded character set identifier 52
command scripts, building 3
command server
  display status 374
  start 567
  stop 585
command string
  entering quotes 2
  preserving case 2
COMMANDQ parameter, DISPLAY QMGR 423
commands
  rules for building 1
  rules for naming objects in 5
  rules for using 1
  synonym 2
COMMANDS parameter
  DEFINE LISTENER 210
  DISPLAY LISTENER 392
COMMENT parameter
  ALTER TRACE 124
  DISPLAY TRACE 505
  START TRACE 575
  STOP TRACE 594
COMPACT parameter, SET ARCHIVE 553
COMPHDR parameter
  DEFINE CHANNEL 180
  DISPLAY CHANNEL 337
  DISPLAY CLUSQMGR 371
COMPHDR parameter, DISPLAY CHSTATUS 355
COMPMSG parameter
  DEFINE CHANNEL 181
  DISPLAY CHANNEL 337
  DISPLAY CLUSQMGR 371
COMPMSG parameter, DISPLAY CHSTATUS 355
COMPRATE parameter, DISPLAY CHSTATUS 355
COMPTIME parameter, DISPLAY CHSTATUS 356
CONFIGEV parameter
  ALTER QMGR 56
  DISPLAY QMGR 423
CONN parameter, DISPLAY CONN 378
CONN parameter, STOP CONN 586
CONNAME parameter
  DEFINE AUTHINFO 135
  DEFINE CHANNEL 181
  DISPLAY AUTHINFO 318
  DISPLAY CHANNEL 337
  DISPLAY CHSTATUS 351
  DISPLAY CLUSQMGR 371
  DISPLAY QSTATUS 443
  STOP CHANNEL 581

CONNAME parameter, DISPLAY CONN 382
connection
  stop 586
connection, displaying 375
CONNOPTS parameter, DISPLAY CONN 382
CONNS parameter
  DISPLAY QMSTATUS 432
CONTROL parameter
  DEFINE LISTENER 210
  DEFINE SERVICE 266
  DISPLAY LISTENER 392
  DISPLAY LSSTATUS 398
  DISPLAY SERVICE 469
  DISPLAY SVSTATUS 484
CONVERT parameter
  DEFINE CHANNEL 184
  DISPLAY CHANNEL 337
  DISPLAY CLUSQMGR 371
COPY parameter
  DEFINE LOG 212
coupling facility structure
  alter 14
  backup 128
  define 141
  delete 291
  display 325
  display status 319
  recover 517
CPILEVEL parameter, DISPLAY QMGR 423
CRDATE parameter
  DISPLAY SUB 479
CRDATE parameter, DISPLAY QUEUE 457
CRTIME parameter
  DISPLAY SUB 479
CRTIME parameter, DISPLAY QUEUE 457
CURDEPTH parameter
  DISPLAY QSTATUS 438
  DISPLAY QUEUE 457
CURLUWID parameter, DISPLAY CHSTATUS 352
CURMSGS parameter, DISPLAY CHSTATUS 353
CURRENT parameter
  DISPLAY CHSTATUS 351
CURRLOG parameter
  DISPLAY QMSTATUS 432
CURSEQNO parameter, DISPLAY CHSTATUS 353
CURSHCNV parameter, DISPLAY CHSTATUS 356

D

DATALEN parameter, PING CHANNEL 514
DEADQ parameter
  ALTER QMGR 56
  DISPLAY QMGR 423
DEALLCT parameter, SET LOG 556
DEFAULT parameter
  SET ARCHIVE 551
  SET LOG 556
DEFAULT parameter *(continued)*
  SET SYSTEM 559
DEFBIND parameter
  DEFINE queues 94, 246
  DISPLAY QUEUE 458
DEFCDISP parameter
  DEFINE CHANNEL 184
  DISPLAY CHANNEL 337
DEFINE AUTHINFO command 134
DEFINE BUFFPOOL command 139
DEFINE CFSTRUCT command 141
DEFINE CHANNEL command 144
DEFINE LISTENER command 208
DEFINE LOG command 212
DEFINE MAXSMSGS command 214
DEFINE NAMELIST command 216
DEFINE PROCESS command 221
DEFINE PSID command 227
DEFINE QALIAS command 230
DEFINE QLOCAL command 232
DEFINE QMODEL command 236
DEFINE QREMOTE command 240
DEFINE SERVICE command 265
DEFINE STGCLASS command 268
DEFINE SUB command 273
DEFINE TOPIC command 278
DEFPRESP parameter
  DEFINE queues 94, 247
  DEFINE TOPIC 281
  DISPLAY QUEUE 458
  DISPLAY TOPIC 495
DEFPRTY parameter
  DEFINE queues 94, 247
  DEFINE TOPIC 280
  DISPLAY QUEUE 458
  DISPLAY TOPIC 495
DEFPSIST parameter
  DEFINE queues 94, 247
  DEFINE TOPIC 280
  DISPLAY QUEUE 458
  DISPLAY TOPIC 495
DEFREADA parameter
  DEFINE queues 95, 247
  DISPLAY QUEUE 458
DEFSOPT parameter
  DEFINE queues 95, 247
  DISPLAY QUEUE 458
DEFTYPE parameter
  DEFINE QMODEL 95, 247
  DISPLAY CLUSQMGR 369
  DISPLAY QUEUE 458
DEFXMITQ parameter
  ALTER QMGR 56
  DISPLAY QMGR 423
DELETE AUTHINFO command 288
DELETE BUFFPOOL command 290
DELETE CFSTRUCT command 291
DELETE CHANNEL command 292
DELETE LISTENER command 295
DELETE NAMELIST command 296
DELETE PROCESS command 298
DELETE PSID command 300
DELETE QALIAS command 301
DELETE QLOCAL command 301
DELETE QMODEL command 302
DELETE QREMOTE command 302
DELETE SERVICE command 306
DELETE STGCLASS command 309
DELETE SUB command 307
DELETE TOPIC command 311
DESCR parameter
  ALTER QMGR 56
  DEFINE AUTHINFO 136
  DEFINE CFSTRUCT 142
  DEFINE CHANNEL 184
  DEFINE LISTENER 210
  DEFINE NAMELIST 217
  DEFINE PROCESS 224
  DEFINE queues 96, 248
  DEFINE SERVICE 266
  DEFINE STGCLASS 269
  DEFINE TOPIC 281
  DISPLAY AUTHINFO 318
  DISPLAY CFSTRUCT 327
  DISPLAY CHANNEL 337
  DISPLAY CLUSQMGR 371
  DISPLAY LISTENER 393
  DISPLAY LSSTATUS 398
  DISPLAY NAMELIST 405
  DISPLAY PROCESS 409
  DISPLAY QMGR 423
  DISPLAY QUEUE 458
  DISPLAY SERVICE 470
  DISPLAY STGCLASS 474
  DISPLAY SVSTATUS 485
  DISPLAY TOPIC 496
DEST parameter
  ALTER SUB 117
  DEFINE SUB 274
  DISPLAY SUB 479
  DISPLAY TRACE 505
  START TRACE 575
  STOP TRACE 594
DEST parameter, DISPLAY CONN 385
DESTCLAS parameter
  ALTER SUB 117
  DEFINE SUB 274
  DISPLAY SUB 479
DESTCORL parameter
  ALTER SUB 118
  DEFINE SUB 274
  DISPLAY SUB 479
DESTQMGR parameter
  ALTER SUB 118
  DEFINE SUB 274
  DISPLAY SUB 479
DESTQMGR parameter, DISPLAY CONN 385
DETAIL parameter, DISPLAY TRACE 505
DISCINT parameter
  DEFINE CHANNEL 184
  DISPLAY CHANNEL 338
  DISPLAY CLUSQMGR 371
DISPLAY ARCHIVE command 313
DISPLAY AUTHINFO command 315
DISPLAY CFSTATUS command 319
DISPLAY CFSTRUCT command 325
DISPLAY CHANNEL command 328
DISPLAY CHINIT command 341
DISPLAY CHSTATUS command 343
DISPLAY CLUSQMGR command 364
DISPLAY CMDSERV command 374
DISPLAY CONN command 375

DISPLAY GROUP command 389
DISPLAY LISTENER command 390
DISPLAY LOG command 394
DISPLAY LSSTATUS command 396
DISPLAY MAXSMSGS command 400
DISPLAY NAMELIST command 401
DISPLAY PROCESS command 406
DISPLAY PUBSUB command 410
DISPLAY QALIAS command 446
DISPLAY QCLUSTER command 446
DISPLAY QLOCAL command 446
DISPLAY QMGR command 414
DISPLAY QMODEL command 446
DISPLAY QMSTATUS command 431
DISPLAY QREMOTE command 446
DISPLAY QSTATUS command 433
DISPLAY QUEUE command 446
DISPLAY SBSTATUS command 462
DISPLAY SECURITY command 466
DISPLAY servicecommand 468
DISPLAY STGCLASS command 471
DISPLAY SUB command 476
DISPLAY SVSTATUS command 483
DISPLAY SYSTEM command 486
DISPLAY THREAD command 488
DISPLAY TOPIC command 491
display topic status
  display 497
DISPLAY TPSTATUS command 497
DISPLAY TRACE command 504
DISPLAY USAGE command 507
DISTL parameter
  DEFINE queues 96, 248
  DISPLAY QMGR 424
  DISPLAY QUEUE 458
DNSGROUP parameter
  ALTER QMGR 56
  DISPLAY QMGR 424
DNSWLM parameter
  ALTER QMGR 57
  DISPLAY QMGR 424
DSN parameter, DEFINE PSID 227
dual BSDS, reestablish 516
DURABLE parameter
  DEFINE SUB 275
  DISPLAY SBSTATUS 464
  DISPLAY SUB 465, 479
DURSUB parameter
  DEFINE TOPIC 281
DURSUBS parameter
  DISPLAY TOPIC 496

E

ENVPARM parameter
  START CHINIT 566
  START QMGR 571
ENVRDATA parameter
  DEFINE PROCESS 224
  DISPLAY PROCESS 409
EVENT parameter
  DISPLAY QMGR 420
EXCLINT parameter
  BACKUP CFSTRUCT 128
EXITTIME parameter, DISPLAY CHSTATUS 356
EXPAND parameter, ALTER PSID 43
EXPAND parameter, DEFINE PSID 227
EXPIRY parameter
  ALTER SUB 118
  DEFINE SUB 275
  DISPLAY SUB 480
EXPRYINT parameter
  ALTER QMGR 57
  DISPLAY QMGR 424
EXTCONN parameter, DISPLAY CONN 379
EXTCONN parameter, STOP CONN 586
EXTURID parameter, DISPLAY CONN 382

F

FACILITY parameter
  RESUME QMGR 547
  SUSPEND QMGR 596
FIFO queue, DEFINE queues 101, 253
FORCE parameter
  ALTER QMGR 49
  queues 96, 248

G

generic values 3
GET parameter
  DEFINE queues 97, 249
  DISPLAY QUEUE 458
GLOBAL parameter
  START TRACE 574
  STOP TRACE 594
group, display 389

H

HARDENBO parameter
  DEFINE queues 97, 249
  DISPLAY QUEUE 459
HBINT parameter
  DEFINE CHANNEL 185
  DISPLAY CHANNEL 338
  DISPLAY CHSTATUS 356
  DISPLAY CLUSQMGR 371
HSTATE parameter
  DISPLAY QSTATUS 443
HSTATE parameter, DISPLAY CONN 385

I

IDBACK parameter, SET SYSTEM 559
IDFORE parameter, SET SYSTEM 559
IFCID parameter
  ALTER TRACE 124
  START TRACE 576
IGQ parameter
  ALTER QMGR 57
  DISPLAY QMGR 424
IGQAUT parameter
  ALTER QMGR 58
  DISPLAY QMGR 424
IGQUSER parameter, DISPLAY QMGR 424
IMS Tpipe, reset sequence numbers manually 539
in-doubt thread
  display 488
  resolve manually 544
INCLINT parameter, REFRESH QMGR 523
INDISP parameter
  START LISTENER 570
  STOP LISTENER 588
INDOUBT parameter
  RESOLVE INDOUBT 544
INDOUBT parameter, DISPLAY CHSTATUS 353
INDXTYPE parameter
  DEFINE queues 97, 249
  DISPLAY QUEUE 459
INHIBTEV parameter
  ALTER QMGR 59
  DISPLAY QMGR 424
INITQ parameter
  DEFINE queues 99, 251
  DISPLAY QUEUE 459
  START CHINIT 566
INPUT parameter, DISPLAY QSTATUS 443
INQUIRE parameter, DISPLAY QSTATUS 444
INTERVAL parameter
  ALTER SECURITY 113
  DISPLAY SECURITY 467
IPADDR parameter
  DEFINE LISTENER 210
  DISPLAY LISTENER 393
  DISPLAY LSSTATUS 398
  START LISTENER 570
  STOP LISTENER 588
IPADDRV parameter
  ALTER QMGR 59
  DISPLAY QMGR 424
IPPROCS parameter
  DISPLAY QSTATUS 438
  DISPLAY QUEUE 459

J

JOBNAME parameter, DISPLAY CHSTATUS 356

K

KAINT parameter
  DEFINE CHANNEL 185
  DISPLAY CHANNEL 338
  DISPLAY CHSTATUS 357
  DISPLAY CLUSQMGR 371

L

LDAPPWD parameter
  DEFINE AUTHINFO 136
  DISPLAY AUTHINFO 318
LDAPUSER parameter
  DEFINE AUTHINFO 136
  DISPLAY AUTHINFO 318

LGETDATE parameter
DISPLAY QSTATUS 438
LGETTIME parameter
DISPLAY QSTATUS 439
LIKE option
DEFINE AUTHINFO 136
DEFINE CHANNEL 186
DEFINE LISTENER 210
DEFINE NAMELIST 218
DEFINE PROCESS 224
DEFINE queues 99, 252
DEFINE SERVICE 266
DEFINE STGCLASS 269
DEFINE TOPIC 281
list of queue names
alter 40
define 216
delete 296
display 401
listener
alter 38
define 208
delete 295
start 568
stop 587
LISTENER parameter
DEFINE LISTENER 209
DELETE LISTENER 295
LISTENER parameter, DISPLAY LISTENER 391
LISTENER parameter, DISPLAY LSSTATUS 397
listener status, displaying 396
listener, displaying 390
local queue
alter parameters 81
clear 130, 131
define 232
delete definition 301
display attributes 446
move 509
LOCALEV parameter
ALTER QMGR 59
DISPLAY QMGR 424
LOCLADDR parameter
DEFINE CHANNEL 187
DISPLAY CHANNEL 338
DISPLAY CHSTATUS 357
DISPLAY CLUSQMGR 371
LOCLNAME parameter
DEFINE LISTENER 211
DISPLAY LISTENER 393
DISPLAY LSSTATUS 398
log
archive 125
define 212
display 394
set 555
LOG parameter
DEFINE LOG 212
RESUME QMGR 547
SUSPEND QMGR 597
LOGGEREV parameter
ALTER QMGR 59
DISPLAY QMGR 424
LOGLOAD parameter, SET SYSTEM 559
LONGRTS parameter, DISPLAY CHSTATUS 357
LONGRTY parameter
DEFINE CHANNEL 190
DISPLAY CHANNEL 338
DISPLAY CLUSQMGR 371
LONGTMR parameter
DEFINE CHANNEL 190
DISPLAY CHANNEL 338
DISPLAY CLUSQMGR 371
LPUTDATE parameter
DISPLAY QSTATUS 439
LPUTTIME parameter
DISPLAY QSTATUS 439
LSTLUWID parameter, DISPLAY CHSTATUS 353
LSTMSGDA parameter, DISPLAY CHSTATUS 357
LSTMSGTI parameter, DISPLAY CHSTATUS 357
LSTRTMR parameter
ALTER QMGR 60
DISPLAY QMGR 424
LSTSEQNO parameter, DISPLAY CHSTATUS 353
LU62ARM parameter
ALTER QMGR 60
DISPLAY QMGR 424
LU62CHL parameter
ALTER QMGR 60
DISPLAY QMGR 425
LUGROUP parameter
ALTER QMGR 60
DISPLAY QMGR 424
LUNAME parameter
ALTER QMGR 60
DISPLAY QMGR 424

M

MARKINT parameter
DISPLAY QMGR 425
MAXARCH parameter, SET LOG 556
MAXCHL parameter
ALTER QMGR 61
DISPLAY QMGR 425
MAXDEPTH parameter
DEFINE QLOCAL 100, 252
DISPLAY QUEUE 459
MAXHANDS parameter
ALTER QMGR 61
DISPLAY QMGR 425
maximum number of messages, define 214
MAXINST parameter
DEFINE CHANNEL 191, 338
DISPLAY CLUSQMGR 371
MAXINSTC parameter
DEFINE CHANNEL 191, 338
DISPLAY CLUSQMGR 372
MAXMSGL parameter
ALTER QMGR 62
DEFINE CHANNEL 191
DEFINE QLOCAL 100, 252
DISPLAY CHANNEL 338
DISPLAY CHSTATUS 357
DISPLAY CLUSQMGR 372
MAXMSGL parameter *(continued)*
DISPLAY QMGR 425
DISPLAY QUEUE 459
MAXPROPL parameter
ALTER QMGR 62
DISPLAY QMGR 425
MAXPRTY parameter, DISPLAY QMGR 425
MAXRTU parameter, SET LOG 556
MAXSHCNV parameter, DISPLAY CHSTATUS 357
maxsmsgs
define 214
display 400
MAXSMSGS parameter, DEFINE MAXSMSGS 214
MAXUMSGS parameter, ALTER QMGR 62
MCANAME parameter
DEFINE CHANNEL 192
DISPLAY CHANNEL 338
DISPLAY CLUSQMGR 372
MCASTAT parameter, DISPLAY CHSTATUS 357
MCATYPE parameter
DEFINE CHANNEL 192
DISPLAY CHANNEL 338
DISPLAY CLUSQMGR 372
MCAUSER parameter
DEFINE CHANNEL 192
DISPLAY CHANNEL 338
DISPLAY CHSTATUS 358
DISPLAY CLUSQMGR 372
MDURMDL parameter
DEFINE TOPIC 282
DISPLAY TOPIC 496
MEDIALOG parameter
DISPLAY QMSTATUS 432
DISPLAY QSTATUS 439
MNDURMDL parameter
DEFINE TOPIC 282
DISPLAY TOPIC 496
MODE parameter
ARCHIVE LOG 126
STOP CHANNEL 581
STOP QMGR 589
SUSPEND QMGR 598
model queue
alter parameters 85
define 236
delete definition 302
display attributes 446
MODENAME parameter
DEFINE CHANNEL 193
DISPLAY CHANNEL 338
DISPLAY CLUSQMGR 372
MONACLS parameter
ALTER QMGR 62
DISPLAY QMGR 425
MONCHL parameter
ALTER QMGR 63
DEFINE CHANNEL 193
DISPLAY CHANNEL 338
DISPLAY QMGR 425
MONCHL parameter, DISPLAY CHSTATUS 358

MONITOR parameter
DISPLAY CHSTATUS 351
DISPLAY QSTATUS 437
MONQ parameter
ALTER QMGR 63
DEFINE QLOCAL 100, 253
DISPLAY QMGR 425
DISPLAY QUEUE 459
MONQ parameter, DISPLAY QSTATUS 439
MOVE QLOCAL command 509
MQADMIN parameter
REFRESH SECURITY 526
MQNLIST parameter
REFRESH SECURITY 527
MQPROC parameter
REFRESH SECURITY 527
MQQUEUE parameter
REFRESH SECURITY 527
MQSC commands 11
MRDATA parameter
DEFINE CHANNEL 193
DISPLAY CHANNEL 338
DISPLAY CLUSQMGR 372
MREXIT parameter
DEFINE CHANNEL 193
DISPLAY CHANNEL 338
DISPLAY CLUSQMGR 372
MRRTY parameter
DEFINE CHANNEL 194
DISPLAY CHANNEL 338
DISPLAY CLUSQMGR 372
MRTMR parameter
DEFINE CHANNEL 194
DISPLAY CHANNEL 338
DISPLAY CLUSQMGR 372
MSGAGE parameter, DISPLAY QSTATUS 439
MSGDATA parameter
DEFINE CHANNEL 194
DISPLAY CHANNEL 338
DISPLAY CLUSQMGR 372
MSGDLVSQ parameter
DEFINE queues 101, 253
DISPLAY QUEUE 459
MSGEXIT parameter
DEFINE CHANNEL 194
DISPLAY CHANNEL 339
DISPLAY CLUSQMGR 372
MSGS parameter, DISPLAY CHSTATUS 358
MXADMIN parameter
REFRESH SECURITY 527
MXNLIST parameter
REFRESH SECURITY 527
MXPROC parameter
REFRESH SECURITY 527
MXQUEUE parameter
REFRESH SECURITY 527
MXTOPIC parameter
REFRESH SECURITY 527

N

NAMCOUNT parameter, DISPLAY NAMELIST 405
NAME parameter, REFRESH QMGR 523
name spaces 5
namelist
alter 40
define 216
delete 296
display contents 401
rules for names of 8
NAMELIST parameter
DELETE NAMELIST 296
DISPLAY NAMELIST 402
NAMELIST parameter, DEFINE NAMELIST 217
NAMES parameter
DEFINE NAMELIST 218
DISPLAY NAMELIST 405
NETPRTY parameter
DEFINE CHANNEL 195
DISPLAY CHANNEL 339
DISPLAY CLUSQMGR 372
NETTIME parameter, DISPLAY CHSTATUS 358
NID parameter, DISPLAY CONN 382
NID parameter, RESOLVE INDOUBT 545
NLTYPE parameter
DEFINE NAMELIST 218
DISPLAY NAMELIST 404
NOHARDENBO parameter, DEFINE queues 97, 249
NOPURGE parameter, DELETE QLOCAL 304
NOREPLACE option
DEFINE AUTHINFO 137
DEFINE CHANNEL 199
DEFINE LISTENER 211
DEFINE NAMELIST 219
DEFINE PROCESS 226
DEFINE queues 106, 258
DEFINE SERVICE 267
DEFINE STGCLASS 271
DEFINE TOPIC 285
NOSHARE parameter, DEFINE queues 108, 260
NOTRIGGER parameter, DEFINE queues 110, 262
NPMCLASS parameter
DEFINE queues 101, 254
DISPLAY QUEUE 459
NPMSGDLV parameter
DEFINE TOPIC 282
DISPLAY TOPIC 496
NPMSPEED parameter
DEFINE CHANNEL 196
DISPLAY CHANNEL 339
DISPLAY CLUSQMGR 372
NPMSPEED parameter, DISPLAY CHSTATUS 358
NTBNAMES parameter
DEFINE LISTENER 211
DISPLAY LISTENER 393
DISPLAY LSSTATUS 398

O

OBJECT parameter, REFRESH QMGR 523
objects, reserved names 8
OBJNAME parameter, DISPLAY CONN 386
OBJTYPE parameter, DISPLAY CONN 386
OBSMSGS parameter, DISPLAY GROUP 389
OPENOPTS parameter, DISPLAY CONN 386
OPENTYPE parameter, DISPLAY QSTATUS 437
operator commands 11
OPORTMAX parameter
ALTER QMGR 64
DISPLAY QMGR 425
OPORTMIN parameter
ALTER QMGR 64
DISPLAY QMGR 426
OPPROCS parameter
DISPLAY QSTATUS 440
DISPLAY QUEUE 459
OUTPUT parameter, DISPLAY QSTATUS 444

P

page set
alter 43
define 227, 268
delete 300
display usage 507
PARENT parameter
ALTER QMGR 64
DISPLAY QMGR 426
PARM parameter
START QMGR 571
PASSTKTA parameter
DEFINE STGCLASS 270
DISPLAY STGCLASS 474
PASSWORD parameter
DEFINE CHANNEL 196
DISPLAY CHANNEL 339
DISPLAY CLUSQMGR 372
PERFMEV parameter
ALTER QMGR 65
DISPLAY QMGR 426
PID parameter
DISPLAY LSSTATUS 398
DISPLAY SVSTATUS 485
PID parameter, DISPLAY CONN 382
PID parameter, DISPLAY QSTATUS 444
PING CHANNEL command 512
PING QMGR command 515
PLATFORM parameter, DISPLAY QMGR 426
PMSGDLV parameter
DEFINE TOPIC 283
DISPLAY TOPIC 496
PORT parameter
DEFINE LISTENER 211
DISPLAY LISTENER 393
DISPLAY LSSTATUS 398
PORT parameter, STOP LISTENER 588

priority queue, DEFINE queues 101, 253
PRIQTY parameter, SET ARCHIVE 553
process definition
  alter 41
  define 221
  delete 298
  display 406
PROCESS parameter
  DEFINE PROCESS 222
  DEFINE queues 102, 254
  DELETE PROCESS 298
  DISPLAY PROCESS 407
  DISPLAY QUEUE 459
processes, rules for names of 8
PROPCTL parameter
  DISPLAY CHANNEL 339
  DISPLAY CLUSQMGR 372
  DISPLAY QUEUE 459
PROTECT parameter, SET ARCHIVE 553
PROXYSUB parameter
  DEFINE TOPIC 283
  DISPLAY TOPIC 496
PSBNAME parameter, DISPLAY CONN 382
PSBNAME parameter, DISPLAY QSTATUS 444
PSID parameter
  DEFINE PSID 227, 300
  DEFINE STGCLASS 270
  DISPLAY QUEUE 452
  DISPLAY STGCLASS 473
  DISPLAY USAGE 507
PSID parameter, ALTER PSID 43
PSMODE parameter
  ALTER QMGR 65
  DISPLAY QMGR 426
PSNPMSG parameter
  ALTER QMGR 66
  DISPLAY QMGR 426
PSNPRES parameter
  ALTER QMGR 66
  DISPLAY QMGR 426
PSPROP parameter
  ALTER SUB 118
  DEFINE SUB 275
  DISPLAY SUB 480
PSRTYCNT parameter
  ALTER QMGR 67
  DISPLAY QMGR 426
PSSYNCPT parameter
  ALTER QMGR 67
  DISPLAY QMGR 426
PSTID parameter, DISPLAY CONN 383
PSTID parameter, DISPLAY QSTATUS 444
PUB parameter
  DEFINE TOPIC 283
  DISPLAY TOPIC 496
Pub status parameters
  DISPLAY TPSTATUS 502
PUBACCT option
  ALTER SUB 118
  DEFINE SUB 275
PUBACCT parameter
  DISPLAY SUB 480
PUBAPPID parameter
  ALTER SUB 118
  DEFINE SUB 275
  DISPLAY SUB 480
publish/subscribe
  display status information for hierarchy connections 410
PUBPRTY parameter
  ALTER SUB 118
  DEFINE SUB 275
  DISPLAY SUB 480
PUBSCOPE parameter
  DEFINE TOPIC 284
  DISPLAY TOPIC 496
PUBSUB parameter
  DISPLAY QMGR 421
PURGE parameter, DELETE QLOCAL 304
PUT parameter
  DEFINE queues 103, 255
  DISPLAY QUEUE 459
PUTAUT parameter
  DEFINE CHANNEL 197
  DISPLAY CHANNEL 339
  DISPLAY CLUSQMGR 372

Q

QALIAS parameter
  DELETE queues 303
QDEPTHHI parameter
  DEFINE QLOCAL 103, 255
  DISPLAY QUEUE 459
QDEPTHLO parameter
  DEFINE QLOCAL 103, 255
  DISPLAY QUEUE 459
QDPHIEV parameter
  DEFINE queues 103, 256
  DISPLAY QUEUE 459
QDPLOEV parameter
  DEFINE queues 104, 256
  DISPLAY QUEUE 460
QDPMAXEV parameter
  DEFINE queues 104, 256
  DISPLAY QUEUE 460
QLOCAL parameter
  DELETE queues 303
  MOVE QLOCAL 509
QMID parameter
  DISPLAY CLUSQMGR 369
  DISPLAY QMGR 427
  DISPLAY QUEUE 460
  DISPLAY TOPIC 496
  RESET CLUSTER 534
QMNAME parameter
  DEFINE CHANNEL 197
  DISPLAY CHANNEL 339
  DISPLAY PUBSUB 411
  DISPLAY QMGR 427
  DISPLAY QMSTATUS 432
  DISPLAY THREAD 489
  RESET CLUSTER 534
  RESOLVE INDOUBT 545
  STOP CHANNEL 582
QMODEL parameter
  DELETE queues 303
QMTYPE parameter, DISPLAY CLUSQMGR 369
QMURID parameter, DISPLAY CONN 383
QMURID parameter, DISPLAY QSTATUS 444
QREMOTE parameter
  DELETE queues 303
QSGDISP parameter
  CLEAR QLOCAL 131
  DEFINE CHANNEL 197
  DEFINE NAMELIST 219
  DEFINE PROCESS 225
  DEFINE queues 104, 257
  DEFINE STGCLASS 270
  DEFINE TOPIC 284
  DELETE AUTHINFO 289
  DELETE CHANNEL 293
  DELETE NAMELIST 297
  DELETE PROCESS 299
  DELETE QLOCAL 304
  DELETE STGCLASS 310
  DELETE TOPIC 312
  DISPLAY AUTHINFO 317
  DISPLAY CHANNEL 332
  DISPLAY NAMELIST 404
  DISPLAY PROCESS 408
  DISPLAY QSTATUS 440, 444
  DISPLAY QUEUE 452
  DISPLAY STGCLASS 473
  DISPLAY TOPIC 494
  MOVE QLOCAL 510
QSGDISP parameter, DISPLAY CONN 387
QSGNAME parameter, DISPLAY QMGR 427
QSTATS parameter, RESET QSTATS 537
QSTATUS parameter, DISPLAY QSTATUS 436
QSVCIEV parameter
  DEFINE queues 105, 258
  DISPLAY QUEUE 460
QSVCINT parameter
  DEFINE queues 106, 258
  DISPLAY QUEUE 460
QTIME parameter, DISPLAY QSTATUS 440
QTYPE parameter, DISPLAY QUEUE 460
queue attributes, display 446
queue manager
  alter parameters 44
  display parameters 414
  ping 515
  refresh parameters 522
  reset 535
  resume 546
  start 571
  stop 589
  suspend 596
queue manager alias, defining 240
queue manager status
  display 431
queue names 6
QUEUE parameter
  CLEAR QLOCAL 130
  DEFINE queues 91, 243

QUEUE parameter *(continued)*
DISPLAY QUEUE 450
queue status
displaying 433
reset 537
queues
rules for names of 6
QUEUES parameter, RESET CLUSTER 534
QUIESCE parameter, SET ARCHIVE 553

R

RAPPLTAG parameter
DISPLAY CHSTATUS 359
RCVDATA parameter
DEFINE CHANNEL 198
DISPLAY CHANNEL 339
DISPLAY CLUSQMGR 372
RCVEXIT parameter
DEFINE CHANNEL 198
DISPLAY CHANNEL 339
DISPLAY CLUSQMGR 372
RCVSEQ parameter, RESET TPIPE 540
RCVTIME parameter
ALTER QMGR 67
DISPLAY QMGR 427
RCVTMIN parameter
ALTER QMGR 68
DISPLAY QMGR 427
RCVTTYPE parameter
ALTER QMGR 68
DISPLAY QMGR 427
READA parameter, DISPLAY CONN 387
RECLOG parameter
DISPLAY QMSTATUS 432
RECOVER BSDS command 516
RECOVER CFSTRUCT
keywords and parameters 517
usage notes 518
RECOVER CFSTRUCT command 517
RECOVER parameter
DEFINE CFSTRUCT 142
DISPLAY CFSTRUCT 327
REFRESH CLUSTER command 519
REFRESH QMGR command 522
REFRESH SECURITY command 525
remote queue
alter parameters 88
define 240
delete local definition 302
display attributes 446
REMOTEEV parameter
ALTER QMGR 68
DISPLAY QMGR 427
REPLACE option
DEFINE AUTHINFO 137
DEFINE CHANNEL 199
DEFINE LISTENER 211
DEFINE NAMELIST 219
DEFINE PROCESS 226
DEFINE queues 106, 258
DEFINE SERVICE 267
DEFINE STGCLASS 271
DEFINE TOPIC 285
reply-to queue alias, defining 240
REPOS parameter
ALTER QMGR 68
DISPLAY QMGR 427
REFRESH CLUSTER 520
REPOSNL parameter
ALTER QMGR 69
DISPLAY QMGR 427
REQONLY parameter
ALTER SUB 119
DEFINE SUB 276
DISPLAY SUB 480
reserved names
objects 8
RESET CHANNEL command 530
RESET CLUSTER command 533
RESET QMGR command 535
RESET QSTATS command 537
RESET TPIPE command 539
RESOLVE CHANNEL command 541
RESOLVE INDOUBT command 544
RESUME QMGR command 546
RETINTVL parameter
DEFINE queues 107, 259
DISPLAY QUEUE 460
RMID parameter
DISPLAY TRACE 506
START TRACE 576
STOP TRACE 595
RNAME parameter
DEFINE QREMOTE 107, 259
DISPLAY QUEUE 460
ROUTEREC parameter
ALTER QMGR 69
DISPLAY QMGR 427
RQMNAME parameter
DEFINE QREMOTE 107, 260
DISPLAY QUEUE 460
RQMNAME parameter, DISPLAY CHSTATUS 359
rules for using commands 1
RVERIFY SECURITY command 548

S

SAVED parameter
DISPLAY CHSTATUS 351
sbstatus display
display 462
SCHINIT parameter
ALTER QMGR 69
DISPLAY QMGR 428
SCMDSERV parameter
ALTER QMGR 70
DISPLAY QMGR 428
SCOPE parameter
DEFINE queues 108, 260
DISPLAY QUEUE 460
SCYCASE parameter
ALTER QMGR 70
DISPLAY QMGR 428
SCYDATA parameter
DEFINE CHANNEL 199
DISPLAY CHANNEL 339
DISPLAY CLUSQMGR 372
SCYEXIT parameter
DEFINE CHANNEL 199
DISPLAY CHANNEL 339
SCYEXIT parameter *(continued)*
DISPLAY CLUSQMGR 373
SECQTY parameter, SET ARCHIVE 553
security
alter parameters 112
display parameters 466
rebuild 525
refresh 525
reverify 548
SECURITY parameter
RVERIFY SECURITY 548
SELECTOR parameter
ALTER SUB 119
DEFINE SUB 276
DISPLAY SUB 480
SENDDATA parameter
DEFINE CHANNEL 199
DISPLAY CHANNEL 339
DISPLAY CLUSQMGR 373
SENDEXIT parameter
DEFINE CHANNEL 200
DISPLAY CHANNEL 339
DISPLAY CLUSQMGR 373
SENDSEQ parameter, RESET TPIPE 540
SEQNUM parameter, RESET CHANNEL 532
sequence numbers, resetting on an IMS Tpipe 539
SEQWRAP parameter
DEFINE CHANNEL 200
DISPLAY CHANNEL 339
DISPLAY CLUSQMGR 373
service
alter 114
define 265
delete 306
start 572
start service 572
stop 591
service parameter
DEFINE SERVICE 266
SERVICE parameter
DELETE SERVICE 306
SERVICE parameter, DISPLAY SERVICE 468
SERVICE parameter, DISPLAY SVSTATUS 483
SERVICE parameter, START SERVICE 572
SERVICE parameter, STOP SERVICE 591
service status, displaying 483
service, displaying 468
SERVTYPE parameter
DEFINE SERVICE 267
DISPLAY SERVICE 470
SESSIONS parameter
DEFINE LISTENER 211
DISPLAY LISTENER 393
DISPLAY LSSTATUS 398
SET ARCHIVE command 550
SET LOG command 555
SET parameter, DISPLAY QSTATUS 445
SET SYSTEM command 558
SHARE parameter
DEFINE queues 108, 260
DISPLAY QUEUE 460

SHARECNV parameter
DEFINE CHANNEL 200
DISPLAY CHANNEL 339
DISPLAY CLUSQMGR 373
SHARED parameter, STOP CHINIT 583
SHORT parameter
DISPLAY CHSTATUS 351
SHORTRTS parameter, DISPLAY CHSTATUS 359
SHORTRTY parameter
DEFINE CHANNEL 201
DISPLAY CLUSQMGR 373
SHORTTMR parameter
DEFINE CHANNEL 201
DISPLAY CHANNEL 339
DISPLAY CLUSQMGR 373
SOCKET parameter
DEFINE LISTENER 211
DISPLAY LISTENER 393, 398
SQQMNAME parameter
ALTER QMGR 67
DISPLAY QMGR 428
SSLCAUTH parameter
DEFINE CHANNEL 202
DISPLAY CHANNEL 339
DISPLAY CLUSQMGR 373
SSLCERTI parameter
DISPLAY CHSTATUS 359
SSLCERTU parameter
DISPLAY CHSTATUS 359
SSLCIPH parameter
DEFINE CHANNEL 202
DISPLAY CHANNEL 339
DISPLAY CLUSQMGR 373
SSLCRLNL parameter
ALTER QMGR 70
DISPLAY QMGR 428
SSLCRYP parameter
ALTER QMGR 70
DISPLAY QMGR 428
SSLEV parameter
ALTER QMGR 70
DISPLAY QMGR 428
SSLFIPS parameter
ALTER QMGR 70
DISPLAY QMGR 428
SSLKEYDA parameter
DISPLAY CHSTATUS 359
SSLKEYR parameter
ALTER QMGR 70
DISPLAY QMGR 428
SSLKEYTI parameter
DISPLAY CHSTATUS 359
SSLPEER parameter
DEFINE CHANNEL 202
DISPLAY CHANNEL 339
DISPLAY CHSTATUS 359
DISPLAY CLUSQMGR 373
SSLRKEYC parameter
ALTER QMGR 70
SSLRKEYS parameter
DISPLAY CHSTATUS 359
SSLRLEYC parameter
DISPLAY QMGR 428
SSLTASKS parameter
ALTER QMGR 70
DISPLAY QMGR 428
START CHANNEL command 561
START CHINIT command 565
START CMDSERV command 567
START LISTENER command 568
START QMGR command 571
START SERVICE command 572
START TRACE command 573
STARTARG parameter
DEFINE SERVICE 267
DISPLAY SERVICE 470
DISPLAY SVSTATUS 485
STARTCMD parameter
DEFINE SERVICE 267
DISPLAY SERVICE 470
DISPLAY SVSTATUS 485
STARTDA parameter
DISPLAY LSSTATUS 398
DISPLAY SVSTATUS 485
STARTTI parameter
DISPLAY LSSTATUS 398
DISPLAY SVSTATUS 485
STAT parameter
START TRACE 574
STOP TRACE 594
STATACLS parameter
ALTER QMGR 73
DISPLAY QMGR 429
STATCHL parameter
ALTER QMGR 74
DEFINE CHANNEL 205
DISPLAY CHANNEL 339
DISPLAY QMGR 429
STATIME parameter, SET SYSTEM 559
STATINT parameter
ALTER QMGR 74
DISPLAY QMGR 429
STATMQI parameter
ALTER QMGR 74
DISPLAY QMGR 429
STATQ parameter
ALTER QMGR 75
DEFINE queues 108, 261
DISPLAY QMGR 429
DISPLAY QUEUE 460
STATUS parameter
DISPLAY CHSTATUS 353
DISPLAY CLUSQMGR 369
DISPLAY LSSTATUS 398
DISPLAY PUBSUB 411
DISPLAY QMSTATUS 432
DISPLAY SVSTATUS 485
STOP CHANNEL 582
STDER|R parameter
DISPLAY SERVICE 470
STDERR parameter
DEFINE SERVICE 267
DISPLAY SVSTATUS 485
STDOUT parameter
DEFINE SERVICE 267
DISPLAY SERVICE 470
DISPLAY SVSTATUS 485
STGCLASS parameter
DEFINE queues 109, 261
DEFINE STGCLASS 269
DELETE STGCLASS 309
DISPLAY QUEUE 454, 460
DISPLAY STGCLASS 472
STOP CHANNEL command 578
STOP CHINIT command 583
STOP CMDSERV command 585
STOP CONN command 586
STOP LISTENER command 587
STOP QMGR command 589
STOP SERVICE command 591
STOP TRACE command 592
STOPARG parameter
DEFINE SERVICE 267
DISPLAY SERVICE 470
DISPLAY SVSTATUS 485
STOPCMD parameter
DEFINE SERVICE 267
DISPLAY SERVICE 470
DISPLAY SVSTATUS 485
STOPREQ parameter, DISPLAY CHSTATUS 359
storage class
alter 115
define 268
delete 309
display 471
rules for names of 8
STRSTPEV parameter
ALTER QMGR 75
DISPLAY QMGR 429
sub delete
delete 307
sub display
display 476
SUB parameter
DEFINE TOPIC 286
DISPLAY SUB 480
DISPLAY TOPIC 496
Sub status parameters
DISPLAY TPSTATUS 501
SUBID parameter
ALTER SUB 119
DELETE SUB 308
DISPLAY SUB 465, 480
SUBID parameter, DISPLAY CONN 387
SUBLEVEL parameter
ALTER SUB 119
DEFINE SUB 276
SUBNAME parameter, DISPLAY CONN 388
SUBSCOPE parameter
ALTER SUB 119
DEFINE SUB 276
DEFINE TOPIC 286
DISPLAY SUB 481
DISPLAY TOPIC 496
subscription
define 273
SUBSTATE parameter, DISPLAY CHSTATUS 359
SUBTYPE parameter
DISPLAY SBSTATUS 464
DISPLAY SUB 465, 481
SUMMARY parameter
DISPLAY SUB 478
SUSPEND parameter, DISPLAY CLUSQMGR 370
SUSPEND QMGR command 596
SWITCHES parameter, DISPLAY SECURITY 467

SYNCPT parameter, DISPLAY QMGR 429
system
display 486
set 558
SYSTEM parameter
DISPLAY QMGR 420

T

TARGET parameter
DISPLAY QUEUE 460
TARGTYPE parameter
DEFINE queues 109, 261
DISPLAY QUEUE 454, 460
TASKNO parameter, DISPLAY CONN 383
TASKNO parameter, DISPLAY QSTATUS 445
TCPCHL parameter
ALTER QMGR 75
DISPLAY QMGR 429
TCPKEEP parameter
ALTER QMGR 76
DISPLAY QMGR 429
TCPNAME parameter
ALTER QMGR 76
DISPLAY QMGR 429
TCPSTACK parameter
ALTER QMGR 76
DISPLAY QMGR 429
TDATA parameter, START TRACE 577
thread
display information about 488
resolving in-doubt manually 544
THREAD parameter
DISPLAY THREAD 488
TID parameter, DISPLAY CONN 383
TID parameter, DISPLAY QSTATUS 445
TIME parameter, ARCHIVE LOG 126
TIMEOUT parameter
ALTER SECURITY 113
DISPLAY SECURITY 467
TNO parameter
ALTER TRACE 124
DISPLAY TRACE 506
STOP TRACE 595
topic
alter 121
display information about 491
topic definition
define 278
TOPIC parameter
DEFINE TOPIC 280
DELETE TOPIC 311
DISPLAY TOPIC 492
Topic status parameters
DISPLAY TPSTATUS 500
topic, deleting 311
TOPICOBJ parameter
ALT SUB 119
DEFINE SUB 276
DISPLAY SUB 481
TOPICSTR parameter
ALT SUB 119
DEFINE SUB 276
DEFINE TOPIC 286
TOPICSTR parameter *(continued)*
DISPLAY SUB 481
DISPLAY TOPIC 496
TOPICSTR parameter, DISPLAY CONN 388
TPIPE parameter
DISPLAY QUEUE 460
RESET TPIPE 539
TPNAME parameter
DEFINE CHANNEL 206
DEFINE LISTENER 211
DISPLAY CHANNEL 340
DISPLAY CLUSQMGR 373
DISPLAY LISTENER 393
DISPLAY LSSTATUS 399
TRACE parameter, DISPLAY TRACE 505
TRACTBL parameter, SET SYSTEM 559
TRANSID parameter, DISPLAY CONN 383
TRANSID parameter, DISPLAY QSTATUS 445
TRAXSTR parameter
ALTER QMGR 76
DISPLAY QMGR 429
TRAXTBL parameter
ALTER QMGR 77
DISPLAY QMGR 429
TREELIFE parameter
ALTER QMGR 77
DISPLAY QMGR 430
TRIGDATA parameter
DEFINE queues 109, 262
DISPLAY QUEUE 460
TRIGDPTH parameter
DEFINE queues 109, 262
DISPLAY QUEUE 461
TRIGGER parameter
DEFINE queues 110, 262
DISPLAY QUEUE 461
TRIGINT parameter
ALTER QMGR 77
DISPLAY QMGR 430
TRIGMPRI parameter
DEFINE queues 110, 262
DISPLAY QUEUE 461
TRIGTYPE parameter
DEFINE queues 110, 262
DISPLAY QUEUE 461
TRPTYPE parameter
DEFINE CHANNEL 206
DEFINE LISTENER 211
DISPLAY CHANNEL 340
DISPLAY CLUSQMGR 373
DISPLAY LISTENER 391
DISPLAY LSSTATUS 399
TSTAMP parameter, SET ARCHIVE 553
TYPE parameter
DISPLAY CHANNEL 333
DISPLAY CONN 380
DISPLAY PUBSUB 410, 411
DISPLAY QSTATUS 438
DISPLAY QUEUE 454
DISPLAY THREAD 489
DISPLAY TOPIC 495
DISPLAY TPSTATUS 500
DISPLAY USAGE 508
TYPE parameter *(continued)*
MOVE QLOCAL 510
RECOVER CFSTRUCT 517
REFRESH QMGR 523
REFRESH SECURITY 527
RESET QMGR 535

U

UNCOM parameter, DISPLAY QSTATUS 440
UNIT parameter, SET ARCHIVE 554
unit-of-work ID, display 488
UNIT2 parameter, SET ARCHIVE 554
UOWLOG parameter, DISPLAY CONN 383
UOWLOGDA parameter
DISPLAY CONN 383
UOWLOGTI parameter
DISPLAY CONN 383
UOWSTATE parameter
DISPLAY CONN 383
UOWSTDA parameter
DISPLAY CONN 384
UOWSTTI parameter
DISPLAY CONN 384
URID parameter, DISPLAY QSTATUS 445
URTYPE parameter
DISPLAY CONN 384
DISPLAY QSTATUS 445
USAGE parameter
DEFINE queues 110, 263
DISPLAY QUEUE 461
usage, page set, display 507
USERDATA parameter
ALT SUB 119
DEFINE PROCESS 226
DEFINE SUB 276
DISPLAY PROCESS 409
DISPLAY SUB 481
USERID parameter
DEFINE CHANNEL 206
DISPLAY CHANNEL 340
DISPLAY CLUSQMGR 373
DISPLAY CONN 384
DISPLAY QSTATUS 445
DISPLAY TRACE 506
START TRACE 577
STOP TRACE 595
Using commands in z/OS 4
using commands, rules for 1

V

VARUSER option
ALT SUB 119
DEFINE SUB 276
DISPLAY SUB 481

W

WAIT parameter, ARCHIVE LOG 126
WHERE parameter
DISPLAY AUTHINFO 316
DISPLAY CFSTATUS 319

WHERE parameter *(continued)*
DISPLAY CFSTRUCT 325
DISPLAY CHANNEL 330
DISPLAY CHSTATUS 348
DISPLAY CLUSQMGR 367
DISPLAY CONN 378
DISPLAY LISTENER 391
DISPLAY LSSTATUS 397
DISPLAY NAMELIST 402
DISPLAY PROCESS 407
DISPLAY QSTATUS 436
DISPLAY QUEUE 450
DISPLAY SERVICE 468
DISPLAY STGCLASS 472
DISPLAY SVSTATUS 483
DISPLAY TOPIC 492
DISPLAY TPSTATUS 499
WILDCARD parameter
DEFINE TOPIC 287
DISPLAY TOPIC 496
WRTHRSH parameter, SET LOG 557
WSCHEMA parameter
ALT SUB 119
DEFINE SUB 276
DISPLAY SUB 481

X

XBATCHSZ parameter, DISPLAY
CHSTATUS 360
XCFGNAME parameter
DEFINE STGCLASS 271
DISPLAY STGCLASS 474
RESET TPIPE 540
XCFMNAME parameter
DEFINE STGCLASS 271
DISPLAY STGCLASS 475
RESET TPIPE 540
XMITQ parameter
DEFINE CHANNEL 207
DEFINE QREMOTE 111, 263
DISPLAY CHANNEL 340
DISPLAY CHSTATUS 351
DISPLAY QUEUE 461
XQMSGSA parameter, DISPLAY
CHSTATUS 361
XQTIME parameter, DISPLAY
CHSTATUS 361

Z

z/OS trace
alter events being traced 123
display list of active traces 504
start 573
stop 592
z/OS, using commands in 4

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.

© Copyright IBM Corp. 1999, 2008

SC34-6941-00

Spine information:

WebSphere MQ Script (MQSC) Command Reference Version 7.0