Message Advisor for IMS
Reference Manual

Supporting

Version 1.7 of Message Advisor for IMS

December 2015
Contacting BMC Software

Several methods are available for contacting BMC Software.

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

<table>
<thead>
<tr>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC SOFTWARE INC</td>
<td>1 713 918 8800</td>
<td>1 713 918 8000</td>
</tr>
<tr>
<td>2103 CITYWEST BLVD</td>
<td>or</td>
<td></td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827 USA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>United States and Canada</td>
<td>or</td>
<td></td>
</tr>
<tr>
<td>Telephone</td>
<td>+01 713 918 8800</td>
<td></td>
</tr>
<tr>
<td>Fax</td>
<td>+01 713 918 8000</td>
<td></td>
</tr>
</tbody>
</table>

Outside United States and Canada

<table>
<thead>
<tr>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>+01 713 918 8800</td>
<td>+01 713 918 8000</td>
</tr>
</tbody>
</table>

© Copyright 2001-2016 BMC Software, Inc.

BMC, BMC Software, the BMC logo, the BMC Software logo, and other BMC marks are the exclusive properties of BMC Software, Inc. and are registered or may be registered in the U.S. and in other countries. BladeLogic and other BladeLogic marks are the exclusive properties of BladeLogic, Inc. and are registered or may be registered in the U.S. and in other countries. All other trademarks or registered trademarks are the property of their respective owners.

IBM, IMS, MQSeries, MVS, RACF, VTAM, and VTF are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

The information included in this documentation is the proprietary and confidential information of BMC Software, Inc., its affiliates, or licensors. Your use of this information is subject to the terms and conditions of the applicable End User License agreement for the product and to the proprietary and restricted rights notices included in the product documentation.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2103 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.
Customer support

Support website
You can obtain technical support from BMC 24 hours a day, 7 days a week at http://www.bmc.com/support. From this website, you can:

■ Read overviews about support services and programs that BMC offers
■ Find the most current information about BMC products
■ Search a database for problems similar to yours and possible solutions
■ Order or download product documentation
■ Download products and maintenance
■ Report a problem or ask a question
■ Subscribe to receive proactive e-mail alerts
■ Find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail
In the United States and Canada, if you need technical support and do not have access to the web, call 1 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter SupID:yourSupportContractID, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC
Have the following information available so that Customer Support can begin working on your issue immediately:

■ Product information
  — Product name
  — Product version (release number)
  — License number and password (trial or permanent)
■ Operating system and environment information
  — Machine type
  — Operating system type, version, and service pack or other maintenance level such as PUT or PTF
  — System hardware configuration
  — Serial numbers
  — Related software (database, application, and communication) including type, version, and service pack or maintenance level
■ Sequence of events leading to the problem
■ Commands and options that you used
■ Messages received (and the time and date that you received them)
  — Product error messages
  — Messages from the operating system
  — Messages from related software
License key and password information

If you have questions about your license key or password, contact Customer Support through one of the following methods:

- Send an e-mail message to customer_support@bmc.com. (In the Subject line, enter SupID:yourSupportContractID, such as SupID:12345.)
- In the United States and Canada, call 1 800 537 1813. Outside the United States and Canada, contact your local support center for assistance.
Chapter 3  QPF_LIST command
Overview  .................................................................63
Display the QPF Problem List ..................................63
Display current QPF options  ....................................65
QPF_LIST command set syntax  ............................65
Sample QPF_LIST command set  ............................65
QPF_LIST command set description  .......................66
ACTIVE keyword .....................................................67
IMSId keyword ......................................................67
ORIGins keyword ....................................................67
PROBlems keyword ...............................................68
SKIP_ORIGins keyword .........................................68
SKIP_PROBlems keyword .......................................68
TYPE keyword ......................................................69

Chapter 4  QPF_OPTIONS command
Overview  .................................................................71
QPF_OPTIONS subcommands and statements  ..........71
QPF_OPTIONS command set syntax  .......................73
Sample QPF_OPTIONS command sets ...................79
Activate QPF_OPTIONS command set ....................80
Test case command set ..........................................81
Set QPF_OPTIONS command set  ...........................85
QPF_OPTIONS command set description  ...............86
General keywords ..................................................86
MONITOR_PHASE keywords ................................96
PROTECT_PHASE keywords ................................102
OVERFLOW_PHASE keywords .............................132
UNLoad_dsn keywords .........................................138

Chapter 5  CUSTOMIZE command
Overview  .............................................................147
Virtual LTERM creation feature  ............................148
Uppercase-only feature .........................................148
CUSTOMIZE command set syntax  .......................149
Sample CUSTOMIZE command sets ......................151
CUSTOMIZE command set description ..................152
Chapter 6  DEQUEUE command  177
Overview ............................................................................................................................ 177
Assigning conditional steps ............................................................................................ 178
DEQUEUE command set syntax .................................................................................... 179
Sample DEQUEUE command sets ................................................................................. 180
DEQUEUE command set description ............................................................................. 182
CONVersations keyword ..................................................................................... 183
FORCE keyword .................................................................................................... 185
 IMSid keyword ....................................................................................................... 187
LOG keyword ......................................................................................................... 187
MODE keyword ..................................................................................................... 188
RATE keyword ....................................................................................................... 188
REPORT/PRINT keyword ....................................................................................... 188
SCOpe keyword ...................................................................................................... 189
TYPE keyword ....................................................................................................... 190

Chapter 7  DISPLAY command  193
Overview ............................................................................................................................ 193
Assigning conditional steps ............................................................................................ 195
DISPLAY command set syntax ....................................................................................... 195
Sample DISPLAY command sets .................................................................................... 198
DISPLAY TYPE=DESTINATION command set description ..................................... 200
 IMSid keyword ....................................................................................................... 200
LIMit keyword ....................................................................................................... 201
RATE keyword ....................................................................................................... 201
SCOpe keyword ...................................................................................................... 201
TYPE keyword ....................................................................................................... 203
DISPLAY TYPE=DEST_QUEUES command set description ..................................... 204
 IMSid keyword ....................................................................................................... 205
LIMit keyword ....................................................................................................... 205
LIST_Queues LQ keyword ................................................................................... 205
RATE keyword ....................................................................................................... 206
SCOpe keyword ...................................................................................................... 206
SORT_by keyword ................................................................................................. 208
TYPE keyword ....................................................................................................... 209
DISPLAY TYPE=MESSAGE command set description .............................................. 210
FORMat keyword .................................................................................................. 210

Contents  7

IMS_Options subcommand keywords ................................................................. 153
LIST_Options subcommand keywords ............................................................... 167
SERVER_Options subcommand keywords ......................................................... 168
INCORE keyword and REQUEUE performance .............................................. 230
Assigning conditional steps ........................................................................ 231
REQUEUE command set syntax .................................................................. 232
Sample REQUEUE command sets ................................................................ 234
REQUEUE command set description ............................................................. 237
CHKPT keyword .......................................................................................... 238
COMpress keyword .................................................................................... 239
CONVersations keyword ............................................................................ 239
DISCard keyword ....................................................................................... 240
ERROR keyword ........................................................................................ 241
EXIT keyword ........................................................................................... 242
IMSid keyword .......................................................................................... 243
INCore keyword ....................................................................................... 243
INTERNAL_PREFIX keyword ....................................................................... 244
MAXDATA keyword ................................................................................... 244
MAXMSGS keyword .................................................................................. 244
MOde keyword ........................................................................................ 245
OLDS keyword ......................................................................................... 246
RATE keyword .......................................................................................... 246
REPORT/PRint keyword ............................................................................. 247
REQ_PROMPT/CHKPT_PRompt keyword ..................................................... 248
SCRap/SCP keyword ................................................................................ 249
TYPE keyword .......................................................................................... 250
USER_PREFIX keyword ............................................................................. 252
VALIDATE_SRC keyword .......................................................................... 253
ZERO_time keyword ................................................................................ 253
CHANGE subcommand ............................................................................... 254
GROUP subcommand ................................................................................. 270
INPUT subcommand .................................................................................. 272
SPILL subcommand .................................................................................. 275

Chapter 10        UNLOAD command ............................................................... 281
Overview ....................................................................................................... 281
Assigning conditional steps ......................................................................... 282
UNLOAD command set syntax ..................................................................... 283
Sample UNLOAD command sets .................................................................. 284
UNLOAD command set description ............................................................. 286
CONVersations keyword ............................................................................. 286
FORCE keyword ....................................................................................... 288
IMSid keyword .......................................................................................... 289
RATE keyword .......................................................................................... 290
Contents

1. COMMITMODE keyword ................................................................. 323
2. CONVID keyword ................................................................. 323
3. DATA keyword ................................................................. 326
4. DATASIZE keyword ............................................................ 326
5. DBD keyword ................................................................. 327
6. DEADQ keyword ............................................................... 328
7. DESTination keyword .......................................................... 328
8. DESType/DTYPe keyword .................................................. 329
9. DRm keyword ................................................................. 332
10. DYNAmic/VIRTual/ETO/VTF/CNT-E/CNT-V keyword .......... 333
11. ENTRYID keyword ............................................................. 334
12. KEY keyword ................................................................. 334
13. HELD keyword ............................................................... 335
14. LABEL keyword .............................................................. 335
15. LTERM keyword ............................................................ 336
16. LUname keyword ............................................................. 336
17. MFSNAME keyword .......................................................... 337
18. MSGTYPE keyword .......................................................... 338
19. MSNAME keyword ........................................................... 338
20. NEWER_than keyword ...................................................... 340
21. NODENAME keyword ....................................................... 340
22. OFFSET keyword ............................................................. 341
23. OIMS keyword ............................................................... 341
24. OLDER_than keyword ...................................................... 342
25. ORIGIN keyword ............................................................. 342
26. ORGQNAME keyword ....................................................... 343
27. ORGQMANAGER keyword ............................................... 344
28. ORGTPIPE keyword ........................................................ 344
29. ORGTMEMBER keyword .................................................. 345
30. OTMAQ keyword ............................................................. 346
31. OTOKEN keyword ........................................................... 346
32. PIMS keyword ............................................................... 347
33. PSB keyword ................................................................. 347
34. PTOKEN keyword ............................................................ 349
35. QPOSITION keyword ..................................................... 349
36. QPOSITION keyword and relative position ......................... 350
37. QTYPE keyword ............................................................. 351
38. Queue keyword .............................................................. 356
39. RCNT keyword .............................................................. 357
40. REGION keyword ........................................................... 358
41. REGION_ID keyword ........................................................ 359
Chapter 14  Message Advisor shared queues command sets 371

DEQUEUE command for shared queues ................................................................. 371
  Sample SQ DEQUEUE command sets ............................................................... 371
  SQ DEQUEUE command set description ......................................................... 372

DISPLAY command for shared queues .............................................................. 374
  Sample SQ DISPLAY command sets ............................................................... 374
  SQ DISPLAY TYPE=DESTINATION command set description ...................... 376
  SQ DISPLAY TYPE=DEST_QUEUES command set description .................... 377
  SQ DISPLAY TYPE=MESSAGE command set description ............................. 379
  SQ DISPLAY TYPE=RECORD command set description .............................. 381
  SQ DISPLAY TYPE=STATISTICS command set description ........................ 383
  SQ DISPLAY TYPE=ANALYZER command set description ......................... 383

REQUEUE command for shared queues .............................................................. 384
  Sample SQ REQUEUE command sets ............................................................ 384

UNLOAD command for shared queues ............................................................... 386
  Sample SQ UNLOAD command sets ............................................................. 386
  SQ UNLOAD command set description ........................................................ 387

SELECT and REJECT subcommands for shared queues ................................. 389
  AFFINITY keyword ...................................................................................... 391
  ENTRYID keyword ...................................................................................... 393
  KEY keyword ............................................................................................ 393
  QTYPE keyword ....................................................................................... 394
About this book

This book describes the commands available with the Message Advisor for IMS solution. The information presented is for system programmers, database administrators, and other individuals who install, operate, or maintain an IMS.

Message Advisor comprises two components: the Queue Protection Facility (QPF) component and the Queue Manager utilities (QMR) component. Through these components, Message Advisor provides a comprehensive solution to the problems associated with managing both local and shared IMS message queues. The Message Advisor solution improves IMS availability by helping you to maintain IMS message queue integrity.

The QPF component automatically monitors and manages IMS local message queues in order to prevent IMS outages (shutdowns and/or abends) caused by message queue overflow. When QPF identifies a problem, it automatically executes corrective action according to criteria you define.

The QMR component allows you to perform the following message queue management tasks for both local and shared queues:

- create and view reports on message queue statistics
- simplify requeueing messages to the message queues
- dequeue unwanted messages
- unload messages for later use

Message Advisor also allows you to perform the following diagnostic tasks for shared queues:

- break down statistics by queue type
- report on a variety of IMS data errors
- print or view online the contents of the primary and overflow structures (which comprise the global queues)
Like most BMC documentation, this book is available in printed and online formats. To request printed books or to view online books and notices (such as release notes and technical bulletins), see the support website at http://www.bmc.com/support.

**Note**

Online books are formatted as PDF or HTML files. To view, print, or copy PDF books, use the free Adobe Reader from Adobe Systems. If your product installation does not install the reader, you can obtain the reader at http://www.adobe.com.

The software also offers online Help. To access Help, press **F1** within any product or click the **Help** button in graphical user interfaces (GUIs).

### Related publications

From the BMC Support Central website, you can use the following methods to access related publications that support your product or solution:


- View Quick Course videos (short overviews of selected product concepts, tasks, or features), which are available from the following locations:
  - Documentation Center (primary center and secured center)
  - Support Central (at http://www.bmc.com/support/mainframe-demonstrations)
  - BMC Mainframe YouTube channel (https://www.youtube.com/user/BMCSoftwareMainframe)


Products with online interfaces also offer online Help via the **F1** key or, for graphical user interfaces (GUIs), via a **Help** button.

**Tip**

If you prefer hardcopy documentation, you can order it from your BMC sales representative or from Support Central. Also, from Support Central you can subscribe to receive proactive e-mail alerts when BMC issues notices.
Conventions

This section provides examples of the conventions used in this book and explains how to read syntax statements.

General conventions

This book uses the following general conventions:

<table>
<thead>
<tr>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>information that you are instructed to type</td>
<td>Type DATABASE in the designated field.</td>
</tr>
<tr>
<td>specific (standard) keyboard key names</td>
<td>Press Enter.</td>
</tr>
<tr>
<td>field names</td>
<td>Type the letter of the action that you want to perform in the Act field</td>
</tr>
<tr>
<td>text on a panel</td>
<td>The LGMG is 35% used, and MONITOR is highlighted.</td>
</tr>
<tr>
<td>nonspecific key names, option names</td>
<td>Use the HELP function key.</td>
</tr>
<tr>
<td>MVS calls, commands, keywords, parameters, reserved words</td>
<td>Use the /DISPLAY command to display IMS resources and status.</td>
</tr>
<tr>
<td>control statements, code examples, syntax statements</td>
<td>//SYSTSIN DD * &lt;==REQUIRED</td>
</tr>
<tr>
<td>system messages</td>
<td>BMC43267I MESSAGE ADVISOR READING INPUT LOG(S), IMSID=R61P</td>
</tr>
<tr>
<td>screen text</td>
<td>PROFILE MSGID</td>
</tr>
<tr>
<td>emphasized words, new terms</td>
<td>The instructions that you give to the software are called commands.</td>
</tr>
<tr>
<td>variables</td>
<td>In this message, the variable file_name represents the file that caused the error.</td>
</tr>
</tbody>
</table>

This book uses the following types of special text:

--- Note ---

Notes contain important information that you should consider.
**WARNING**

Warnings alert you to situations that could cause problems, such as loss of data, if you do not follow instructions carefully.

**Tip**

Tips contain useful information that can improve product performance or that can make procedures easier to follow.

### IMS versions

The term IMS refers to all supported versions and releases of IMS. The specific product name, version, and release numbers are noted only when this information is significant.
Introduction

This chapter provides an overview of all commands that are available with the Message Advisor for IMS solution, then explains the command set structure and methods for building and executing a command set.

Overview

The Message Advisor solution comprises the following components:

- Queue Protection Facility (QPF) component
- Queue Manager utilities (QMR) component

The QPF component automatically monitors and manages IMS local message queues to prevent IMS outages (shutdowns and/or abends) caused by message queue overflow. The QMR component lets you perform message queue management tasks for local and shared queues. Message Advisor also includes some diagnostic functions that are specific to shared queues.

Message Advisor QPF commands

The following commands are available with the Message Advisor QPF component:

- QPF_ACTION
- QPF_LIST
- QPF_OPTIONS

You must build the QPF commands outside of the Message Advisor Interactive System Productivity Facility (ISPF) interface panels. You can use ISPF or a similar data set editing facility to build the QPF commands.
**QPF_ACTION command**

You can use the QPF_ACTION command to initiate any of the following manual QPF actions:

- **ABEND** terminates an application the next time it tries to insert a message.
- **DEQUEUE** deletes queued messages for a destination.
- **FAIL** issues an A7 status code to the application the next time the IMS region tries to insert a message.
- **IPOST** restarts an IMS region or input device that QPF has placed in a wait.
- **IWAIT** waits a region or input device the next time the region or device tries to insert a message.
- **STOP** stops a destination, region, or input device (destinations stop immediately; regions and input devices stop the next time they try to insert a message).
- **UNLOAD_DEQUEUE** unloads and dequeues the queued messages for a destination.

For more information, see “QPF_ACTION command” on page 47, and see the Message Advisor for IMS User Guide.

**QPF_LIST command**

You can use the QPF_LIST command to display the current message queue problem list.

QPF generates a list of message queue problems (the QPF Problem List) when it reaches the threshold you define for the Monitor Phase.

For more information, see “QPF_LIST command” on page 63, and see the Message Advisor for IMS User Guide.

**QPF_OPTIONS command**

You can use the QPF_OPTIONS command to set, replace, list, or unload QPF options.

You can define options that determine how QPF performs the following tasks:

- monitors an IMS system
- sends warning messages
takes action to prevent an IMS outage

For more information, see “QPF_OPTIONS command” on page 71, and see the Message Advisor for IMS User Guide.

QPF WTOR commands

The QPF component provides operator commands that you can issue only in response to a QPF WTOR (write to operator with reply).

You can issue these commands to accomplish a variety of tasks quickly.

Message Advisor uses the following types of QPF WTOR commands:

■ Initialization WTOR commands
  You can use Initialization WTOR commands to respond to a WTOR that Message Advisor issues because of a problem loading the QPF options at IMS start-up. If there is a problem, Message Advisor issues the following message:

  BMC43169A Message Advisor QPF FOR IMSID=imsid - INVALID QPFOPTS FILE - REPLY 'RETRY' OR 'IGNORE'

  You can issue the following Initialization WTOR commands in response to Initialization WTOR message BMC43169A:
  — IGNORE continues processing without QPF being active.
  — RETRY reloads the QPF options from the QPFOPTS file.
  — SVCDUMP takes an immediate SVC dump.

  Note
  Use the SVCDUMP command only for diagnostic purposes at the request of BMC Software Product Support.

  Note
  IMS restart is not dependent on your reply to the Initialization WTOR. Restart continues without waiting for QPF to initialize.

■ Processing WTOR commands
  You can use Processing WTOR commands to respond to a WTOR that Message Advisor issues when a phase starts, if you specify WTOR=YES for that phase in the
QPF_OPTIONS command set. If you specify WTOR=YES, you will receive the following message:

```
BMC43168A Message Advisor QPF FOR IMSID=imsid - REPLY 'LIST' OR 'HELP'
```

You can issue the following Processing WTOR commands in response to Processing WTOR message BMC43168A:

— Operator commands:

— HELP displays a list of operator commands.

— IGNORE stops issuing this WTOR; you can issue the IGNORE command at any time.

— SVCDUMP takes an immediate SVC dump, and shuts down QPF; you can issue the SVCDUMP command at any time.

— LIST nn mm lists the worst nn problem destinations and/or regions on the IMS queues (default is 20), and lists the mm most active origins in the IMS system (default is 5), where nn and mm are numbers from 0 to 99.

You will find it convenient to use the Processing WTOR LIST command to display a problem list in response to a Processing WTOR message if Time Sharing Option (TSO) is not available or if you are at the system console.

— Action commands:

— ABEND abends a region (BMP or MPP).

— DEQUEUE dequeues all messages for a destination.

— FAIL returns a bad status code to an application.

— IPOST releases a region or an input device from a QPF wait.

— IPOST ALL releases all regions and input devices from QPF waits.

— IWAIT places a region or an input device in a wait.

— STOP stops a destination, region, or input device.

— UNLOAD unloads and dequeues all messages for a destination.

You will find it convenient to issue a Processing WTOR Action command in response to a Processing WTOR message if TSO is not available or if you are at the system console.

All Processing WTOR Action commands except the IPOST ALL command accept the NAME and TYPE positional parameters. The NAME parameter is
required; the TYPE parameter is optional, but BMC Software recommends using the parameter. The TYPE parameter indicates the type of IMS entity for which you are requesting the action. The NAME parameter indicates the name of the IMS entity for which you are requesting the action. Masking is not allowed for the NAME parameter. You can specify the TYPE and NAME parameters for destinations, regions, and input devices.

**Note**
You should usually obtain the name from the QPF Problem List (displayed by issuing the LIST command), but you are not required to do so.

Table 1 on page 23 lists the TYPE and NAME values allowed for destinations.

### Table 1: TYPE and NAME positional parameters for destinations

<table>
<thead>
<tr>
<th>TYPE Parameter</th>
<th>NAME Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUNAME</td>
<td>luname</td>
</tr>
<tr>
<td>CNT-V, CNT-E, CNT, DYNAMIC, STATIC, VIRTUAL</td>
<td>lterm name</td>
</tr>
<tr>
<td>DEADQ</td>
<td>lterm name</td>
</tr>
<tr>
<td>MSNAME</td>
<td>msname</td>
</tr>
<tr>
<td>RSMB</td>
<td>remote SMB name</td>
</tr>
<tr>
<td>SMB</td>
<td>SMB name</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>system LTERM name</td>
</tr>
<tr>
<td>TPIPE</td>
<td>OTMA TPIPE name</td>
</tr>
<tr>
<td>VSPCNT</td>
<td>LU6 subpool name</td>
</tr>
</tbody>
</table>

Note: If more than one tpname is associated with the luname, the command will fail.

Table 2 on page 23 lists the TYPE and NAME values allowed for regions.

### Table 2: TYPE and NAME positional parameters for regions

<table>
<thead>
<tr>
<th>TYPE Parameter</th>
<th>NAME Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP-nnnn</td>
<td>job name</td>
</tr>
<tr>
<td>MPP-nnnn</td>
<td>job name</td>
</tr>
</tbody>
</table>
Table 3 on page 24 lists the TYPE and NAME values allowed for input devices.

### Table 3: TYPE and NAME positional parameters for input devices

<table>
<thead>
<tr>
<th>TYPE Parameter</th>
<th>NAME Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN-CLB</td>
<td>VTAM node name of the input device</td>
</tr>
<tr>
<td>IN-CNT</td>
<td>CNT or LTERM name that is assigned to the input device</td>
</tr>
<tr>
<td>IN-LUN</td>
<td>APPC LUNAME that is generating input</td>
</tr>
<tr>
<td>IN-MSN</td>
<td>MSNAME that is generating input</td>
</tr>
<tr>
<td>IN-TPI</td>
<td>OTMA TPIPE that is generating input</td>
</tr>
<tr>
<td>QMR</td>
<td>QMR job that is requeueing messages</td>
</tr>
</tbody>
</table>

### Message Advisor QMR commands

The following commands are available with the Message Advisor QMR component:

- CUSTOMIZE
- DEQUEUE
- DISPLAY
- IMSCMD
- REQUEUE
- UNLOAD

You can build DEQUEUE, DISPLAY, REQUEUE, and UNLOAD commands interactively through the Message Advisor ISPF interface panels. You can also use
ISPF or a similar data set editing facility to build these commands outside of the Message Advisor ISPF interface panels.

You cannot build the CUSTOMIZE and IMSCMD commands interactively. You must use normal data set editing techniques (ISPF or a similar data set editing facility) to build these commands. However, you may use the ISPF interface to directly customize Message Advisor or to issue IMS commands interactively.

**CUSTOMIZE command**

You can use the CUSTOMIZE command to perform the following tasks:

- set or change Message Advisor Server options
  - define Server name
  - define Virtual Telecommunications Access Method (VTAM) parameters (for example, APPLID)
- set or change options that define the associated IMS system
  - define auto requeue parameters
  - define data sets for Message Advisor use
  - set VIRTUAL_CREATE option
  - set WTO_CMDS option

For more information, see “CUSTOMIZE command” on page 147 and see the installation guide.

**DEQUEUE command**

You can use the DEQUEUE command to perform the following tasks:

- dequeue all messages for a destination
- dequeue messages by destination type (for example, CNT, MSNAME, VSPCNT)
- dequeue a specific message from a destination
- dequeue by specific queue number
- dequeue transactions (SMBs)
- validate a dequeue request before dequeueing messages
exit conversations and delete conversational messages

unload messages that satisfy all selection criteria, then dequeue the unloaded messages (MODE=UNLOAD_DEQUEUE)

For more information, see “DEQUEUE command” on page 177 and see the Message Advisor for IMS User Guide.

**DISPLAY command**

You can use the DISPLAY command to perform the following tasks:

- display message queue statistics, including the following:
  - number of messages that have been sent to each message queue
  - current queue usage
  - 10 destinations with the largest queue counts
- display the number of messages in the message queues
- display the number of messages that are queued to a destination
- display hexadecimal and formatted message contents for an individual message
- display the number of messages that are queued to destinations meeting your selection criteria
- identify top queue space users sorted by the following criteria:
  - percentage of short message queue
  - percentage of long message queue
  - number of messages
  - number of segments
  - queue count
- search for and display messages that meet certain selection criteria, such as messages containing a specific text string

For more information, see the Message Advisor for IMS User Guide.
**IMSCMD command**

The IMSCMD command lets you submit IMS commands through Message Advisor.

*Note*

Use IMSCMD to issue IMS commands. IMS commands can be coordinated with Message Advisor commands.

For more information, see “IMSCMD and PLEXCMD commands” on page 221 and see the Message Advisor for IMS User Guide.

**REQUEUE command**

You can use the REQUEUE command to perform the following tasks:

- requeue all messages after a cold start following a normal shutdown (DUMPQ/PURGE)
- requeue all messages after a cold start following an abnormal IMS shutdown and emergency restart (/ERE) failure
- requeue messages that were previously processed so that they can be processed again (normally done for testing purposes)
- requeue messages that were previously unloaded with the UNLOAD command
- extract messages for later requeue
- requeue messages that were previously extracted with the REQUEUE MODE=EXTRACT command
- alter data upon requeue with the CHANGE subcommand

For more information, see “REQUEUE command” on page 227 and see the Message Advisor for IMS User Guide.

**UNLOAD command**

You can use the UNLOAD command to unload messages from active IMS message queues to a data set, based on criteria you specify.

You can unload the following items:

- all messages for a destination
- messages by destination type (for example, CNT, MSNAME, VSPCNT)
a specific message from a destination
messages by specific queue number
transactions (SMBs)
messages assigned to DEADQ destinations
messages that meet certain selection criteria, such as messages containing a specific text string

For more information, see “UNLOAD command” on page 281 and see the Message Advisor for IMS User Guide.

**QMR WTOR commands**

The QMR component also provides operator commands that you can issue only in response to a Message Advisor Server WTOR:

- **CANCEL** terminates any QMR task and cancels all subsequent command sets.

  **Note**
  There is no confirmation to a WTOR command, and changes in process are not saved.

  — The TASK keyword indicates the number of the task to be canceled, in `nnnnnnnn` format (where `nnnnnnnn` is a number that you specify).

  — The USERID keyword indicates the user ID of the user who initiated the task being canceled, in `cccccccc` format (where `cccccccc` is a user ID that you specify).

  **Note**
  The USERID keyword is required for the CANCEL command.

  **Note**
  You can use the STATUS command to display a list of tasks.

- **HELP** displays a list of WTOR commands.

- **HIDE** causes the Message Advisor Server to stop issuing the WTOR.

  **Note**
  You can use `IGNORE` as an alias for HIDE.

- **F <server>,SHOW** displays the Message Advisor Server WTOR after it has been stopped with the HIDE command.
Note

You can use UNHIDE as an alias for SHOW.

- STATUS displays sessions that are in process (active and inactive).
- STOP stops the Message Advisor Server after completing tasks that are in process.

You can also issue QMR base command sets to the Message Advisor Server by typing the required commands, subcommands, keywords, and parameters directly into the Server’s outstanding WTOR. Message Advisor continues accepting input from the outstanding WTOR until it receives an END command, at which point the Server validates and executes the command set.

QMR WTOR sample command set

The following figure shows a command set for cancelling a QMR task by using an operator command.

The command will cancel task number 2 (TASK) for user ID R61P (USERID).

**Figure 1: Command set: Cancel a QMR task by using an operator command**

```
CANCEL USERID=R61P,TASK=2
END
```

Command sets

You can define Message Advisor tasks by grouping commands, subcommands, keywords, and parameters to form command sets. A command set is a task within a request. A request consists of a command and one or more subcommands, keywords, and parameters, followed by the END command.

Message Advisor QPF command sets

Each QPF command set requires a single primary command and the END command.

You can repeat some QPF subcommands multiple times within a command set to select specific messages to process.

**QPF_ACTION command set**

The QPF_ACTION command set consists of the following items:
QPF_LIST command set

The QPF_LIST command set consists of the following items:

- Primary Command: QPF_LIST
- Subcommands/Statements: <none>
- Closing Command: END

QPF_OPTIONS command set

The QPF_OPTIONS command set consists of the following items:

- Primary Command: QPF_OPTIONS
- Subcommands/Statements:
  - MONITOR_PHASE
  - PROTECT_PHASE
  - ENFORCE statement
  - PROCESS statement
  - OVERFLOW_PHASE
  - UNLOAD_DSN
- Closing Command: END

**Note**

The ENFORCE and PROCESS statements apply only to actions that occur during the Protect Phase.
Message Advisor QMR command sets

You define QMR tasks in the same manner as you define QPF tasks.

CUSTOMIZE command set

The CUSTOMIZE command set consists of the following items:

■ Primary Command: CUSTOMIZE

■ Subcommands:
  — IMS_OPTIONS
  — LIST_OPTIONS
  — SERVER_OPTIONS

■ Closing Command: END

DEQUEUE command set

The DEQUEUE command set consists of the following items:

■ Primary Command: DEQUEUE

■ Subcommands:
  — INTERVAL
  — OUTPUT
  — REJECT
  — SELECT

■ Closing Command: END

DISPLAY command set

The DISPLAY command set consists of the following items:

■ Primary Command: DISPLAY

■ Subcommands:
IMSCMD command set

The IMSCMD command set consists of the following items:

- Primary Command: IMSCMD
- Subcommands: <none>
- Closing Command: END

REQUEUE command set

The REQUEUE command set consists of the following items:

- Primary Command: REQUEUE
- Subcommands:
  - CHANGE
  - EXTRACT
  - GROUP
  - INPUT
  - INTERVAL
  - REJECT
  - SCRAP
  - SELECT
- Closing Command: END

TRACE command set

The TRACE command set consists of the following items:
Primary Command: TRACE

Closing Command: END

--- Note ---
For information about diagnostic tools, see the installation guide.

UNLOAD command set

The UNLOAD command set consists of the following items:

- Primary Command: UNLOAD

- Subcommands:
  - INTERVAL
  - OUTPUT
  - REJECT
  - SELECT

- Closing Command: END

Building and executing a command set

Before you can execute a Message Advisor command, you must build a request for the function that you want to perform.

A request is a unit of work containing one or more command sets. A command set is a task within a request. Each command set consists of the following items:

- primary command

- one or more subcommands, keywords, and parameters

- END command

You may want to build several command sets and execute them as one request. You can accomplish multiple tasks with one request by using multiple command sets within a single request. For example, you can unload, dequeue, and requeue all messages as a single task with one request containing multiple command sets.
You can build a Message Advisor command set from scratch by typing the desired commands, subcommands, keywords, and parameters; or you can use a sample command set. A sample command set is a template that you can copy and modify to meet your needs, minimizing errors and saving the effort of keying in the required data. One or more sample command sets are available for each Message Advisor command.

Building a command set

You can use ISPF or a similar data set editing facility to create any Message Advisor command set. The Message Advisor ISPF interface contains a direct interface to ISPF EDIT for this purpose. The Message Advisor ISPF interface also contains structured data entry panels that let you build certain command sets by filling in the necessary values; you do not have to remember keywords or consult reference materials. You can build DEQUEUE, DISPLAY, REQUEUE, and UNLOAD commands for the QMR component in this manner.

Executing a command set

You can use the following methods to execute Message Advisor command sets:

- as a batch job
- from the Message Advisor ISPF interface panels in foreground or background
- through an outstanding WTOR

Note

Executing a Message Advisor command set through an outstanding WTOR can be tedious for complicated commands. If you issue a command set through a WTOR, you cannot save the command set for later use.

This chapter provides a summary of each method for building and executing command sets. For detailed information, as well as information about using the sample command sets, see the Message Advisor for IMS User Guide.

Interactive method

The Message Advisor ISPF interface provides structured data entry panels and pop-ups that you complete to produce, update, save, and execute command sets for the following QMR commands:
If you use a sample command set for one of these commands with the interface panels, the sample command set populates the interface panel fields. You can accept or modify the populated field values to meet your needs.

You can also use the Message Advisor ISPF interface panels to execute all types of command sets, interactively or by submitting a batch job.

Building and executing a command set includes the following tasks:

- allocating a request library
- selecting a sample command set
- making any necessary changes
- saving the command set to the request library
- executing the command set

For more information about performing these tasks, see the Message Advisor for IMS User Guide.

**Batch method**

You can choose either of the following approaches to build and save a Message Advisor command set through ISPF or a similar data set editing facility:

- Create a partitioned data set (PDS) member in the command set library and type the desired commands, subcommands, keywords, and parameters directly into the member.

  —*or*—

- Create a PDS member in the command set library, copy a sample command set provided by BMC Software into the PDS, and modify it to meet your needs.

The second approach minimizes errors and saves the effort of keying in the required data. The sample command sets are located in MAQSAMP.
Regardless of which approach you use to build the command set, the PDS is specified as SYSIN to the Message Advisor Batch Server job control language (JCL).

When you build the command set, you can execute it by submitting a batch job or by using the Message Advisor ISPF interface panels in foreground or background.

Note
All sample command sets in this reference manual are shown with the option of including the default values. For information about creating command sets without the defaults being shown, see the Message Advisor for IMS User Guide.

Building and executing a command set includes the following tasks:

- allocating a request library
- selecting a sample command set
- making any necessary changes
- saving the command set to the request library
- executing the command set

For more information about building a command set in ISPF or a similar data set editing facility, see the Message Advisor for IMS User Guide. You can submit a batch job to execute command sets built in this manner, or you can execute the command sets through the Message Advisor ISPF interface panels, interactively or as a batch job.

WTOR method

Message Advisor provides the following WTORs:

- Message Advisor Server WTOR
  You can issue command sets to the Message Advisor Server through the Server’s outstanding WTOR. The Message Advisor Server WTOR accepts command sets for the QPF and QMR components. If you issue a command set in this manner, you cannot save it for later use. Also, executing a command set in this manner can be tedious for complicated commands.
  To issue a command set to the Message Advisor Server, you must type the required commands, subcommands, keywords, and parameters directly at the Server’s outstanding WTOR. Message Advisor continues accepting input from the
outstanding WTOR until it receives an END; then the Server validates and executes the command set.

In addition, you can issue QMR operator commands to the Message Advisor Server through the Server’s outstanding WTOR.

- **QPF WTOR**

  QPF provides a group of operator commands that you can issue to the QPF WTOR.

  For normal processing, operator commands are issued through a WTOR, but command sets are not. If you cannot access TSO, you may want to issue a command set to the Message Advisor Server WTOR.

  For example, if you temporarily cannot build a command set through ISPF or the Message Advisor ISPF interface panels, you can issue a command set through an outstanding WTOR.

  You will find it useful to issue an operator command when you want to perform the following tasks:

  - check the status of a command set
    
    If you execute a command set that does not perform as expected, you can issue a **STATUS** command through an outstanding WTOR to determine the command set status.

  - stop Message Advisor
    
    To shut Message Advisor down, issue a **STOP** command to the outstanding Server WTOR or use the MVS STOP command.

---

### Command syntax considerations

When modifying an existing command set or creating a new one, consider the following:

- If you type an asterisk (*) in column 1, the entire line of text becomes a comment. If you type a semicolon (;) anywhere in a line, any text on the line following the semicolon becomes a comment.

- For a string to be continued across multiple lines, it must meet the following conditions:

  - The string must extend through the last valid column on the current line. For a standard 80-byte fixed-length record, the last valid column is column 71.

  - The first non-blank character on the next line must be the delimiter that you typed at the beginning of the string.
Message Advisor commands, subcommands, and keywords can begin in any column and are delimited with a comma. An equal sign (=) must follow keywords.

Command names, keywords, and undelimited values are not case-sensitive. Message Advisor converts all lowercase characters in command names, keywords, and undelimited values to uppercase before processing begins. Therefore, you can type commands, keywords, and undelimited values in lowercase, uppercase, or mixed case.

If a keyword value contains lowercase or special characters, you must delimit the value using one of the following methods:

— Enclose the value in single or double quotation marks (for example, KEYWORD='value'). The value may not contain the delimiters.

— Type the character C, followed by any special character (including quotation marks) to be used as a delimiter, followed by the string, followed by the delimiter (for example, KEYWORD=C/value/). The value may not contain the delimiters.

— Type the character X, followed by any special character (including quotation marks) to be used as a delimiter, followed by the value of the keyword (expressed in hexadecimal format), followed by the delimiter (for example, KEYWORD=X/E5C1D3E4C5/).

The LUNAME parameter accepts a network ID qualifier. The format is as follows:

LUNAME=netid.luname

The network ID and the LUNAME may contain mask characters where masks are allowed.

You can negate certain flag values by adding NO in front of keywords. For example, to process all destination types except APPC and OTMA, you can specify DESTYPE=(NOAPPC,NOOTMA).

You can include symbolic keywords when specifying values for the following items:

— the CUSTOMIZE command AUTO_REQ_CMD keyword

— QPF UNLOAD_DSN data set names

— all data set name keywords; all data set name keywords can contain the following symbolic values:

— %TIME converts to the current time of day, in hhmmss format.

— %TIMET converts to the current time of day, in hhmmssst format.
—%DATE converts to the current day of the year, in yyddd format.

—%DATE4 converts to the current day of the year, in yyyyddd format.

—%IMSID converts to the IMSID specified for the command.

—%USER converts to the current Resource Access Control Facility (RACF) user ID of the requestor.

The following considerations and restrictions apply when using symbolic keywords to specify values:

— A non-alphanumeric character must follow a symbolic keyword. If a percent symbol (%) that is not part of another symbolic keyword follows the keyword, the trailing percent symbol is removed and the following characters are concatenated to the symbolic value.

— More than one keyword may appear in a data set name.

— You cannot use keywords that begin with a number at the beginning of a qualifier, because qualifiers cannot start with a number.

— Specification of a data set name with symbolic keywords cannot exceed 44 characters. In addition, the maximum length of the data set name after symbolic substitution cannot exceed 44 characters.

Any violation of the 44-character length will result in syntax error message BMC43016.

Table 4 on page 39 lists examples of valid data set names that include symbolic keywords.

<table>
<thead>
<tr>
<th>Valid Data Set Name</th>
<th>Converts To...</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS.%IMSID.D%DATE.T%TIMET</td>
<td>IMS.IMSA.D95213.T1312232</td>
</tr>
<tr>
<td>IMS.%IMSID.D%DATE.TM%TIME.UNLOAD</td>
<td>IMS.IMSA.D95213.TM131223.UNLOAD</td>
</tr>
<tr>
<td>IMS.%IMSID.D%IDATE.T%ITIME.N%DSNNO</td>
<td>IMS.IMSA.D95213.T0601473.N000001</td>
</tr>
<tr>
<td>IMS.%IMSID%A.DEST.%DEST</td>
<td>IMS.IMSAA.DEST.LTRM0001</td>
</tr>
</tbody>
</table>

Table 5 on page 40 lists examples of invalid data set names that include symbolic keywords.
Table 5: Invalid data set names using symbolic keywords

<table>
<thead>
<tr>
<th>Invalid Data Set Name</th>
<th>Syntax Error Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS.%IMSIDIMS.D%DATE.T%TIME</td>
<td>Characters cannot appear after the %IMSID keyword, unless a % is used.</td>
</tr>
<tr>
<td>IMS.%IMSID.%DATE.%TIME</td>
<td>%DATE and %TIME substitution values begin with numbers and cannot begin a qualifier.</td>
</tr>
<tr>
<td>IMS.ABCDE%IMSID.DATE%DATE.TM%TIME</td>
<td>All three qualifiers would be more than eight characters after substitution.</td>
</tr>
</tbody>
</table>

- You can include system symbols when specifying values for all data set name keywords. The following considerations and restrictions apply when using system symbols in data set names:

  — System symbols are symbols which have system-specific values. The operating system defines certain static and dynamic system symbols. For example, &SYSNAME is a static symbol whose value is the system name, and &LHMMSS is a dynamic symbol whose value is the current local time. Your installation can also define local system symbols in SYS1.PARMLIB. For more information, see the System Initialization and Tuning Reference Manual and the JCL Reference Manual.

  — A system symbol must begin with an ampersand (&) followed by a name. The name may be a maximum of eight alphanumeric characters and must start with a non-numeric character. A non-alphanumeric character must follow the system symbol (unless the symbol is at the end of the data set name). If a period (.) follows the system symbol, it is considered to be part of the symbol. If you want a period to follow the system symbol value, you must include two periods.

  — Specification of the data set name with system symbols cannot exceed 44 characters. Specification of a member name cannot exceed 8 characters. Message Advisor does not validate the data set name after substitution. For syntax checking purposes, Message Advisor merely validates that the data set name would be valid if the system symbol were replaced with a single alphabetic character.

  — The system substitutes system symbols at allocation. If the system symbol is undefined or if the resulting data set name is invalid, an allocation error will result when the command set is executed. Be aware that the use of certain dynamic system symbols depends on the location of the allocation. For example, if you use &JOBNAME, the substituted value may be the jobname of the Message Advisor Server or of the IMS control region, depending on where the allocation occurs.

  — You may combine Message Advisor symbolic keywords and system symbols in a data set name. Message Advisor substitutes the symbolic keywords first, then the system substitutes any system symbols. Symbolic keywords may be part of
the system symbol name. For example, if DSNAME=&REC1%IMSID when the
IMSID is IMSA, it resolves to DSNAME=&REC1IMSA, so there must be a
system symbol named &REC1IMSA whose value is the data set name to be
used. Be sure to use a terminating period (.) after a system symbol that is
followed by a symbolic keyword if you do not want the value of the symbolic
keyword to be part of the system symbol (for example,
DSNAME=SYS1.&SYSNAME.%IMSID.EXTRACT would resolve to
SYS1.SYSAIMSA.EXTRACT when the system name is SYSA and the IMSID is
IMSA).

You can use system symbols for the following SERVER_Options subcommand
keywords:

—GRS_qname
—SSCTname.
—TITLE

A system symbol begins with an ampersand (&) and optionally ends with a
period (.). The system symbols are resolved on the operating system where the
Message Advisor Server resides. If a system symbol cannot be resolved, no
error message is issued and the unresolved system symbol remains as part of
the keyword's value. The unresolved system symbol may cause unpredictable
problems.

You can use system symbols for the following IMS_Options subcommand
keywords:

—AUTO_CMD_COLDCOM
—AUTO_CMD_COLDSYS
—AUTO_REQ_COLDCOM

A system symbol begins with an ampersand (&) and optionally ends with a
period (.). The system symbols are resolved on the operating system where the
IMS system resides. If a system symbol cannot be resolved, no error message is
issued and the unresolved system symbol remains as part of the keyword's
value. The unresolved system symbol may cause unpredictable problems.

Locating tasks

Message Advisor tasks are described in the manuals that are provided with the
product.
The tables in this section are a quick reference to the most common Message Advisor tasks.

Table 6 on page 42 lists tasks that are specific to the QPF component, the manual containing information about the task, and the number of the chapter that describes the task.

**Table 6: QPF task locator table**

<table>
<thead>
<tr>
<th>Subject or Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>QPF overview</td>
<td>The chapter in the <em>Message Advisor for IMS User Guide</em> that describes the Message Advisor QPF</td>
</tr>
<tr>
<td>installation procedures</td>
<td>Table 7 on page 43</td>
</tr>
<tr>
<td>QPF automated processing</td>
<td>The chapters in the <em>Message Advisor for IMS User Guide</em> that describe the Message Advisor QPF and how to define QPF options</td>
</tr>
<tr>
<td>QPF manual processing</td>
<td>The chapters in the <em>Message Advisor for IMS User Guide</em> that describe the Message Advisor QPF and how to manually initiate QPF actions</td>
</tr>
<tr>
<td>building QPF_OPTIONS command sets</td>
<td>■ The chapters in the <em>Message Advisor for IMS User Guide</em> that describe how to define QPF options and how to build a QPF_OPTIONS command set</td>
</tr>
<tr>
<td></td>
<td>■ “QPF_OPTIONS command” on page 71</td>
</tr>
<tr>
<td>testing QPF_OPTIONS command sets</td>
<td>In the <em>Message Advisor for IMS User Guide</em>, the chapters that describe how to define QPF options and how to build a QPF_OPTIONS command set, and the appendix that describes the Message Advisor test applications</td>
</tr>
<tr>
<td>accessing QPF Problem List</td>
<td>■ The chapter in the <em>Message Advisor for IMS User Guide</em> that describes how to manually initiate QPF actions</td>
</tr>
<tr>
<td></td>
<td>■ “QPF_LIST command” on page 63</td>
</tr>
<tr>
<td>processing commands from QPF Problem List</td>
<td>■ The chapter in the <em>Message Advisor for IMS User Guide</em> that describes how to manually initiate QPF actions</td>
</tr>
<tr>
<td></td>
<td>■ “QPF_LIST command” on page 63</td>
</tr>
<tr>
<td>Subject or Task</td>
<td>Reference</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Activate QPF_OPTIONS example</td>
<td>■ “QPF_OPTIONS command” on page 71</td>
</tr>
<tr>
<td></td>
<td>■ installation guide</td>
</tr>
<tr>
<td>Set QPF_OPTIONS example</td>
<td>■ The chapter in the <em>Message Advisor for IMS User Guide</em> that describes how to build a QPF_OPTIONS command set</td>
</tr>
<tr>
<td></td>
<td>■ “QPF_OPTIONS command” on page 71</td>
</tr>
<tr>
<td>List QPF_OPTIONS example</td>
<td>“QPF_OPTIONS command” on page 71</td>
</tr>
<tr>
<td>List QPF Problems example</td>
<td>“QPF_LIST command” on page 63</td>
</tr>
<tr>
<td>reports and statistics</td>
<td>The appendix in the <em>Message Advisor for IMS User Guide</em> that describes the Message Advisor reports</td>
</tr>
<tr>
<td>syntax diagrams—QPF commands</td>
<td>The command set syntax topics in this book</td>
</tr>
<tr>
<td>WTOR commands—QPF</td>
<td>“Introduction” on page 19</td>
</tr>
</tbody>
</table>

Table 7 on page 43 lists tasks that are specific to the QMR component, the manual containing information about the task, and the number of the chapter that describes the task.

### Table 7: QMR task locator table

<table>
<thead>
<tr>
<th>Subject or Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Advisor overview</td>
<td>The introduction chapter in the <em>Message Advisor for IMS User Guide</em></td>
</tr>
<tr>
<td>Message Advisor initial installation</td>
<td>installation guide</td>
</tr>
<tr>
<td>Message Advisor maintenance installation</td>
<td>installation guide</td>
</tr>
<tr>
<td>Message Advisor customization</td>
<td>■ installation guide</td>
</tr>
<tr>
<td></td>
<td>■ “CUSTOMIZE command” on page 147</td>
</tr>
<tr>
<td>Message Advisor installation verification procedure</td>
<td>installation guide</td>
</tr>
<tr>
<td>accessing help online</td>
<td>The chapter in the <em>Message Advisor for IMS User Guide</em> that describes the QPF_LIST command</td>
</tr>
<tr>
<td>accessing messages online</td>
<td>The chapter in the <em>Message Advisor for IMS User Guide</em> that describes the QPF_LIST command</td>
</tr>
<tr>
<td>Subject or Task</td>
<td>Reference</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>displaying IMS message queues</td>
<td>The chapter in the <em>Message Advisor for IMS User Guide</em> that describes the CUSTOMIZE command</td>
</tr>
<tr>
<td>checkpoint selection</td>
<td>■ The chapter in the <em>Message Advisor for IMS User Guide</em> that describe the QPF_ACTION command, and the chapter that describes the IMSCMD and PLEXCMD commands</td>
</tr>
<tr>
<td></td>
<td>■ “REQUEUE command” on page 227</td>
</tr>
<tr>
<td>using the Message Advisor ISPF interface</td>
<td>The chapter in the <em>Message Advisor for IMS User Guide</em> that describes the QPF_LIST command</td>
</tr>
<tr>
<td>creating and executing commands sets</td>
<td>The chapter in the <em>Message Advisor for IMS User Guide</em> that describes the QPF_OPTIONS command</td>
</tr>
<tr>
<td>DEQUEUE examples</td>
<td>■ The chapter in the <em>Message Advisor for IMS User Guide</em> that describes the DEQUEUE command</td>
</tr>
<tr>
<td></td>
<td>■ “DEQUEUE command” on page 177</td>
</tr>
<tr>
<td>DISPLAY examples</td>
<td>■ The chapter in the <em>Message Advisor for IMS User Guide</em> that describes the DISPLAY command</td>
</tr>
<tr>
<td>REQUEUE examples</td>
<td>■ The chapter in the <em>Message Advisor for IMS User Guide</em> that describes the REQUEUE command</td>
</tr>
<tr>
<td></td>
<td>■ “REQUEUE command” on page 227</td>
</tr>
<tr>
<td>UNLOAD examples</td>
<td>■ The chapter in the <em>Message Advisor for IMS User Guide</em> that describes the UNLOAD command</td>
</tr>
<tr>
<td></td>
<td>■ “UNLOAD command” on page 281</td>
</tr>
<tr>
<td>reports and statistics</td>
<td>The appendix in the <em>Message Advisor for IMS User Guide</em> that describes the Message Advisor reports</td>
</tr>
<tr>
<td>syntax diagrams—Message Advisor commands</td>
<td>The command set syntax topics in this book</td>
</tr>
<tr>
<td>CPU ID password processing</td>
<td>installation guide</td>
</tr>
<tr>
<td>WTOR commands—Message Advisor Server</td>
<td>“Introduction” on page 19</td>
</tr>
</tbody>
</table>
Table 8 on page 45 lists tasks that are specific to using Message Advisor in a shared queues environment, the manual containing information about the task, and the number of the chapter that describes the task.

### Table 8: Message Advisor shared queues task locator table

<table>
<thead>
<tr>
<th>Subject or Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Advisor utilities</td>
<td>The appendix in the <em>Message Advisor for IMS User Guide</em> that describes the Message Advisor utilities</td>
</tr>
<tr>
<td>diagnostics tools—TRACE</td>
<td>installation guide</td>
</tr>
<tr>
<td>assigning conditional steps to a Message Advisor request</td>
<td>“COND pseudo command” on page 399</td>
</tr>
<tr>
<td>using the Message Advisor application interface</td>
<td>“Message Advisor for IMS Application Interface” on page 405</td>
</tr>
</tbody>
</table>

<p>| overview of QMR in a shared queues environment      | The chapter in the <em>Message Advisor for IMS User Guide</em> that describes how to use Message Advisor in a shared queues environment |
| installation procedures                             | Table 7 on page 43                                                         |
| converting QMR command sets to QMR shared queues command sets | The chapter in the <em>Message Advisor for IMS User Guide</em> that describes how to use Message Advisor in a shared queues environment |
| specifying QMR shared queues keywords               | The chapter in the <em>Message Advisor for IMS User Guide</em> that describes how to use Message Advisor in a shared queues environment |
| defining QMR shared queues command sets             | The chapter in the <em>Message Advisor for IMS User Guide</em> that describes how to use Message Advisor in a shared queues environment |
| displaying QMR shared queues messages               | The chapter in the <em>Message Advisor for IMS User Guide</em> that describes how to use Message Advisor in a shared queues environment |
| DEQUEUE examples                                    | • The chapter in the <em>Message Advisor for IMS User Guide</em> that describes how to use Message Advisor in a shared queues environment |
|                                                    | • “Message Advisor shared queues command sets” on page 371                |</p>
<table>
<thead>
<tr>
<th>Subject or Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY examples</td>
<td>- The chapter in the <em>Message Advisor for IMS User Guide</em> that describes how to use Message Advisor in a shared queues environment</td>
</tr>
<tr>
<td></td>
<td>- “Message Advisor shared queues command sets” on page 371</td>
</tr>
<tr>
<td>REQUEUE examples</td>
<td>- The chapter in the <em>Message Advisor for IMS User Guide</em> that describes how to use Message Advisor in a shared queues environment</td>
</tr>
<tr>
<td></td>
<td>- “Message Advisor shared queues command sets” on page 371</td>
</tr>
<tr>
<td>UNLOAD examples</td>
<td>- The chapter in the <em>Message Advisor for IMS User Guide</em> that describes how to use Message Advisor in a shared queues environment</td>
</tr>
<tr>
<td></td>
<td>- “Message Advisor shared queues command sets” on page 371</td>
</tr>
<tr>
<td>using the QMR Analyzer Facility</td>
<td>The chapter in the <em>Message Advisor for IMS User Guide</em> that describes the Message Advisor Analyzer Facility</td>
</tr>
</tbody>
</table>
QPF_ACTION command

This chapter describes the Message Advisor for IMS QPF_ACTION command set, and also provides a syntax diagram and sample command set. For more information about building and executing a QPF_ACTION command set, see the Message Advisor for IMS User Guide.

Overview

The Queue Protection Facility (QPF) is primarily an automated facility that manages local message queues according to a set of options that you define. If your QPF_OPTIONS command set does not automatically resolve a situation, you can issue one or more manual corrective actions through the QPF_ACTION command. The command’s primary purpose is communicating and initiating QPF actions within an IMS address space, through the Message Advisor Interactive System Productivity Facility (ISPF) interface.

Initiate an action

You can use any of the following methods to initiate an action against a problem that is displayed on the QPF Problem List:

- display the problem list from the Message Advisor ISPF interface panels and issue a QPF_ACTION command set interactively
- issue a QPF Processing WTOR Action command
- build and submit a QPF_ACTION command set
Display a problem list from the Message Advisor ISPF interface panels

Normally, the QPF_ACTION command functions as an internal Message Advisor command that initiates an action against a problem which is displayed on a QPF Problem List.

You can access a problem list from the Message Advisor ISPF interface panels when QPF passes the Monitor Phase threshold, then initiate a corrective action against any problem that is displayed on the list.

To initiate an action, type the code for the QPF action that you are requesting in the Act field adjacent to the problem. QPF initiates an internal system QPF_ACTION command to implement the requested action.

Issue a QPF Processing WTOR Action command

QPF issues the following Processing WTOR (write-to-operator with reply) message when a phase starts, if you specify WTOR=YES for that phase in the QPF_OPTIONS command set:

```
BMC43168A PRD QPF FOR IMSID=imsid - REPLY 'LIST' OR 'HELP'
```

You can respond to message BMC43168A with a Processing WTOR Action command.

You can issue a Processing WTOR Action command in response to a Processing WTOR message if Time Sharing Option (TSO) is not available or if you are at the system console.

Build a QPF_ACTION command set

You can also initiate an action by building a QPF_ACTION command set that specifies the requested action. You can use ISPF EDIT or the Message Advisor ISPF interface panels to build the command set. You can submit a batch job to execute the command set, or you can execute the command set through the Message Advisor ISPF interface panels in foreground or background.

At times, you may find it useful to build a QPF_ACTION command set. For example, if ISPF is unavailable you cannot initiate an action from the QPF Problem List displayed through the Message Advisor ISPF interface. In this situation, you might want to initiate a corrective action through a batch request.
QPF actions

You can initiate the following QPF actions against problems that are displayed on the QPF Problem List:

- **ABEND** schedules a task to terminate the application the next time the IMS region tries to insert a message. The application terminates with pseudo-abend U0474.

- **DEQUEUE** starts a task to delete the queued messages for a destination.

  **Note**  
  Before initiating a dequeue, QPF performs /STOP commands to ensure no activity during the dequeue. If the STOP action is not also specified, the entity restarts after the dequeue is complete.  
  QPF performs the /STOP commands that are listed in Table 9 on page 50. In addition, QPF may perform a /STOP USER or a /PSTOP LINK.

- **FAIL** schedules a task to issue an A7 status code to the application or Advanced Program-to-Program Communications (APPC) input device on the next attempt to insert a message.

  **Note**  
  QPF ignores the FAIL action for all input devices except APPCs. FAIL results in de-allocation of an APPC input device.

- **IPOST** starts a task to restart an IMS region or input device that QPF has placed in a wait through the IWAIT command.

  **Note**  
  You can use ACTION=POST as an alias for ACTION=IPOST.

- **IWAIT** schedules a task to IWAIT the region or input device on the next attempt to insert a message.

  **Note**  
  You can use ACTION=WAIT as an alias for ACTION=IWAIT.

- **STOP** starts a task to issue a /STOP command and wait for the response. Destinations stop immediately; regions and input devices stop the next time they try to insert a message.  
  Table 9 on page 50 lists the commands that the STOP action performs for each device type.
### Table 9: STOP action commands by device type

<table>
<thead>
<tr>
<th>Device Type</th>
<th>STOP Action Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMB destinations</td>
<td>/STOP TRANSACTION xxxxxxxxx</td>
</tr>
<tr>
<td>APPC destinations</td>
<td>/STOP LUNAME xxxxxxxxx INPUT &amp; OUTPUT</td>
</tr>
<tr>
<td>MSNAME destinations</td>
<td>/STOP MSNAME xxxxxxxxx</td>
</tr>
<tr>
<td>OTMA destinations</td>
<td>/STOP TMEMBER tmember TPIPE tpipe</td>
</tr>
<tr>
<td>other destinations</td>
<td>/STOP LTERM xxxxxxxxx</td>
</tr>
<tr>
<td>MPP regions</td>
<td>/STOP TRANSACTION xxxxxxxxx</td>
</tr>
<tr>
<td>BMP regions</td>
<td>/STOP PROGRAM xxxxxxxxx</td>
</tr>
<tr>
<td>regions (if TRAN/PGM unknown)</td>
<td>/STOP REGION nnnn</td>
</tr>
<tr>
<td>APPC input devices</td>
<td>/STOP LUNAME xxxxxxxxx INPUT</td>
</tr>
<tr>
<td>MSNAME input devices</td>
<td>/STOP MSNAME xxxxxxxxx</td>
</tr>
<tr>
<td>OTMA input devices</td>
<td>/STOP TMEMBER tmember TPIPE tpipe</td>
</tr>
<tr>
<td>VTAM input devices</td>
<td>/STOP NODE xxxxxxxxx</td>
</tr>
<tr>
<td>other devices</td>
<td>/STOP LTERM xxxxxxxxx</td>
</tr>
</tbody>
</table>

- **UNLOAD_DEQUEUE** starts a task to unload and dequeue the queued messages for a destination.

  If you select the UNLOAD_DEQUEUE action, an UNLOAD_DSN subcommand must be present in the QPF options. The UNLOAD_DSN subcommand determines the unload data set to be used.

  **Note**

  Before initiating an unload and dequeue, QPF performs /STOP commands to ensure no activity during the unload and dequeue. If the STOP action is not also specified, the entity restarts after the unload and dequeue is complete.

  QPF performs the /STOP commands listed in Table 9 on page 50. QPF may also perform a /STOP USER or a /PSTOP LINK.

  You can use `ACTION=DEQUEUE_UNLOAD`, `ACTION=DEQ_UNL`, and `ACTION=UNL_DEQ` as aliases for `ACTION=UNLOAD_DEQUEUE`.

QPF can only take certain actions for destinations, regions, and input devices. Table 9 on page 50 lists the actions that QPF can take for each type. “TYPE keyword” on page 60 includes a table that lists the destinations, regions, and input devices for which QPF can take the actions.
Table 10: QPF actions for destinations, regions, and input devices

<table>
<thead>
<tr>
<th>QPF Action</th>
<th>Destinations</th>
<th>Regions</th>
<th>Input Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABEND</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>DEQUEUE</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FAIL</td>
<td>X</td>
<td></td>
<td>X&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>IPOST</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>IWAIT</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>STOP</td>
<td>X</td>
<td>X</td>
<td>X&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>UNLOAD_DEQUEUE</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

1. FAIL is not supported for OTMA input devices.
2. STOP is not supported for BTAM devices.

For more information about initiating a QPF action from the QPF Problem List, see the *Message Advisor for IMS User Guide*. 

---

*a, b* Indicate specific notes or restrictions related to the input devices.
The following figure shows the QPF_ACTION command set syntax.

**Figure 2: QPF_ACTION command set syntax**

```plaintext
QPF_ACTION
   | IMSId/IMS=cccc
   | ,TYPE= Any/All/A
       | BMP
       | CNT
       | DEADQ
       | DYNAmic/VIRtual/ETO/V/TF/CNT-E/CNT-V
   | IN-CLB
   | IN-CNT
   | IN-LUN
   | IN-MSN
   | IN-TPI
       | LTERMs
       | LUname/TPName/APPC
   | MAQ/QMR
   | MPP/REGion/RGN
   | MSName
   | OTMA/TPPIPE
   | RSMB
   | SMB
       | STATic
   | SYSTEM
   | TRN/TRANsection
       | VSPont
   | ,NETID=cccccccc
       | up to 64 characters
   | ,TPName=cc...cc
   | ,MEMBER/ TMEMBER=cccccccccccccccccc
   | ,REGion=nnnn
       | range 1–4095
   | ACTION= ABEND
       | DEQueue
       | FAIL
       | IPOST/POST
       | IWAIT/WAIT
       | STOP
       | UNLoad/DEQUEUE/DEQUEUE_UNLoad/
       | DEQ_UNL/UNL_DEQ
   | ,WAIT=No Yes
   | ,FORCE=No Yes
   | END
```
Sample QPF_ACTION command set

Message Advisor provides a sample QPF_ACTION command set that can serve as a template for building your own QPF_ACTION command set. The MAQSAMP library contains the sample command set.

Table 11 on page 53 identifies the library member that contains the sample QPF_ACTION command set. MAQSAMP also contains an index in member QMR@INDX that lists all sample command sets that are provided with Message Advisor.

Table 11: Sample QPF_ACTION Command Set

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QPF@ACT1</td>
<td>unloads, dequeues, and stops a certain destination</td>
</tr>
</tbody>
</table>

**Note**

The sample contains ?, which you should replace with your site-specific values. You cannot build this command set interactively by using the Message Advisor ISPF interface; however, you can directly invoke ISPF EDIT from the interface for this purpose.

Figure 3 on page 53 shows a command set for taking action to correct a message queue overflow problem identified on the QPF Problem List. This command set is identical to the QPF_ACTION command set that is provided in MAQSAMP(QPF@ACT1).

In this sample, the command requests that processing occur against the CNT destination (TYPE) for IMSID (?). ACTION=UNLOAD_DEQUEUE indicates that all CNT destinations will be unloaded and dequeued. Message Advisor waits (WAIT=YES) until the action is complete before resuming.

**Figure 3: Command set: Initiate a QPF action**

\[
\begin{align*}
\text{QPF\_ACTION} & \quad \text{IMSID=?}, \text{TYPE=CNT}, \text{NAME=?}, \\
& \quad \text{ACTION=UNLOAD\_DEQUEUE}, \text{WAIT=YES} \\
\text{END}
\end{align*}
\]

QPF_ACTION command set description

The QPF_ACTION command set consists of the following items:

- QPF_ACTION primary command
- keywords and parameters
Keyword and parameter descriptions are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

---

**ACTION keyword**

ACTion is a required keyword for the QPF_ACTION command.

Table 12 on page 54 lists the ACTion keyword parameters. You can specify only one parameter for the ACTion keyword.

**Table 12: ACTION keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| ABEND     | terminates the application program with pseudo-abend U0474 the next time that the application tries to insert a message  
*Note:* ABEND is valid only for an IMS message processing region. |
| DEQueue   | deletes the queued messages for a destination  
*Note:* DEQueue is valid only for destinations. |
| FAIL      | returns an A7 status code to the application the next time an IMS region tries to insert a message; deallocates the input device the next time an APPC program tries to insert a message; rejects all subsequent input messages with either a NAK or a DFS1289 message to the client for OTMA TPIPEC  
*Note:* FAIL is valid only for applications and APPC or OTMA input devices. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPOST/POST</td>
<td>restarts an IMS region or input device that QPF has placed in a wait through the IWAIT command, and allows the region or input device to continue; this may cause a message queue overflow if a shortage still exists.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> ENFORCE statements may cause another IWAIT on the next attempt to insert a message. You can use <code>ACTion=POST</code> as an alias for <code>ACTion=IPOST</code>.</td>
</tr>
<tr>
<td></td>
<td>When QPF posts a TPIPE, QPF also posts all YTIBs and YQABs for which QPF has performed the IWAIT action. QPF also restarts the TPIPE if QPF stopped it. QPF cancels any pending FAIL and STOP actions for the TPIPE. For any subsequent TPIPE insert attempts that match the same ENFORCE statement, QPF allows the attempt to proceed without taking action until the end of the current rate interval (the default is 10 minutes). This rule also applies to any newly created ITASKS. However, if a different ENFORCE statement matches, QPF takes the action prescribed in that statement. When the current rate interval ends, the same ENFORCE statement might match again.</td>
</tr>
<tr>
<td>IWAIT/WAIT</td>
<td>places the IMS region or input device in a wait on the next attempt to insert a message. The region or input device remains in a wait until the IPOST action is used to restart it or the Monitor Phase ends.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can use <code>ACTion=WAIT</code> as an alias for <code>ACTion=IWAIT</code>.</td>
</tr>
<tr>
<td>STOP</td>
<td>stops the destination, region, or input device. Destinations stop immediately; regions and input devices stop the next time they try to insert a message.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> BTAM devices are not supported.</td>
</tr>
<tr>
<td>UNLoad_DEQUEUE/</td>
<td>unloads and deletes the queued messages for a destination.</td>
</tr>
<tr>
<td>DEQUEUE_UNLoad/</td>
<td><strong>Note:</strong> If you select this action, an UNLoad_DSN statement must be present in the QPF options. The UNLoad_DSN statement determines the unload data set to be used.</td>
</tr>
<tr>
<td>DEQ_UNL/UNL_DEQ</td>
<td>You can use the following as aliases for <code>ACTion=UNLoad_DEQUEUE</code>:</td>
</tr>
<tr>
<td></td>
<td>- <code>ACTion=DEQUEUE_UNLoad</code></td>
</tr>
<tr>
<td></td>
<td>- <code>ACTion=DEQ_UNL</code></td>
</tr>
<tr>
<td></td>
<td>- <code>ACTion=UNL_DEQ</code></td>
</tr>
</tbody>
</table>
FORCE keyword

FORCE is an optional keyword for the QPF_ACTION command.

Table 13 on page 56 lists the FORCE keyword parameters. You can specify only one parameter for the FORCE keyword.

Table 13: FORCE keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| NO        | stops and quiesces the destination before performing a dequeue or unload action  
If QPF cannot stop or quiesce the destination, it will not perform the action and you will receive an error message with a "DEST NOT STOPPED" status. |
| Yes       | performs unload and dequeue actions without stopping the destination  
**WARNING:** If you specify `FORCE=Yes` for an active destination, the action may fail, IMS message DFS1959E may be received, or (in rare cases) IMS may abend with a U0757. |

IMSID keyword

IMSid is a required keyword for the QPF_ACTION command. IMS must be active before you can specify the IMSid keyword.

**Note**
You can use IMS as an alias for IMSid.

Table 14 on page 56 lists the IMSid keyword parameter.

Table 14: IMSid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| ccccc     | the target IMS system  
**Note:** ccccc is a value that you specify. |

MEMBER keyword

MEMBER is an optional keyword for the QPF_ACTION command.
You can use TMEMBER as an alias for MEMBER.

Table 15 on page 57 lists the MEMBER keyword parameter.

**Table 15: MEMBER keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccc</td>
<td>if TYPE=OTMA, you can specify a TMEMBER name to further identify the destination</td>
</tr>
<tr>
<td>Note: cccccc is a value that you specify.</td>
<td></td>
</tr>
</tbody>
</table>

**NAME keyword**

NAME is a required keyword for the QPF_ACTION command.

Table 16 on page 58 lists the NAME keyword parameter.
### Table 16: NAME keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccccc</td>
<td>the name of the IMS element to be processed; the meaning of NAME depends on the type that you specify:</td>
</tr>
<tr>
<td></td>
<td>- For destination types other than LUNAME or OTMA, the name is the CNT (LTERM) or SMB (TRANSACTION) name.</td>
</tr>
<tr>
<td></td>
<td>- For TYPE=LUNAME, the name is the APPC LUNAME.</td>
</tr>
<tr>
<td></td>
<td>- For TYPE=OTMA, the name is the TPIPE name.</td>
</tr>
<tr>
<td></td>
<td>- For MPP and BMP, the name is the jobname of the IMS REGION.</td>
</tr>
<tr>
<td></td>
<td>- For TRN, the name is the transaction name that is running in the IMS region.</td>
</tr>
<tr>
<td></td>
<td>- For IN-CLB, the name is the VTAM node name of the input device.</td>
</tr>
<tr>
<td></td>
<td>- For IN-CNT, the name is the CNT or LTERM name of the input device.</td>
</tr>
<tr>
<td></td>
<td>- For IN-LUN, the name is the APPC LUNAME of the input device.</td>
</tr>
<tr>
<td></td>
<td>- For IN-MSN, the name is an MSNAME associated with the link for which the action is to be taken.</td>
</tr>
<tr>
<td></td>
<td>- For IN-TPI, the name is the TPIPE name of the input device.</td>
</tr>
<tr>
<td></td>
<td>- For Message Advisor (TYPE=MAQ or QMR), the name is the Message Advisor Server jobname.</td>
</tr>
</tbody>
</table>

**Note:** cccccccc is a value that you specify.

### NETID keyword

NETID is an optional keyword for the QPF_ACTION command.

Table 17 on page 59 lists the NETID keyword parameter.
Table 17: NETID keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| cccccccc  | you can specify a network ID to further identify the destination (allowed only if TYPE=LUNAME)  
  Note: ccccccc is a value that you specify. |

REGION keyword

REGION is an optional keyword for the QPF_ACTION command.

Table 18 on page 59 lists the REGION keyword parameter.

Table 18: REGION keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| nnnn      | if you specify an IMS region (TYPE=MPP/BMP/TRN), you must also specify the region number  
  REGION is required for TYPE=MPP, TYPE=BMP, and TYPE=TRN and is not allowed for any other types.  
  Note: nnnn is a value that you specify. |
  range 1-4095

TPNAME keyword

TPName is an optional keyword for the QPF_ACTION command.

Table 19 on page 59 lists the TPName keyword parameter.

Table 19: TPName keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| cc....cc  | if TYPE=LUNAME, you can specify an APPC TPName to further identify the destination  
  Note: TPName is required if more than one TPName is associated with the LUNAME.  
  cc....cc is a value that you specify. |
  up to 64 characters
**TYPE keyword**

TYPE is a required keyword for the QPF_ACTION command.

Table 20 on page 60 lists the TYPE keyword parameters. You can specify only one parameter for the TYPE keyword. Parameters that are separated by slashes (/) are aliases.

**Table 20: TYPE keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANY/All/A</td>
<td>the type of IMS element to be processed</td>
</tr>
<tr>
<td>IMS destinations:</td>
<td>Obtain the type from the QPF Problem List. Use any of the following methods to view the QPF Problem List:</td>
</tr>
<tr>
<td>CNT</td>
<td>■ issue the operator LIST command</td>
</tr>
<tr>
<td>DEADQ</td>
<td>■ build and submit a QPF_LIST command</td>
</tr>
<tr>
<td>DYNAMIC/VIRTual/ETO/</td>
<td>■ select option 8 from the Message Advisor for IMS Primary Menu</td>
</tr>
<tr>
<td>VTF/CNT-E/ CNT-V</td>
<td>For input devices:</td>
</tr>
<tr>
<td>LTERM</td>
<td>Links and MSNAMEs will not match the IN-CLB or IN-CNT types. Only IN-MSN will match for remote input.</td>
</tr>
<tr>
<td>LUname/TPName/APPC</td>
<td></td>
</tr>
<tr>
<td>MSNAME</td>
<td></td>
</tr>
<tr>
<td>OTMA/TPIPE</td>
<td></td>
</tr>
<tr>
<td>RSMB</td>
<td></td>
</tr>
<tr>
<td>SMB</td>
<td></td>
</tr>
<tr>
<td>STATIC</td>
<td></td>
</tr>
<tr>
<td>SYSTEM</td>
<td></td>
</tr>
<tr>
<td>VSPCNT</td>
<td></td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>IMS regions:</td>
<td></td>
</tr>
<tr>
<td>BMP</td>
<td></td>
</tr>
<tr>
<td>MPP/REGion/RGN</td>
<td></td>
</tr>
<tr>
<td>TRN/TRANsaction</td>
<td></td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>input devices:</td>
<td></td>
</tr>
<tr>
<td>IN-CLB</td>
<td></td>
</tr>
<tr>
<td>IN-CNT</td>
<td></td>
</tr>
<tr>
<td>IN-LUN</td>
<td></td>
</tr>
<tr>
<td>IN-MSN</td>
<td></td>
</tr>
<tr>
<td>IN-TPI</td>
<td></td>
</tr>
<tr>
<td>MAQ</td>
<td></td>
</tr>
</tbody>
</table>
WAIT keyword

WAIT is an optional keyword for the QPF_ACTION command and indicates what Message Advisor is to do after initiating a QPF action. The default is to wait for actions that are taken against a destination, but not to wait for actions that are taken against an IMS region or input device. Message Advisor takes actions against IMS regions or input devices only when they try to insert a message. If no messages are inserted, Message Advisor does not take the action and the wait does not terminate.

Table 21 on page 61 lists the WAIT keyword parameters. You can specify only one parameter for the WAIT keyword.

Table 21: WAIT Keyword Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Message Advisor continues processing other commands. The return code will be zero unless QPF could not initiate the action. Message Advisor does not return messages from the action. Message Advisor sends messages for the action to the operator console instead.</td>
</tr>
<tr>
<td>Yes</td>
<td>Message Advisor waits until the action is complete and returns any messages that are associated with the action.</td>
</tr>
</tbody>
</table>
QPF_LIST command

This chapter describes the Message Advisor for IMS QPF_LIST command set, and also provides a syntax diagram and sample command set. For more information about building and executing a QPF_LIST command set, see the Message Advisor for IMS User Guide.

Overview

When the Queue Protection Facility (QPF) reaches the threshold you define for the Monitor Phase, it begins analyzing message and temporary queue utilization. Based on the analysis, QPF identifies and produces a list of potential problem destinations and regions. QPF also identifies and tracks origins that have experienced high activity since the beginning of the Monitor Phase.

Display the QPF Problem List

You can use any of the following methods to view the QPF Problem List:

- display the list from the Message Advisor ISPF interface panels
- issue the QPF Processing WTOR LIST command
- build and submit a QPF_LIST command set

In most circumstances, the easiest way to display the QPF Problem List is from the Message Advisor Interactive System Productivity Facility (ISPF) interface panels. Regardless of which method you choose, QPF does not create the problem list until message and temporary queue usage reaches the Monitor Phase threshold.

Display from the Message Advisor ISPF interface panels

You can view the QPF Problem List interactively through the Message Advisor ISPF interface. On the Message Advisor for IMS Primary Menu, type 8 (Queue Protection
Facility) in the choice entry field and press Enter to display the QPF Problem List. This action initiates an internal system QPF_LIST command that displays the list of QPF problems.

The maximum number of problems initially displayed by Message Advisor depends on how you customize your Session Control options. When the QPF Problem List is displayed, you can dynamically change the maximum number of problems that are allowed.

**Issue the QPF Processing WTOR LIST command**

QPF issues the following Processing WTOR (write-to-operator with reply) message when a phase starts, if you specify WTOR=YES for that phase in the QPF_OPTIONS command set:

```
BMC43168A PRD QPF FOR IMSID=imsid - REPLY 'LIST' OR 'HELP'
```

You can respond to message BMC43168A with the Processing WTOR LIST command. The command has the following format:

```
LIST nn mm
```

The command lists the worst nn problem destinations and/or regions on the IMS queues (default is 20), and lists the mm most active origins in the IMS system (default is 5), where nn and mm are numbers from 0 to 99.

You can issue the Processing WTOR LIST command to display a problem list in response to a Processing WTOR message if Time Sharing Option (TSO) is not available or if you are at the system console.

**Build a QPF_LIST command set**

Normally, the QPF_LIST command functions as an internal Message Advisor command that accesses the QPF Problem List through the Message Advisor ISPF interface. You can display a problem list by building a QPF_LIST command set that specifies TYPE=PROBLEMS or TYPE=DETAIL. You can use ISPF EDIT or the Message Advisor ISPF interface panels to build the command set. You can submit a batch job to execute the command set, or you can execute the command set through the Message Advisor ISPF interface panels in foreground or background.

At times, you may find it useful to build a QPF_LIST command set. For example, if ISPF is unavailable you cannot display the QPF Problem List through the Message Advisor ISPF interface. In this situation, you might want to issue a batch request to display the QPF Problem List.
Display current QPF options

The QPF_LIST command also lets you display the active QPF options. To list the current QPF options, issue a QPF_LIST command set that specifies `TYPE=OPTIONS`.

QPF_LIST command set syntax

The following figure shows the QPF_LIST command set syntax.

Figure 4: QPF_LIST command set syntax

Sample QPF_LIST command set

Message Advisor provides sample QPF_LIST command sets that can serve as templates for building your own QPF_LIST command set. The MAQSAMP library contains the sample command sets.

Table 22 on page 65 identifies the library members that contain the sample QPF_LIST command sets. MAQSAMP also contains an index in member QMR@INDX that lists all sample command sets that are provided with Message Advisor.

Table 22: Sample QPF_LIST command sets

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QPF@LST</td>
<td>lists all QPF problem entries</td>
</tr>
<tr>
<td>QPF@LST1</td>
<td>lists the top 20 QPF problem entries</td>
</tr>
</tbody>
</table>
QPF@LST2  lists the active options

---

**Note**
The samples contain ?, which you should replace with your site-specific values. You cannot build these command sets interactively by using the Message Advisor ISPF interface; however, you can directly invoke ISPF EDIT from the interface for this purpose.

---

**Figure 5 on page 66** shows a command set for displaying the QPF Problem List. This command set is identical to the QPF_LIST command set that is provided in MAQSAMP(QPF@LST1).

In this sample, the command requests that QPF display the first 20 problems (PROBLEMS) for IMSID (?).

**Figure 5: Command set: Display the QPF problem list**

```
QPF_LIST IMSID=?,?,PROBLEMS=20
END
```

For more information about accessing the QPF Problem List through the Message Advisor ISPF interface, see the *Message Advisor for IMS User Guide*.

---

**QPF_LIST command set description**

The QPF_LIST command set consists of the following items:

- QPF_LIST primary command
- keywords and parameters
- END command

---

**Note**
Keyword and parameter descriptions are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.
ACTIVE keyword

ACTIVE is an optional keyword for the QPF_List command.

Table 23 on page 67 lists the ACTIVE keyword parameter.

Table 23: ACTIVE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>indicates that some destinations and programs for which a QPF action (such as IWAIT) is in progress are to be included in the problem list, even if the number of queued messages is too small for the destination or program to make the list otherwise. The nn value indicates the maximum number of active destinations and programs to include in the list. The default is half the value that is specified for the PROBlems keyword. If ACTIVE is specified with PROBlems=0, QPF lists only active destinations and programs, up to the maximum specified. <strong>Note:</strong> nn is a value that you specify.</td>
</tr>
</tbody>
</table>

IMSid keyword

IMSid is a required keyword for the QPF_List command.

IMS must be active before you can specify the IMSid keyword.

**Note**
You can use IMS as an alias for IMSid.

Table 24 on page 67 lists the IMSid keyword parameter.

Table 24: IMSid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccc</td>
<td>the target IMS system</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> cccc is a value that you specify.</td>
</tr>
</tbody>
</table>

ORIGins keyword

ORIGins is an optional keyword for the QPF_List command.
Table 25 on page 68 lists the ORIGins keyword parameter.

Table 25: ORIGins keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 nn</td>
<td>the number of origins to include in the QPF Problem List</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> nn is a value that you specify.</td>
</tr>
</tbody>
</table>

**PROBlems keyword**

PROBlems is an optional keyword for the QPF_List command.

Table 26 on page 68 lists the PROBlems keyword parameter.

Table 26: PROBlems keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 nn</td>
<td>the number of destinations and programs to include in the QPF Problem List</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> nn is a value that you specify.</td>
</tr>
</tbody>
</table>

**SKIP_ORIGins keyword**

SKIP_ORIGins is an optional keyword for the QPF_List command.

Table 27 on page 68 lists the SKIP_ORIGins keyword parameter.

Table 27: SKIP_ORIGins keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 nn</td>
<td>the number of origins to skip when printing the QPF Problem List</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> nn is a value that you specify.</td>
</tr>
</tbody>
</table>

**SKIP_PROBlems keyword**

SKIP_PROBlems is an optional keyword for the QPF_List command.
Table 28 on page 69 lists the SKIP_PROBLEMS keyword parameter.

**Table 28: SKIP_PROBLEMS keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0         | the number of destinations and programs to skip when printing the QPF Problem List  
**Note:** *nn* is a value that you specify. |
| *nn*      |             |

**TYPE keyword**

TYPE is an optional keyword for the QPF_List command. TYPE indicates the type of report desired.

Table 29 on page 69 lists the TYPE keyword parameters.

**Table 29: TYPE keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROBLEMS</td>
<td>displays a problem list similar to the report generated for the operator console</td>
</tr>
<tr>
<td>DETAIL</td>
<td>displays a problem list with an expanded format, including: the TPNAME for APPC destinations, the TMEMBER for OTMA destinations, and the statement number for actions in progress</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>displays the same report generated by the QPF_OPTIONS TYPE=LIST command</td>
</tr>
<tr>
<td>STATISTICS</td>
<td>displays statistics relating to the use of QPF for this IMS</td>
</tr>
</tbody>
</table>
QPF_OPTIONS command

This chapter describes the Message Advisor for IMS QPF_OPTIONS command set, and also provides a syntax diagram and sample command sets. For more information about building and executing a QPF_OPTIONS command set, see the Message Advisor for IMS User Guide.

Overview

The QPF_OPTIONS command lets you set, replace, load, unload, or list Queue Protection Facility (QPF) options while IMS is active.

When QPF is active, it continuously monitors the state of the IMS message queues to determine which processing phase should be in effect. QPF takes action based on tasks that you define for each phase. You use the QPF_OPTIONS command set to define the thresholds and tasks for each phase.

A phase starts when either of the IMS message queue data sets (SHMSG or LGMSG) exceed the pre-defined phase threshold. A phase ends when it meets reset criteria, which consist of a percentage value and a time. To meet the reset criteria, both queue data sets must stay below the percentage value for the time indicated.

Once you have defined the QPF options, QPF monitors and resolves most message queue overflow situations with little or no intervention from you. QPF automatically initiates the tasks that you define for the QPF options at the appropriate time.

Although QPF is primarily an automated facility, authorized users can manually initiate some tasks while QPF is active. Authorized users can intervene to resolve any situation that the QPF_OPTIONS command set does not automatically resolve.

QPF_OPTIONS subcommands and statements

The QPF_OPTIONS command set consists of the main command and keywords, with subcommands and statements and their associated keywords. The QPF_OPTIONS subcommands and statements are as follows:
- MONITOR_PHASE subcommand sets Monitor Phase thresholds and options

- PROTECT_PHASE subcommand sets Protect Phase thresholds and options
  - ENFORCE statement identifies automatic limits and restrictions to enforce during the Protect Phase
  - PROCESS statement identifies automatic actions to initiate during the Protect Phase

  **Note**
  To dequeue the master or secondary master terminal when the Protect Phase threshold is reached, create a PROCESS statement as follows:
  - Specify the DESTINATION keyword as the explicit name of the master terminal or secondary master terminal.
  - Specify FORCE=YES.
  - Do not specify ACTION=STOP since the STOP parameter is not supported for master terminals.

- OVERFLOW_PHASE subcommand sets Overflow Phase thresholds and options

- UNLOAD_DSN subcommand identifies data sets to be used when you specify an UNLOAD action on any PROCESS statement or QPF_ACTION command
  The UNLOAD_DSN subcommand is required when you specify an UNLOAD action on a PROCESS statement. The QPF_ACTION command and the UNLOAD Processing WTOR Action command also require that the UNLOAD_DSN subcommand be present in the QPF_OPTIONS command set.

  **Note**
  When you set or replace QPF options, the QPF_OPTIONS command set must contain the MONITOR_PHASE, PROTECT_PHASE, and OVERFLOW_PHASE subcommands. The ENFORCE and PROCESS statements and the UNLOAD_DSN subcommand are optional. However, when you list, load, or unload QPF options, you cannot specify any of the subcommands or statements. You can only use the main command keywords to list, load, or unload QPF options.

For more information about building and executing a QPF_OPTIONS command set, see the *Message Advisor for IMS User Guide*. 
QPF_OPTIONS command set syntax

The following figures show the QPF_OPTIONS command set syntax.

Figure 6: QPF_OPTIONS command set syntax (part 1 of 7)
Figure 7: QPF_OPTIONS command set syntax (part 2 of 7)
Figure 8: QPF_OPTIONS command set syntax (part 3 of 7)
Figure 9: QPF_OPTIONS command set syntax (part 4 of 7)
Figure 10: QPF_OPTIONS command set syntax (part 5 of 7)
Figure 11: QPF_OPTIONS command set syntax (part 6 of 7)
Sample QPF_OPTIONS command sets

Message Advisor provides sample QPF_OPTIONS command sets that can serve as templates for building your own QPF_OPTIONS command sets.
The MAQSAMP library contains the command sets.

Table 30 on page 80 identifies the library members that contain the sample QPF_OPTIONS command sets.

### Table 30: Sample QPF_OPTIONS command sets

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QPF@OPT1</td>
<td>sample options with various commented examples</td>
</tr>
</tbody>
</table>
| QPF@OPT2 | defines QPF options that address the queue overflow problems that are described in the test case  
          | For more information about the test case, see the Message Advisor for IMS User Guide.  
          | **Note:** You should not use this command set as a template for building your own QPF_OPTIONS command set. |
| QPF@OPT3 | template using variables  
          | **Note:** You can use this sample command set as a template for building your own QPF_OPTIONS command set. |
| QPF@OPT4 | suggested options when using shared queues                                  |
| QPF@OPT5 | defines escalating ENFORCE actions                                          |

**Note**

The samples contain `?`, which you should replace with your site-specific values. You cannot build these command sets interactively by using the Message Advisor ISPF interface; however, you can directly link the interface to ISPF EDIT for this purpose.

---

**Activate QPF_OPTIONS command set**

The following figure shows the sample command set for activating QPF. MAQSAMP(QPF@OPT1) contains the command set.

**Figure 13: Command set: Activate QPF options**

```
QPF_OPTIONS IMSID=*,TYPE=SET,ACTIVE=YES  
MONITOR_PHASE  
PROTECT_PHASE  
OVERFLOW_PHASE  
END
```
Test case command set

The following figure shows a command set that was created as a test case. MAQSAMP(QPF@OPT2) contains a similar command set. For more information about the test case, see the Message Advisor for IMS User Guide.

Figure 14: Command set: Test case QPF_OPTIONS

```
QPF_OPTIONS IMSID=imsid,TYPE=SET,ACTIVE=YES,MAX_REGION=5M
MONITOR_PHASE THRESHOLD%=0,RESET%=0,
   NOTIFY=(WTO,MTO),NOTIFY%=10,
   WTOR=YES

PROTECT_PHASE THRESHOLD%=60,RESET%=55,RESET_TIME=5,
   NOTIFY=(WTO,MTO),NOTIFY%=5,NOTIFY_INTERVAL=5,
   WTOR=YES,MODE=ALL

PROCESS DESTYPE=cnt,#msgs=10,DEST=ARFR*,
   ACTION=(WARN,DEQUEUE),THRESHOLD%=62
PROCESS DESTYPE=cnt,#msgs=100,
   ACTION=(WARN,UNLOAD_DEQUEUE),THRESHOLD%=65
PROCESS DESTYPE=rsmb,%USED=10,
   NOTIFY=(useridI),ACTION=(WARN,STOP)

ENFORCE TYPE=BMP,PSB=QPF#BMP2,ACTION=NONE,%USED=0
ENFORCE TYPE=BMP,%USED=40,ACTION=FAIL
ENFORCE TYPE=BMP,%USED=20,ACTION=WARN

UNLOAD_DSN DSNAME=QPF.UNLOAD.%DEST.D%DATE.T%TIME,
   DISP=NEW,UNIT=SYSALLDA,CYLS_PRIM=1,CYLS_SEC=1

OVERFLOW_PHASE THRESHOLD%=90,RESET%=75,
   NOTIFY=(WTO,useridI),NOTIFY%=2,NOTIFY_INTERVAL=1,
   WTOR=YES,ACTION=(IWAIT)

END
```

Command descriptions

A discussion of each segment from the command set shown in follows. Each segment’s purpose is explained, and keywords used to specify options appear in parentheses.

General keywords

The General keywords are not associated with a specific subcommand or statement.
General keywords control global command functions such as identifying the targeted IMS system and specifying whether QPF is active.

The following General keywords request that QPF options be set (TYPE=SET) and activated (ACTIVE=YES) for PRD1 (IMSID=PRD1). In addition, the keywords specify that QPF can use up to 5 megabytes of private storage (MAX_REGION=5M).

```
QPF_OPTIONS IMSID=PRD1,TYPE=SET,ACTIVE=YES,MAX_REGION=5M
```

**Note**
For the MAX_REGION keyword parameter, you must specify at least 2MB of storage. Failing to specify at least 2MB of storage for the MAX_REGION parameter could cause problems because of insufficient storage.

Most of the storage that QPF uses is in the extended private area, and it is used primarily for building the record-tracking tables.

**MONITOR_PHASE keywords**

The following MONITOR_PHASE keywords request that QPF begin the Monitor Phase upon IMS initialization (THRESHOLD%=0) and issue a warning message to the WTO and the MTO (NOTIFY=(WTO,MTO)).

QPF will also issue a WTOR to the console so that users can enter manual QPF commands from the console (WTOR=YES). Each time the queue grows by 10 percent (NOTIFY%=10), QPF will reissue the warning message to the WTO and the MTO. When THRESHOLD%=0, the Monitor Phase is always active and QPF ignores the RESET% keyword.

```
MONITOR_PHASE THRESHOLD%=0,RESET%=0,
                NOTIFY=(WTO,MTO),NOTIFY%=10,
                WTOR=YES
```

**PROTECT_PHASE keywords**

The following PROTECT_PHASE keywords request that QPF issue a warning message to the WTO and the MTO (NOTIFY=(WTO,MTO)) when either the LGMSG or the SHMSG queue reaches 60 percent capacity (THRESHOLD%=60). Each time the queue grows by five percent (NOTIFY%=5) or five minutes pass (NOTIFY_INTERVAL=5), QPF will reissue the warning message to the WTO and the MTO. QPF will also issue a WTOR to the console (WTOR=YES). For any ENFORCE or PROCESS statements supplied (MODE=ALL), QPF will perform all processing actions — LOG, WARN, and PROCESS. When the queue remains below 55 percent (RESET%=55) for five minutes (RESET_TIME=5), QPF will return to the Monitor Phase.
The following PROCESS keywords request that QPF issue a warning message and then dequeue the messages (ACTION=(WARN,DEQUEUE)) if any CNT (DESTYPExCNT) destination starting with ARFR (DEST=ARFR*) has at least 10 messages (#MSGS=10). QPF will not take the actions until either the LGMSG or the SHMSG queue reaches 62 percent capacity (THRESHOLD%=62).

```
PROCESS DESTYPE=CNT,#MSGS=10,DEST=ARFR*,
   ACTION=(WARN,DEQUEUE),THRESHOLD%=62
```

Specifying a threshold percentage for each of the PROCESS statements in the command set lets QPF execute the statements gradually.

**Note**

If you specify the NOTIFY keyword for an ENFORCE or PROCESS statement, it will override any NOTIFY keyword that you specify for the PROTECT_PHASE subcommand. When you specify NOTIFY for either type of statement, it only applies to messages generated as a result of the specific ENFORCE or PROCESS statement. If you specify ACTION=WARN for an ENFORCE or PROCESS statement without also specifying the NOTIFY keyword, the NOTIFY keyword that you specify for the PROTECT_PHASE subcommand applies.

The following keywords request that QPF issue a warning message and then unload and dequeue the messages (ACTION=(WARN,UNLOAD,DEQUEUE)) if a CNT (DESTYPExCNT) has at least 100 messages queued (#MSGS=100). QPF will not take the actions until either the LGMSG or the SHMSG queue reaches 65 percent capacity (THRESHOLD%=65).

```
PROCESS DESTYPE=CNT,#MSGS=100,
   ACTION=(WARN,UNLOAD,DEQUEUE),THRESHOLD%=65
```

The following keywords request that QPF issue a warning message and stop the RSMB (ACTION=(WARN,STOP)) if an RSMB (DESTYPExRSMB) is using more than 10 percent (%USED=10) of a queue data set.

```
PROCESS DESTYPE=RSMB,%USED=10,
   NOTIFY=(userid1),ACTION=(WARN,STOP)
```

**ENFORCE keywords**

The following ENFORCE keywords request that QPF take no action (ACTION=NONE) for the BMP (TYPE=BMP) whose PSB is QPFBMP2 (PSB=QPFBMP2), regardless of the
number of messages and records on its temporary queue or on the destination queue at commit time.

```
ENFORCE TYPE=BMP,PSB=QPFBMP2,ACTION=NONE,%USED=0
```

The following keywords request that QPF fail the BMP (ACTION=FAIL) if any BMP (other than QPFBMP2 because of the first ENFORCE statement) uses more than 40 percent of either message queue data set for its temporary queue or for the destination queue at commit time (%USED=40).

```
ENFORCE TYPE=BMP,%USED=40,ACTION=FAIL
```

The %USED=40 ENFORCE statement must be in front of the third ENFORCE statement (shown below) because QPF searches ENFORCE statements for matches in the order in which you specify the statements. If you place the %USED=20 statement first, QPF will never execute the %USED=40 statement because %USED=20 will match first.

---

**Note**

The order of PROCESS and ENFORCE statements determines when, or sometimes even if, QPF processes a statement. Therefore, the order of the statements is very important. For information about how QPF processes statements, see the *Message Advisor for IMS User Guide*.

BMC Software recommends the use of multiple ENFORCE statements with escalating thresholds, which lets a series of manageable events occur before the IMS message queues reach an overflow situation. See MAQSAMP member QPF@OPT5 for an example.

The following keywords request that QPF issue a warning message (ACTION=WARN) if any BMP (other than QPFBMP2 because of the first ENFORCE statement) uses more than 20 percent of either message queue data set for its temporary queue or for the destination queue at commit time (%USED=20).

```
ENFORCE TYPE=BMP,%USED=20,ACTION=WARN
```

The first ENFORCE statement excludes BMP QPFBMP2 from these actions. The second and third ENFORCE statements issue a warning message when a BMP uses 20 percent of a queue data set, and abend the BMP when it uses 40 percent.

**UNLOAD_DSN keywords**

The following UNLOAD_DSN keywords specify the unload data set (DSNAME) to be used for the UNLOAD_DEQUEUE action requested in two of the PROCESS statements.

QPF will allocate a new data set (DISP=NEW) with one primary cylinder and one secondary cylinder. The data set name will include the current date and time.
(DSNAME=QPF.UNLOAD.%DEST.D%DATE.T%TIME) to ensure uniqueness. If the data set fills up, QPF will allocate a new data set and continue.

**OVERFLOW_PHASE keywords**

The following OVERFLOW_PHASE keywords request that QPF issue warning messages to the WTO and one user ID if the queue reaches 80 percent capacity (THRESHOLD%=80).

QPF will reissue the warning messages every one minute (NOTIFY_INTERVAL=1) or each time the queue fills by another two percent (NOTIFY%=2). QPF will issue an IWAIT for any task that attempts to insert a message to the message queue (ACTION=IWAIT). Issuing IWAITs ensures that IMS stays up so that Message Advisor can determine the problem and the appropriate manual actions to be taken. Once queue utilization drops below 75 percent (RESET%=75), the Overflow Phase ends and QPF returns to the Protect Phase.

Set QPF_OPTIONS command set

The following figure shows the sample command set for setting your own QPF_OPTIONS. MAQSAMP(QPF@OPT3) contains the command set.

**Figure 15: Sample set QPF_OPTIONS command set**
For information about modifying this command set to address your site-specific requirements, see the *Message Advisor for IMS User Guide*.

**QPF_OPTIONS command set description**

The QPF_OPTIONS command set consists of the following items:

- QPF_OPTIONS primary command
- general keywords and parameters
- subcommands and statements with associated keywords and parameters
- END command

The command set keywords are divided into the following categories:

- General keywords
- MONITOR_PHASE subcommand keywords
- PROTECT_PHASE subcommand keywords
  - ENFORCE statement keywords
  - PROCESS statement keywords
- OVERFLOW_PHASE subcommand keywords
- UNLOAD_DSN subcommand keywords

---

**Note**

Keyword and parameter descriptions are listed in alphabetical order within each category: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

---

**General keywords**

The General keywords control global QPF_OPTIONS command functions such as identifying the targeted IMS system and specifying whether QPF is active.
These keywords are not specific to any of the QPF_OPTIONS subcommands or statements.

**Note**

IMS must be active before you can specify the QPF_OPTIONS command.

### ACTIVE keyword

ACTIVE is an optional keyword for the QPF_OPTIONS command and controls the overflow protection feature and the hiperassist feature simultaneously.

To control the features individually, you can specify the PROTECT and HIPERASSIST keywords instead of the ACTIVE keyword. If you specify either of the PROTECT or HIPERASSIST keywords, you cannot specify the ACTIVE keyword.

**Note**

If you do not specify any of the three keywords upon initialization, Message Advisor activates the overflow protection feature and the hiperassist feature by default.

Table 31 on page 87 lists the ACTIVE keyword parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>starts the overflow protection feature and the hiperassist feature</td>
</tr>
<tr>
<td>NO</td>
<td>stops the overflow protection feature and the hiperassist feature</td>
</tr>
</tbody>
</table>

### DUMP keyword

DUMP is an optional keyword for the QPF_OPTIONS command.

Table 32 on page 87 lists the DUMP keyword parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON_ERROR</td>
<td>when a non-recoverable error condition occurs, QPF takes a dump and deactivates</td>
</tr>
<tr>
<td>NEVER</td>
<td>QPF never takes a dump; when a non-recoverable error condition occurs, QPF issues messages and deactivates, but does not take a dump</td>
</tr>
</tbody>
</table>
Parameter | Description
--- | ---
ON_WARNing | when there is a "recoverable" error condition where QPF would normally continue processing (e.g., syntax errors, I/O errors), QPF takes a dump and deactivates

**FORMAT keyword**

FORMAT is an optional keyword for the QPF_OPTIONS command.

*Note*

If you specify the FORMAT keyword, you must also specify TYPE=LIST.

Table 33 on page 88 lists the FORMAT keyword parameters.

**Table 33: FORMAT keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>generates a formatted report for the current QPF options</td>
</tr>
<tr>
<td>No</td>
<td>lists the current QPF_Options command set without formatting</td>
</tr>
</tbody>
</table>

**HIPER_ASM% keyword**

HIPER_ASM% is an optional keyword for the QPF_OPTIONS command and indicates the maximum percentage of total auxiliary storage slots to be used by the hipersassist feature.

Use the HIPER_ASM% keyword to prevent excessive use of auxiliary storage. Consider the following when specifying the HIPER_ASM% keyword:

- HIPER_ASM% is not valid without HIPER_USAGE=STANDARD.
- When hiperassist starts, the value specified for the HIPER_ASM% keyword is converted to a number of pages based on the total auxiliary storage slots. The smaller of the converted value and the value specified for the HIPER_PAGES keyword is used to limit hiperspace usage.

Table 34 on page 88 lists the HIPER_ASM% keyword parameter.

**Table 34: HIPER_ASM% keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5(nn)</td>
<td>the maximum percentage of total auxiliary storage slots to be used by the hiperassist feature; prevents excessive use of auxiliary storage</td>
</tr>
</tbody>
</table>
HIPERASSIST keyword

HIPERASSIST is an optional keyword for the QPF_OPTIONS command and controls the hiperassist feature independently of the overflow protection feature.

If you specify the HIPERASSIST keyword, you cannot specify the ACTIVE keyword, which controls both features.

For example, to activate the hiperassist feature without also activating the overflow protection feature, you would specify HIPERASSIST=Yes. You would not specify the ACTIVE keyword in this case.

The hiperassist feature uses hiperspace buffers to supplement the IMS queue buffer pool.

Note

If you do not specify either of the ACTIVE or HIPERASSIST keywords upon initialization, Message Advisor activates hiperassist by default.

Table 35 on page 89 lists the HIPERASSIST keyword parameters.

Table 35: HIPERASSIST keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>starts the hiperassist feature</td>
</tr>
<tr>
<td>NO</td>
<td>stops the hiperassist feature</td>
</tr>
</tbody>
</table>

HIPER_PAGES keyword

HIPER_PAGES is an optional keyword for the QPF_OPTIONS command and indicates the approximate maximum number of hiperspace pages to be allocated.

Use the HIPER_PAGES keyword to limit the amount of expanded or auxiliary storage used by the hiperassist feature. Consider the following when specifying the HIPER_PAGES keyword:

- The minimum value that you can specify is 1024 or 1K.
- The maximum value that you can specify is 1048576 or 1M.
- You can also use the HIPER_ASM% keyword to limit the number of hiperspace pages. If you specify HIPER_ASM% and HIPER_PAGES, Message Advisor enforces the smaller of the two limits.
Message Advisor enforces the limit by limiting hiperspace pages for QBLKS to approximately 1% of the maximum, and limiting hiperspace pages for each of SHMSG and LGMSG to approximately 50% of the maximum. This is only an approximate limit on expanded and auxiliary storage and does not take page and segment tables into account.

Table 36 on page 90 lists the HIPER_PAGES keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1M nnnn</td>
<td>the approximate maximum number of hiperspace pages to be allocated; limits the amount of expanded or auxiliary storage used by the hiperassist feature</td>
</tr>
</tbody>
</table>

HIPER_USAGE keyword

HIPER_USAGE is an optional keyword for the QPF_OPTIONS command and controls hiperspace usage.

Table 37 on page 90 lists the HIPER_USAGE keyword parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESO</td>
<td>only expanded storage will be used; expanded storage will not be backed by auxiliary storage (if pages are stolen for other processes, the queue data sets will be read to obtain the data)</td>
</tr>
<tr>
<td>PRIORITY</td>
<td>only expanded storage will be used; expanded storage will not be backed by auxiliary storage, but the task will be given priority to use expanded storage to limit the amount of page stealing</td>
</tr>
<tr>
<td>STANDARD</td>
<td>hiperspace buffers will use expanded storage; expanded storage will be backed by auxiliary storage (paging will be performed as necessary)</td>
</tr>
</tbody>
</table>

IMSid keyword

IMSid is a required keyword for the QPF_OPTIONS command.

Table 38 on page 91 lists the IMSid keyword parameter.
Table 38: IMSid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccccc</td>
<td>the ID of the IMS system to which the QPF options apply. A mask is allowed at start-up (in the PROCLIB data set), but not when changing options through a server.</td>
</tr>
</tbody>
</table>

**LOGcode keyword**

LOGcode is an optional keyword for the QPF_OPTIONS command.

Table 39 on page 91 lists the LOGcode keyword parameter.

Table 39: LOGcode keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EF cc</td>
<td>the code to be used for IMS log records that QPF generates. The code must be unique to your IMS system, but it can be the same log code that you specify for the Message Advisor CUStomize command.</td>
</tr>
</tbody>
</table>

**MAX_REGion keyword**

MAX_REGion is an optional keyword for the QPF_OPTIONS command.

Table 40 on page 91 lists the MAX_REGion keyword parameters.

Table 40: MAX_REGion keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnK</td>
<td>the maximum amount of storage that QPF is allowed to allocate. You must specify at least 2MB of storage. Specifying 2MB of storage supports QPF system requirements and enables QPF to monitor approximately 15,000 destinations. You must specify an additional 60 bytes of storage for each destination exceeding 15,000. QPF allocates storage as each destination begins queueing messages. If QPF reaches the storage limit, it monitors the destinations with the highest queue utilization and does not monitor the remaining destinations. The destinations that QPF does not monitor probably provide little useful information. Limiting storage to 2MB does not cause additional CPU overhead.</td>
</tr>
</tbody>
</table>
MOVEPAGE keyword

MOVEPAGE is an optional keyword for the QPF_OPTIONS command and controls whether the processor move page facility will be used to move buffers.

Table 41 on page 92 lists the MOVEPAGE keyword parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>the processor move page facility (if available) is used to move buffers</td>
</tr>
<tr>
<td>NO</td>
<td>the processor move page facility is not used to move buffers</td>
</tr>
</tbody>
</table>

NOTIFY keyword

NOTIFY is an optional keyword for the QPF_OPTIONS command and indicates the recipients of informational messages that QPF issues when it takes certain actions in the IMS control region.

You can use the keyword to control the recipients of the following items:

- messages that QPF issues for actions not related to overflow protection, such as intercepting abends caused by invalid messages
- hiperassist messages

Note

If you do not also specify the NOTIFY keyword for the MONITOR_Phase subcommand, the recipients you specify for the QPF_OPTIONS command become the default recipients for overflow protection messages.

Table 42 on page 92 lists the NOTIFY keyword parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WTO</td>
<td>QPF sends messages to the MVS console</td>
</tr>
<tr>
<td>MTO</td>
<td>QPF sends messages to the IMS master terminal</td>
</tr>
<tr>
<td>NONE</td>
<td>QPF does not issue messages</td>
</tr>
</tbody>
</table>

Note: During the Overflow Phase, the WTO receives messages intended for the MTO.
**Parameter**

tso-userids

**Description**
QPF sends messages to the specified TSO users

*Note:* If you specify a user more than once, the user receives more than one copy of each message. TSO users only receive messages if they are logged on and terminal output buffers are available.

---

**PROTECT keyword**

PROTECT is an optional keyword for the QPF_OPTIONS command and controls the overflow protection feature independently of the hiperassist feature.

If you specify the PROTECT keyword, you cannot specify the ACTIVE keyword, which controls both features.

For example, to activate the overflow protection feature without also activating the hiperassist feature, you would specify `PROTECT=Yes`. You would not specify the ACTIVE keyword in this case.

*Note*

If you do not specify either of the ACTIVE or PROTECT keywords upon initialization, Message Advisor activates overflow protection by default.

Table 43 on page 93 lists the PROTECT keyword parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>starts the overflow protection feature</td>
</tr>
<tr>
<td>NO</td>
<td>stops the overflow protection feature</td>
</tr>
</tbody>
</table>

---

**RESET keyword**

RESET is an optional keyword for the QPF_OPTIONS command. Use the RESET keyword to manually reset all or specific QPF processing phases.

*Note*

If queue usage exceeds any of the thresholds when you specify the RESET keyword, QPF immediately restarts the phase and reissues all appropriate AUTO_cmds.

Table 44 on page 94 lists the RESET keyword parameters.
Table 44: RESET keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONe</td>
<td>QPF does not reset any of the processing phases</td>
</tr>
</tbody>
</table>
| All       | QPF resets all of the processing phases  
If you specify RESET=All, QPF frees the UNLoad data set. |
| MONitor   | QPF resets only the Monitor Phase |
| OVERflow  | QPF resets only the Overflow Phase |
| PROTect   | QPF resets only the Protect Phase |

SQ_MAXQBUF keyword

SQ_MAXQBUF is an optional keyword for the QPF_OPTIONS command that applies only to a Shared Queues environment. When the IMS QBUFMAX startup parameter is set to 0, SQ_MAXQBUF is used to override the local queues QBUF value used to calculate the high RRNs for the Short and Large messages. The high RRNs are key to determining when to invoke the QPF Monitor, Protect and Overflow Phases. If this parameter is not specified, and QBUFMAX is set to 0, QPF will calculate the high RRN values using either the current numbers of local Queue buffers allocated, or 9999, the higher of the two values. SQ_MAXQBUF can be set to any value between 10000 and 9999999.

TRACE keyword

TRACE is an optional keyword for the QPF_OPTIONS command.

Table 45 on page 94 lists the TRACE keyword parameters.

Table 45: TRACE keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| INternal  | QPF uses an internal trace table  
**Note:** QPF uses a 100-entry internal trace table by default. You must specify the TRACE_SIZE keyword to use a larger table. |
| All       | QPF uses both an internal and external trace table |
| EXternal  | QPF uses an external GTF trace; the external trace uses an x'039' GTF event ID, the same as Message Advisor |
| NONe      | QPF does not perform any tracing |
TRACE_SIZE keyword

TRACE_SIZE is an optional keyword for the QPF_OPTIONS command.

Table 46 on page 95 lists the TRACE_SIZE keyword parameter.

Table 46: TRACE_SIZE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>the number of trace table entries for the internal trace table</td>
</tr>
<tr>
<td>nnnn</td>
<td>You can specify a value from 100 to 99,999. If you do not specify a value, QPF uses 100 by default.</td>
</tr>
</tbody>
</table>

TYPE keyword

TYPE is an optional keyword for the QPF_OPTIONS command.

Table 47 on page 95 lists the TYPE keyword parameter.

Table 47: TYPE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET</td>
<td>allows you to set or change the QPF options</td>
</tr>
<tr>
<td></td>
<td>If you specify any subcommand or any keywords other than FORMat, SET is the default. If you specify any subcommand, the options that you choose for the QPF_OPTIONS subcommands replace the options currently in effect for all subcommands. If you do not specify subcommands, the new general keyword values replace the current general options, but the subcommand options remain unchanged. If you specify TYPE=SET in the QPFOPTS data set with no subcommands, QPF ignores the command set.</td>
</tr>
<tr>
<td>LIST</td>
<td>lists the current QPF options</td>
</tr>
<tr>
<td></td>
<td>If you specify TYPE=LIST, you cannot use any subcommands and FORMat is the only valid keyword. TYPE=LIST is the default if you do not specify any subcommands or any keywords other than FORMat. If you specify TYPE=LIST in the QPFOPTS data set with no subcommands, QPF ignores the command set.</td>
</tr>
<tr>
<td>LOAD</td>
<td>reloads the QPF options from the PROCLIB data set</td>
</tr>
<tr>
<td></td>
<td>If you specify TYPE=LOAD, you cannot specify any subcommands. You can specify other general keywords. If you specify other keywords, the keyword values override the QPFOPTS data set values. If you specify TYPE=LOAD in the QPFOPTS data set, QPF ignores the command set.</td>
</tr>
</tbody>
</table>
**Parameter** | **Description**
--- | ---
UNLoad | deactivates QPF in the IMS control region.

If you later load QPF by specifying `TYPE=LOAD`, Message Advisor reloads all QPF load modules and you can apply fixes to QPF without restarting IMS. If you specify `TYPE=UNLoad`, you cannot specify any other keywords or subcommands. If you specify `TYPE=UNLoad` in the QPF_OPTS data set, QPF ignores the command set.

---

**MONITOR_PHASE keywords**

Before you can set or replace the QPF options, the phase subcommands listed below must be present in the QPF_OPTIONS command set:

- MONITOR_PHASE
- PROTECT_PHASE
- OVERFLOW_PHASE

---

**Note**

IMS must be active before you can specify the QPF_OPTIONS command.

The keywords described in this section apply to the Monitor Phase.

**AUTO_cmd keyword**

AUTO_cmd is an optional keyword for the MONITOR_PHASE subcommand. The AUTO_cmd keyword allows further flexibility in performing automated actions. You can use this keyword to specify an MVS operator command that QPF will issue each time the phase starts.

Table 48 on page 96 lists the AUTO_cmd keyword parameter.

**Table 48: AUTO_cmd keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cmd</code></td>
<td>QPF issues the specified MVS operator command each time the Monitor Phase starts</td>
</tr>
</tbody>
</table>

**AUTODQ_MASTER keyword**

AUTODQ_MASTER is an optional keyword for the MONITOR_PHASE subcommand.
The AUTODQ_MASTER keyword allows DEQUEUEs for the IMS Master Terminal destination to be scheduled automatically when the number of messages queued to that terminal exceeds \textit{n}nnnn where \textit{n}nnnn must be in the range of 100 to 32767.

The automatic DEQUEUEs will be for the entire destination and will remove all the messages found for that destination at the time the DEQUEUEs execute.

Table 49 on page 97 lists the AUTODQ_MASTER keyword parameter.

Table 49: AUTODQ_MASTER keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{n}nnnn</td>
<td>QPF allows DEQUEUEs for the IMS Master Terminal destination to be scheduled automatically when the number of messages queued to that terminal exceeds \textit{nnnnn} where \textit{nnnnn} must be in the range of 100 to 32767</td>
</tr>
</tbody>
</table>

**AUTODQ_SECONDARY keyword**

AUTODQ_SECONDARY is an optional keyword for the MONITOR_PHASE subcommand.

The AUTODQ_SECONDARY keyword allows DEQUEUEs for the IMS Secondary Master Terminal destination, if any, to be scheduled automatically when the number of messages queued to that terminal exceeds \textit{nnnnn} where \textit{nnnnn} must be in the range of 100 to 32767.

The automatic DEQUEUEs will be for the entire destination and will remove all the messages found for that destination at the time the DEQUEUEs execute.

Table 50 on page 97 lists the AUTODQ_SECONDARY keyword parameter.

Table 50: AUTODQ_SECONDARY keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{nnnnn}</td>
<td>QPF allows a DEQUEUE for the IMS Secondary Master Terminal destination, if any, to be scheduled automatically when the number of messages queued to that terminal exceeds \textit{nnnnn} where \textit{nnnnn} must be in the range of 100 to 32767</td>
</tr>
</tbody>
</table>

**NOTIFY keyword**

NOTIFY is an optional keyword for the MONITOR_PHASE subcommand and indicates the recipients of informational messages that QPF issues when the Monitor Phase starts and stops.
The default is the value specified for the NOTIFY keyword on the QPF_OPTIONS command.

Table 42 on page 92 lists the NOTIFY keyword parameters.

Table 51: NOTIFY keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTO</td>
<td>QPF sends messages to the IMS master terminal when the Monitor Phase starts and stops</td>
</tr>
<tr>
<td>NONE</td>
<td>QPF does not issue messages</td>
</tr>
<tr>
<td>WTO</td>
<td>QPF sends messages to the MVS console when the Monitor Phase starts and stops</td>
</tr>
<tr>
<td>tso-userids</td>
<td>QPF sends messages to the specified TSO users when the Monitor Phase starts and stops</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>If you specify a user more than once, the user receives more than one copy of each message. TSO users only receive messages if they are logged on and terminal output buffers are available.</td>
</tr>
</tbody>
</table>

**NOTIFY% keyword**

NOTIFY% is an optional keyword for the MONITOR_PHASE subcommand.

Table 52 on page 98 lists the NOTIFY% keyword parameters.

Table 52: NOTIFY% keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>QPF issues the NOTIFY message only when the Monitor Phase starts and stops, or as specified for the NOTIFY_INTerval keyword</td>
</tr>
<tr>
<td>nn</td>
<td>QPF reissues the NOTIFY message when the maximum of the percent full values for the message queue data sets changes by nn percentage points</td>
</tr>
</tbody>
</table>

**NOTIFY_INTerval keyword**

NOTIFY_INTerval is an optional keyword for the MONITOR_PHASE subcommand.

Table 53 on page 99 lists the NOTIFY_INTerval keyword parameters.
Table 53: NOTIFY_INTERVAL keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>QPF issues the NOTIFY message only when the Monitor Phase starts and stops, or as specified for the NOTIFY% keyword</td>
</tr>
<tr>
<td>nn</td>
<td>QPF allows nn minutes between NOTIFY messages; indicates that QPF should issue a NOTIFY message at least this often</td>
</tr>
</tbody>
</table>

**RESET% keyword**

RESET% is an optional keyword for the MONITOR_PHASE subcommand. The value is a percentage that indicates when the Monitor Phase should end. The phase ends when both message queue data sets (SHMSG and LGMSG) remain below the RESET% value for the length of time specified for the RESET_TIME keyword.

Table 54 on page 99 lists the RESET% keyword parameters.

Table 54: RESET% keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>the Monitor Phase ends when the message queue data sets remain below the percentage value specified for the THReshold% keyword for the length of time specified for the RESET_TIME keyword</td>
</tr>
<tr>
<td>nn</td>
<td>the Monitor Phase ends when the message queue data sets remain below this percentage value for the length of time specified for the RESET_TIME keyword. The percentage value must be less than or equal to the THReshold% keyword value.</td>
</tr>
<tr>
<td>0</td>
<td>the Monitor Phase remains active until you manually terminate it using the QPF_Options command</td>
</tr>
</tbody>
</table>

**RESET_TIME keyword**

RESET_TIME is an optional keyword for the MONITOR_PHASE subcommand. The value is a number of minutes that is used to determine when the Monitor Phase should end. The phase ends when both message queue data sets (SHMSG and LGMSG) remain below the RESET% value for the number of minutes specified.

Note

It is possible for a phase to reach the reset threshold, begin to shut down, and then reach the start-up threshold before shut down completes. In this case, stopped messages are not received, but all normal start-up processing occurs (including re-issuing the AUTO_cmd).

Table 55 on page 100 lists the RESET_TIME keyword parameter.
Table 55: RESET_TIME keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| 15 nn     | the Monitor Phase ends when the message queue data sets remain below the RESET% value for the number of minutes specified.  
Note: When THReshold% equals RESET%, Message Advisor forces the RESET_TIME to at least 15 minutes. |

**THReshold% keyword**

THReshold% is an optional keyword for the MONITOR_PHASE subcommand. The value is a percentage that indicates when the Monitor Phase should begin. The phase begins when either message queue data set (SHMSG or LGMSG) reaches the percentage value.

**Note**
The Monitor Phase threshold must be less than or equal to the Protect Phase threshold.

If a phase is not already active, it begins automatically when IMS reaches the IMS SHUTDWN value.

Table 56 on page 100 lists the THReshold% keyword parameters.

Table 56: THReshold% keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>the Monitor Phase starts when either message queue data set reaches the IMS QTL value. Message Advisor adjusts the value downward if you set the Protect Phase or the Overflow Phase to a lower threshold.</td>
</tr>
<tr>
<td>nn up to 99</td>
<td>the Monitor Phase starts when either message queue data set reaches this percentage value</td>
</tr>
</tbody>
</table>
Parameter | Description
--- | ---
0 | the Monitor Phase is continuously active

**Note:** In a production environment, BMC Software recommends that you specify `THReshold%=0`. Specifying a value greater than 0 could impact IMS performance while QPF determines the record count of all active destinations. For example, if you specify `THReshold%= x`, `RESET=0`, performance could be impacted in proportion to the following factors:

- the value of `x`
- the number of active destinations
- the activity level for active destinations

---

**WTO_Desc keyword**

WTO_Desc is an optional keyword for the MONITOR_PHASE subcommand.

Table 57 on page 101 lists the WTO_Desc keyword parameter.

**Table 57: WTO_Desc keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>X’cccc’</code></td>
<td>the descriptor code to be used when you specify <code>NOTIFY=WTO</code> or <code>WTOR=YES</code>&lt;br&gt;The default is the value that is used by the IMS control region.</td>
</tr>
</tbody>
</table>

---

**WTO_Routcde keyword**

WTO_Routcde is an optional keyword for the MONITOR_PHASE subcommand.

Table 58 on page 101 lists the WTO_Routcde keyword parameter.

**Table 58: WTO_Routcde keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>X’cccc’</code></td>
<td>the routing code to be used when you specify <code>NOTIFY=WTO</code> or <code>WTOR=YES</code>&lt;br&gt;The default is the value that is used by the IMS control region.</td>
</tr>
</tbody>
</table>
WTOR keyword

WTOR is an optional keyword for the MONITOR_PHASE subcommand and indicates whether QPF issues a WTOR when the Monitor Phase starts. If QPF issues a WTOR, you can manually initiate QPF actions.

Table 59 on page 102 lists the WTOR keyword parameters.

Table 59: WTOR keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>QPF issues a WTOR when the Monitor Phase starts</td>
</tr>
<tr>
<td>No</td>
<td>QPF does not issue a WTOR when the Monitor Phase starts</td>
</tr>
</tbody>
</table>

PROTECT_PHASE keywords

Before you can set or replace the QPF options, the phase subcommands listed below must be present in the QPF_OPTIONS command set:

- MONITOR_PHASE
- PROTECT_PHASE
- OVERFLOW_PHASE

Note
IMS must be active before you can specify the QPF_OPTIONS command.

The keywords in this section apply to the Protect Phase.

AUTO_cmd keyword

AUTO_cmd is an optional keyword for the PROTECT_PHASE subcommand. The AUTO_cmd keyword allows further flexibility in performing automated actions. You can use this keyword to specify an MVS operator command that QPF will issue each time the phase starts.

Table 60 on page 103 lists the AUTO_cmd keyword parameter.
Table 60: AUTO_cmd keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'cmd'</td>
<td>QPF issues the specified MVS operator command each time the Protect Phase starts</td>
</tr>
</tbody>
</table>

**MODe keyword**

MODe is an optional keyword for the PROTECT_PHASE subcommand and indicates the action processing mode for ENForce and/or PROCess statements.

You can specify All, NONe, or a list of values.

Table 61 on page 103 lists the MODe keyword parameters.

Table 61: MODe keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>QPF performs and logs actions, repeats if necessary, and issues warning messages</td>
</tr>
<tr>
<td>LOG</td>
<td>QPF writes an IMS log record, but does not take action</td>
</tr>
<tr>
<td>NONe</td>
<td>QPF ignores ENForce and PROCess statements. You can specify MODe=NONE to turn off processing without removing the statements</td>
</tr>
<tr>
<td>PROCESS</td>
<td>QPF performs and logs requested actions, but does not issue warning messages; implies MODe=LOG</td>
</tr>
<tr>
<td>REPEAT</td>
<td>indicates that QPF may repeat actions if necessary; implies MODe=PROCESS</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> QPF can repeat DEQueue and UNLoad actions, if necessary. QPF never repeats an action if the first action fails. In most cases, repeating successful actions is desirable. In some cases, repetition could cause an undesirable looping situation. If you do not want QPF to repeat successful actions, do not specify MODe=REPEAT.</td>
</tr>
<tr>
<td>WARN</td>
<td>QPF sends warning messages to the recipients specified for the NOTIFY keyword, but does not take action</td>
</tr>
</tbody>
</table>

**NOTIFY keyword**

NOTIFY is an optional keyword for the PROTECT_PHASE subcommand and indicates the recipients of informational messages that QPF issues when the Protect Phase starts and stops. The message includes the phase name and information regarding message queue data set usage.

QPF issues a stop message when the criteria for stopping the phase have been met, but a processing task remains active. QPF does not stop the phase until it completes all processing tasks.
QPF also sends messages when it takes any processing action. The processing messages include the destination or origin affected, the action taken, and the relative number of the ENForce or PROCess statement that caused the action. The NOTIFY keyword for specific ENForce and/or PROCess statements may override the NOTIFY keyword for the Protect Phase subcommand.

The default is the value specified for the NOTIFY keyword on the MONITOR_PHASE subcommand.

Table 62 on page 104 lists the NOTIFY keyword parameters.

**Table 62: NOTIFY keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTO</td>
<td>QPF sends messages to the IMS master terminal when the Protect Phase starts and stops</td>
</tr>
<tr>
<td>NONE</td>
<td>QPF does not issue messages</td>
</tr>
<tr>
<td>WTO</td>
<td>QPF sends messages to the MVS console when the Protect Phase starts and stops</td>
</tr>
<tr>
<td>tso-userids</td>
<td>QPF sends messages to the specified TSO users when the Protect Phase starts and stops</td>
</tr>
<tr>
<td><strong>Note:</strong> If you specify a user more than once, the user receives more than one copy of each message. TSO users only receive messages if they are logged on and terminal output buffers are available.</td>
<td></td>
</tr>
</tbody>
</table>

**NOTIFY% keyword**

NOTIFY% is an optional keyword for the PROTECT_PHASE subcommand.

Table 63 on page 104 lists the NOTIFY% keyword parameters.

**Table 63: NOTIFY% keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>QPF reissues the NOTIFY message when the maximum of the percent full values for the message queue data sets changes by the value specified for the Monitor Phase NOTIFY% keyword</td>
</tr>
<tr>
<td>0</td>
<td>QPF issues the NOTIFY message only when the Protect Phase starts and stops, or as specified for the NOTIFY_INTERVAL keyword</td>
</tr>
<tr>
<td>nn</td>
<td>QPF reissues the NOTIFY message when the maximum of the percent full values for the message queue data sets changes by nn percentage points</td>
</tr>
</tbody>
</table>
**NOTIFY_INTerval keyword**

NOTIFY_INTerval is an optional keyword for the PROTECT_PHASE subcommand.

Table 64 on page 105 lists the NOTIFY_INTerval keyword parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>QPF issues messages according to the value specified for the Monitor Phase NOTIFY_INTerval keyword</td>
</tr>
<tr>
<td>0</td>
<td>QPF issues the NOTIFY message only when the Protect Phase starts and stops, or as specified for the NOTIFY% keyword</td>
</tr>
<tr>
<td>nn</td>
<td>QPF allows nn minutes between NOTIFY messages; indicates that QPF should issue a NOTIFY message at least this often</td>
</tr>
</tbody>
</table>

**RESET% keyword**

RESET% is an optional keyword for the PROTECT_PHASE subcommand. The value is a percentage that indicates when the Protect Phase should end. The phase ends when both message queue data sets (SHMSG and LGMSG) remain below the RESET% value for the length of time specified for the RESET_TIME keyword.

Table 65 on page 105 lists the RESET% keyword parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>the Protect Phase ends when the message queue data sets remain below the percentage value specified for the Monitor Phase THReshold% keyword, for the length of time specified for the Monitor Phase RESET_TIME keyword</td>
</tr>
<tr>
<td>nn</td>
<td>the Protect Phase ends when the message queue data sets remain below this percentage value for the length of time specified for the RESET_TIME keyword. The percentage value must be less than or equal to the THReshold% keyword value.</td>
</tr>
<tr>
<td>0</td>
<td>the Protect Phase remains active until you manually terminate it using the QPF_Options command</td>
</tr>
</tbody>
</table>

**RESET_TIME keyword**

RESET_TIME is an optional keyword for the PROTECT_PHASE subcommand. The value is a number of minutes that is used to determine when the Protect Phase should end. The phase ends when both message queue data sets (SHMSG and LGMSG) remain below the RESET% value for the number of minutes specified.
Note

It is possible for a phase to reach the reset threshold, begin to shut down, and then reach the start-up threshold before shut down completes. In this case, stopped messages are not received, but all normal start-up processing occurs (including re-issuing the AUTO_cmd).

Table 66 on page 106 lists the RESET TIME keyword parameter.

Table 66: RESET TIME keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1 nn      | the Protect Phase ends when the message queue data sets remain below the RESET% value for the number of minutes specified  
Note: When THreshold% equals RESET%, Message Advisor forces the RESET TIME to at least one minute. |

THreshold% keyword

THreshold% is an optional keyword for the PROTECT PHASE subcommand. The value is a percentage that indicates when the Protect Phase should begin. The phase begins when either message queue data set (SHMSG or LGMSG) reaches the percentage value.

Note

QPF may delay specific actions until the message queue data sets reach a higher threshold. Regardless of the reset keywords, QPF will not initiate any new protective actions when the queues drop below the THreshold%. QPF completes actions that are currently in progress before terminating the Protect Phase, regardless of the reset keywords.

If a phase is not already active, it begins automatically when IMS reaches the IMS SHUTDWN value.

Table 67 on page 106 lists the THreshold% keyword parameters.

Table 67: THreshold% keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>the Protect Phase starts when either message queue data set reaches the IMS QTU value; Message Advisor adjusts the value downward if you set the Overflow Phase to a lower threshold</td>
</tr>
<tr>
<td>nn up to 99</td>
<td>the Protect Phase starts when either message queue data set reaches this percentage value</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>the Protect Phase is continuously active</td>
</tr>
</tbody>
</table>

**WTO_Desc keyword**

WTO_Desc is an optional keyword for the PROTECT_PHASE subcommand.

Table 68 on page 107 lists the WTO_Desc keyword parameter.

**Table 68: WTO_Desc keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'cccc'</td>
<td>the descriptor code to be used when you specify NOTIFY=WTO or WTOR=YES</td>
</tr>
<tr>
<td></td>
<td>The default is the value that is used by the IMS control region.</td>
</tr>
</tbody>
</table>

**WTO_Routcde keyword**

WTO_Routcde is an optional keyword for the PROTECT_PHASE subcommand.

Table 69 on page 107 lists the WTO_Routcde keyword parameter.

**Table 69: WTO_Routcde keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'cccc'</td>
<td>the routing code to be used when you specify NOTIFY=WTO or WTOR=YES</td>
</tr>
<tr>
<td></td>
<td>The default is the value that is used by the IMS control region.</td>
</tr>
</tbody>
</table>

**WTOR keyword**

WTOR is an optional keyword for the PROTECT_PHASE subcommand and indicates whether QPF issues a WTOR when the Protect Phase starts. If QPF issues a WTOR, you can manually initiate QPF actions.

Table 70 on page 107 lists the WTOR keyword parameters.

**Table 70: WTOR keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>if you specified a value for the Monitor Phase WTOR keyword, that value remains in effect by default</td>
</tr>
<tr>
<td>No</td>
<td>QPF does not issue a WTOR when the Protect Phase starts</td>
</tr>
</tbody>
</table>
ENForce statement keywords

ENForce statements identify automatic limits that QPF implements during the Protect Phase, and also specify the conditions under which QPF allows application programs and input devices to insert messages. The statements only apply when queue usage exceeds the Protect Phase threshold.

The order of the ENForce statements is meaningful. If a program or transaction matches more than one statement, QPF only applies the first statement. Therefore, you should place the most restrictive statement with the highest threshold first.

Within ENForce statements, the JOBNAME, PSB, and TRANsaction keywords identify application programs (TYPE=MPP or TYPE=BMP). The NAME and NODE keywords identify input devices (other TYPE parameters). You cannot mix these keywords on the same statement.

QPF applies an ENForce statement when any of the %USED, #RECords, #Msgs, or TEMP_records keywords match. That is, there is a logical OR between those keywords, and each ENForce statement requires at least one of those keywords. All other selection and utilization keywords must match for the statement to match. That is, there is a logical AND between all other keywords.

A statement with the TEMP_records keyword applies to temporary queues only. A statement with the #RECords keyword or the #Msgs keyword applies to permanent destination queues only. A statement with the %USED keyword applies to both temporary and destination queues.

For statements that apply to permanent destination queues, you can further qualify the destination queues to which the statements apply. You can choose from the following pairs of destination-related keywords:

- DESTination and DESTYpe
- TPName and LUname
- TMEMBER and TPIPE

ABEND_code keyword

ABEND_code is an optional keyword for the ENForce statement.

Table 71 on page 109 lists the ABEND_code keyword parameter.
### Table 71: ABEND_code keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>474 nnnn</td>
<td>the pseudo-abend code to be used by the ABEND action</td>
</tr>
</tbody>
</table>

### ACTion keyword

ACTion is an optional keyword for the ENForce statement. The ACTion keyword determines what action QPF takes when application programs or input devices match the statement.

QPF takes action when queue usage exceeds the Protect Phase threshold and the application program or input device tries to insert a message. You can specify multiple actions. If the actions are incompatible, QPF takes the least intrusive action on the first insert attempt, and takes the more intrusive action on subsequent insert attempts.

QPF takes actions in the following order: WARN, STOP, FAIL, ABEND, and IWAIT.

**Note**

QPF does not take actions when queue usage falls below the Protect Phase threshold, even though the Protect Phase remains in effect. Therefore, when queue usage falls below the Protect Phase threshold, QPF will not IWAIT or ABEND new programs. If QPF has already placed programs in an IWAIT, the programs remain in a wait until the Protect Phase ends.

Table 72 on page 109 lists the ACTion keyword parameters.

### Table 72: ACTion keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WARN</td>
<td>QPF sends a message to the recipients specified for the NOTIFY keyword. QPF only sends the message on the first insert attempt. All other actions (except NONE) imply WARN.</td>
</tr>
<tr>
<td>ABEND</td>
<td>QPF abends the application. If the abend fails, QPF places the application in an IWAIT on the next insert attempt. The ABEND action is not valid for input devices.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>FAIL</td>
<td>QPF returns a status code (indicated by the STATUS_code keyword) to the application, and the application does not insert the message. QPF repeats the FAIL action each time the application tries to insert a message, unless you also specify ABEND or IWAIT. If you specify ABEND or IWAIT, QPF only returns the status code on the first insert attempt and takes the other action on the next insert attempt. FAIL causes deallocation of APPC input devices. When OTMA attempts to insert a record for a TPIPE and an ENFORCE statement with a FAIL action matches, QPF rejects the current input message and sends either a NAK or a DFS1289 message to the client. The NAK or message indicates that no queue space is available. In addition, all future input messages for the TPIPE will fail, even if they do not match any ENFORCE statement. This action continues until the PROTECT phase ends, a different ENFORCE statement matches (with certain exceptions), or an IPOST action is taken. The IPOST action cancels the FAIL action and allows subsequent messages to proceed without any action until the end of the current rate interval.</td>
</tr>
<tr>
<td>IWAIT/WAIT</td>
<td>QPF places the application or input device in a wait state until the Protect Phase terminates. <strong>Note:</strong> You can use ACTION=WAIT as an alias for ACTION=IWAIT. When a statement matches at a commit point or enqueue, QPF does not take the IWAIT action until the next insert attempt so that the syncpoint can complete. When OTMA attempts to insert a record for a TPIPE, and an ENFORCE statement with an IWAIT action matches, QPF takes the IWAIT action for that ITASK immediately. QPF also takes the IWAIT action for any other OTMA ITASKs for the TPIPE when those ITASKs attempt to insert their next record, even if no match occurs on an ENFORCE statement for that insert. This process continues until the PROTECT phase ends, a different ENFORCE statement matches (with certain exceptions), or an IPOST action is taken. In addition, when QPF takes the IWAIT action for the first ITASK, QPF automatically issues the /STOP TMEM x TPIPE x command to stop the TPIPE. This action causes IMS to stop creating new ITASKs for the TPIPE. Input and output remain suspended until the TPIPE is restarted. In the overflow phase, QPF stops TPIPEs by directly updating the control block, since issuing commands can cause problems when queue space is short. In most cases, QPF automatically issues the /START command to restart the TPIPEs when the TPIPE is POSTed, the protect phase ends, or QPF is unloaded. In case of an IMS shutdown or a QPF abend, QPF restarts the TPIPE by directly modifying the control block. In some cases (such as an IMS abend followed by an emergency restart or an XRF takeover), QPF cannot restart the TPIPE. The TPIPE will remain stopped until you manually start it by using the /START TMEM x TPIPE x command. However, if QPF takes the IWAIT action for another ITASK for the TPIPE, QPF will stop the TPIPE again.</td>
</tr>
<tr>
<td>NONe</td>
<td>QPF does not take action during the Protect Phase</td>
</tr>
</tbody>
</table>
### DESTination keyword

DESTination is an optional keyword for the ENForce statement.

Table 73 on page 111 lists the DESTination keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| mm...mm or (mm...mm,...) up to 64 characters for TPNames | a destination name or mask, or a list of destination names or masks enclosed in parentheses
For APPC destinations, the name is the TPName. For OTMA destinations, the name is the TMEMBER.
The ENForce statement applies only when an application program or input device enqueues one or more messages to the destination(s) specified. |

### DESTYpe/DTYPe keyword

DESTYpe is an optional keyword for the ENForce statement.

You can use DTYPe as an alias for DESTYpe.

Some of the DESTYpe parameters include other DESTYpe parameters:
LTERM includes:  
<table>
<thead>
<tr>
<th>CNT includes:</th>
<th>STATic</th>
<th>DYNAMic</th>
<th>DEADQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSName</td>
<td>VSPcnt</td>
<td>APPC</td>
<td>OTMA</td>
</tr>
</tbody>
</table>

TRANSACTION includes:  
<table>
<thead>
<tr>
<th>SMB includes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP</td>
</tr>
<tr>
<td>RSMB</td>
</tr>
</tbody>
</table>

Table 74 on page 112 lists the DESTYpe keyword parameters.

Table 74: DESTYpe keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All/A</td>
<td>a destination type, or a list of destination types</td>
</tr>
<tr>
<td>BMP</td>
<td>The ENForce statement applies only when an application program or input device enqueues one or more messages to destinations of the type(s) specified. You can specify a list of values. For example, DESTYpe=(APPC,OTMA) matches for either APPC or OTMA destinations. To exclude a type (except for All), add NO in front of the applicable parameter. For example, DESTYpe=NOBMP.</td>
</tr>
<tr>
<td>CNT</td>
<td></td>
</tr>
<tr>
<td>DEADQ</td>
<td></td>
</tr>
<tr>
<td>DYNAMic/ VIRTual/ETO/VTF/CNT-E/CNT-V</td>
<td></td>
</tr>
<tr>
<td>LTERM:s</td>
<td></td>
</tr>
<tr>
<td>MPP</td>
<td></td>
</tr>
<tr>
<td>MSName</td>
<td></td>
</tr>
<tr>
<td>OTMA</td>
<td></td>
</tr>
<tr>
<td>RSMB</td>
<td></td>
</tr>
<tr>
<td>SMB</td>
<td></td>
</tr>
<tr>
<td>STATic</td>
<td></td>
</tr>
<tr>
<td>SYSTEM</td>
<td></td>
</tr>
<tr>
<td>TPName/APPC</td>
<td></td>
</tr>
<tr>
<td>TRANsaction</td>
<td></td>
</tr>
<tr>
<td>VSPcnt</td>
<td></td>
</tr>
</tbody>
</table>

JOBNAME keyword

JOBNAME is an optional keyword for the ENForce statement.
Table 75 on page 113 lists the JOBNAME keyword parameter.

Table 75: JOBNAME keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mmmmmmmmm or (mmmmmmmm, mmmmmmm..)</td>
<td>a job name or mask, or a list of job names and masks enclosed in parentheses. The ENForce statement applies only to BMP regions, MPP regions, and Message Advisor servers that match the jobname(s) specified.</td>
</tr>
</tbody>
</table>

**LUname keyword**

LUname is an optional keyword for the ENForce statement.

Consider the following when specifying the LUname keyword:

- You can specify a list of LU names or patterns for the LUname keyword value.
- You can use LUname and TPName together to identify a specific APPC destination.
- Message Advisor assumes DESTYpe=TPName when you specify the LUname keyword.
- An LU name can include a network ID (netid.luname).

Table 76 on page 113 lists the LUname keyword parameter.

Table 76: LUname keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmm. mmmmmmmmm,...)</td>
<td>a name or mask, or a list of names or masks enclosed in parentheses. The ENForce statement applies only when an application program or input device enqueues one or more messages to APPC destinations that match the LUname(s) specified. If you specify the LUname keyword, QPF assumes DESTYpe=TPName.</td>
</tr>
</tbody>
</table>

**NAME keyword**

NAME is an optional keyword for the ENForce statement.

Table 77 on page 114 lists the NAME keyword parameter.
Table 77: NAME keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mmmmmmmmmm or (mmmmmmmmnm, mmmmmmmmmn..)</td>
<td>a name or mask, or a list of names and masks enclosed in parentheses. The ENForce statement applies only to input devices with LTERM names, APPC LU names, or OTMA TPIPE names that match the name(s) specified.</td>
</tr>
</tbody>
</table>

**NODE keyword**

NODE is an optional keyword for the ENForce statement.

Table 78 on page 114 lists the NODE keyword parameter.

Table 78: NODE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mmmmmmmmmm or (mmmmmmmmnm, mmmmmmmmmn..)</td>
<td>a node name or mask, or a list of node names or masks enclosed in parentheses. The ENForce statement applies only to input devices with node names that match the name(s) specified.</td>
</tr>
</tbody>
</table>

**NOTIFY keyword**

NOTIFY is an optional keyword for the ENForce statement.

The default is the value specified for the NOTIFY keyword on the PROTECT_PHASE subcommand.

Table 79 on page 114 lists the NOTIFY keyword parameters.

Table 79: NOTIFY keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTO</td>
<td>QPF sends messages to the IMS master terminal when the ENForce statement takes effect</td>
</tr>
<tr>
<td>NONE</td>
<td>QPF does not issue messages</td>
</tr>
<tr>
<td>WTO</td>
<td>QPF sends messages to the MVS console when the ENForce statement takes effect</td>
</tr>
</tbody>
</table>
Parameter | Description
--- | ---
*tso-userids* | QPF sends messages to the specified TSO users when the ENForce statement takes effect

*Note:* If you specify a user more than once, the user receives more than one copy of each message. TSO users only receive messages if they are logged on and terminal output buffers are available.

### OTMA_LIMIT keyword

OTMA_LIMIT is an optional keyword for the ENForce statement.

When an ENForce statement with OTMA_LIMIT matches a TPIPE, QPF reduces the limit for input messages for a TMEMBER to slightly less than the current number of messages. This action stops new messages from being created until existing messages are processed. The limit applies to the entire TMEMBER, not just the TPIPE. You can also change this limit by using the IMS /START TMEM *name* INPUT *count* command. The OTMA_LIMIT keyword specifies the minimum acceptable limit, but QPF will not necessarily lower the limit to the minimum.

For example, the default limit is 5000. You specify `OTMA_LIMIT=500` on an ENForce statement that matches when 923 input messages are being processed. QPF will lower the limit to 920, attempting to maintain the load at about the level that caused the ENForce statement to match. If the ENForce matches when 274 input messages exist, QPF will lower the limit to 500. This limit has no immediate effect, but it will be the new minimum allowed by the ENForce statement.

QPF will never raise the limit, even when the protect phase ends. The intent is to keep input at a level that can be maintained without flooding the queues. You can use the /START TMEM command to raise the limit. This command can specify a limit of 200 through 65000. The OTMA_LIMIT keyword allows a range of 1 to 65000, but BMC recommends against setting the limit to a very low number. A low limit will cause IMS to go in and out of "flood" state too often. (Specifying `OTMA_LIMIT=1` causes IMS to go in and out of "flood" state with every message.)

Table 80 on page 115 lists the OTMA_LIMIT keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>nmmnn</em> (1 through 65000)</td>
<td>Reduce the limit for input messages for a TMEMBER to slightly less than the current number of messages, but never less than <em>nmmnn</em>.</td>
</tr>
</tbody>
</table>

### PSB keyword

PSB is an optional keyword for the ENForce statement.
Table 81 on page 116 lists the PSB keyword parameter.

### Table 81: PSB keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>mmmmmmmm</code> or <code>(mmmmmmmm, mmmmmmm,..)</code></td>
<td>a PSB name or mask, or a list of PSB names or masks enclosed in parentheses. The ENForce statement applies only to application programs that match the PSB name(s) specified.</td>
</tr>
</tbody>
</table>

**SEGNO keyword**

SEGNO is an optional keyword for the ENForce statement.

Table 82 on page 116 lists the SEGNO keyword parameter.

### Table 82: SEGNO keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>nnnn</code></td>
<td>the SEGNO value to be set for transactions that match the ENForce statement; the values will be changed permanently</td>
</tr>
</tbody>
</table>

**SEGSIZE keyword**

SEGSIZE is an optional keyword for the ENForce statement.

Table 83 on page 116 lists the SEGSIZE keyword parameter.

### Table 83: SEGSIZE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>nnnn</code></td>
<td>the SEGSIZE value to be set for transactions that match the ENForce statement; the values will be changed permanently</td>
</tr>
</tbody>
</table>

**STATUS_code keyword**

STATUS_code is an optional keyword for the ENForce statement.

Table 84 on page 117 lists the STATUS_code keyword parameters.
Table 84: STATUS_code keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A7</td>
<td>the status code to be used by the FAIL action</td>
</tr>
<tr>
<td>A6</td>
<td>Note: You can specify any two-character code, but BMC Software recommends A7 (SEGNO exceeded) and A6 (SEGSIZE exceeded) because application programs handle them more easily.</td>
</tr>
</tbody>
</table>

TEMp_records keyword

The ENForce statement requires one of the following keywords: #Msgs, #RECords, %USED, or TEMp_records.

Table 85 on page 117 lists the TEMp_records keyword parameter.

Table 85: TEMp_records keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>the ENForce statement applies when an application program attempts to insert a message to its temporary queue (between commit points), and the insert would cause the number of records on either the short or long message queue data set FOR THE TEMPORARY QUEUE to exceed nn records. QPF only checks queue data sets (short, long, or both) that exceed the Protect Phase threshold.</td>
</tr>
</tbody>
</table>

THReshold% keyword

THReshold% is an optional keyword for the ENForce statement.

Table 86 on page 117 lists the THReshold% keyword parameter.

Table 86: THReshold% keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>the ENForce statement applies only when a message queue reaches the percentage value that you indicate. You can specify a value from 0 to 99. Note: Percentages below the Protect Phase threshold have no effect because ENForce processing does not begin until the Protect Phase starts.</td>
</tr>
</tbody>
</table>
**TMEMBER/MEMBER keyword**

TMEMBER is an optional keyword for the ENForce statement.

You can use the TMEMBER keyword in place of the DESTination keyword to specify a TMEMBER.

You can use MEMBER as an alias for TMEMBER.

Table 87 on page 118 lists the TMEMBER keyword parameter.

**Table 87: TMEMBER keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| `mmmmmmmmmmmmmm` or `(mmmmmmmmmmmmmm,...)` up to 16 characters per TMEMBER | a name or mask, or a list of names or masks enclosed in parentheses.  
If you specify the TMEMBER keyword, QPF assumes DESType=OTMA.  
The ENForce statement applies only when an application program or input device enqueues one or more messages to the destination(s) specified. |

**TPIPE/PIPE keyword**

TPIPE is an optional keyword for the ENForce statement.

You can use PIPE as an alias for TPIPE.

Table 88 on page 118 lists the TPIPE keyword parameter.

**Table 88: TPIPE keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| `mmmmmmmmmm` or `(mmmmmmmmmm,...)` | a name or mask, or a list of names or masks enclosed in parentheses.  
If you specify the TPIPE keyword, QPF assumes DESType=OTMA.  
The ENForce statement applies only when an application program or input device enqueues one or more messages to the destination(s) specified. |

**TPName keyword**

TPName is an optional keyword for the ENForce statement. You can use the TPName keyword in place of the DESTination keyword to specify a TPName.

Table 89 on page 119 lists the TPName keyword parameter.
Table 89: TPName keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm...mm or (mm...mm,...)</td>
<td>a name or mask, or a list of names or masks enclosed in parentheses</td>
</tr>
<tr>
<td></td>
<td>If you specify the TPName keyword, QPF assumes DESType=TPName.</td>
</tr>
<tr>
<td></td>
<td>The ENForce statement applies only when an application program or input</td>
</tr>
<tr>
<td></td>
<td>device enqueues one or more messages to the destination(s) specified.</td>
</tr>
</tbody>
</table>

**TRANsaction keyword**

TRANsaction is an optional keyword for the ENForce statement.

Table 90 on page 119 lists the TRANsaction keyword parameter.

Table 90: TRANsaction keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mmmmmmmmm or</td>
<td>a transaction name or mask, or a list of transaction names or masks enclosed</td>
</tr>
<tr>
<td>(mmmmmmmmmm,</td>
<td>in parentheses</td>
</tr>
<tr>
<td>mmmmmmmmm,...)</td>
<td>The ENForce statement applies only to transactions that match the name(s)</td>
</tr>
<tr>
<td></td>
<td>specified.</td>
</tr>
</tbody>
</table>

**TYPE keyword**

TYPE is an optional keyword for the ENForce statement.

Some of the TYPE parameters include other TYPE parameters.

LTERM includes:

- STATic
- DYNAMic
- DEADQ

CNT includes:

- MSName
- VSPcnt
- APPC
- OTMA

Table 91 on page 120 lists the TYPE keyword parameters.
Table 91: TYPE keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any/All/A BMP</td>
<td>indicates whether the ENForce statement applies to:</td>
</tr>
<tr>
<td>CNT</td>
<td>■ application programs running as BMPs</td>
</tr>
<tr>
<td>DYNAMic/VIRTual/ETO/VTF/CNT-E/CNT-V LTERMs</td>
<td>■ application programs running as MPPs</td>
</tr>
<tr>
<td>LUname/TPName/APPC QMR</td>
<td>■ input devices (identified by type LTERM)</td>
</tr>
<tr>
<td>MPP/REGion/RGN</td>
<td>■ QMR utilities</td>
</tr>
<tr>
<td>MSName</td>
<td>You can specify multiple values for the TYPE keyword.</td>
</tr>
<tr>
<td>OTMA</td>
<td>To exclude a type (except for Any), add NO to the applicable parameter.</td>
</tr>
<tr>
<td>STATic</td>
<td>For example, TYPE=NOBMP.</td>
</tr>
<tr>
<td>VSPcnt</td>
<td></td>
</tr>
</tbody>
</table>

#Msgs keyword

The ENForce statement requires one of the following keywords: #Msgs, #RECORDs, %USED, or TEMp_records.

Table 92 on page 120 lists the #Msgs keyword parameter.

Table 92: #Msgs keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>The ENForce statement applies when an application program or input device enqueues one or more messages to a destination, and the enqueue would cause the number of messages queued to that destination to equal or exceed (nn) messages.</td>
</tr>
</tbody>
</table>

Normally, you can use multiple ENFORCE statements to take different actions as the situation gets worse. With OTMA, this approach might not work as expected. Statements with the #MSGS or #RECORDS keywords do not match until a message is enqueued. (They do not match when a record is inserted.) For a non-OTMA message, when a statement matches on an enqueue, the action remains pending until the next insert, when the action is taken once. For an OTMA message, the action remains pending and is taken on all subsequent inserts from the same TPIPE. In most cases, this processing prevents the TPIPE from ever performing another enqueue, and thus prevents it from matching any other ENFORCE statement with #MSGS or #RECORDS.
In this example, when the destination reaches 1000 records, QPF starts taking the FAIL action and takes it on all future inserts, preventing any more enqueues for the TPIPE. If the destination reaches 2000 records (from other sources), QPF continues to take the FAIL action for inserts for the TPIPE. The first ENFORCE statement can match other TPIPEs, but cannot match any TPIPE that is already failing. Because the ENFORCE statement only matches on enqueues, the statement does not match. However, you can use an IPOST action to cancel the FAIL action, after which the enqueues will resume, and the IWAIT will match. Also, if the first action is WARN (which does not stop the enqueue from occurring), QPF takes the next action as expected.

**#RECORDs keyword**

The ENForce statement requires one of the following keywords: #Msgs, #RECORDs, %USED, or TEMp_records.

Table 93 on page 121 lists the #RECORDs keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>the ENForce statement applies when an application program or input device enqueues one or more messages to a destination, and the enqueue would cause the number of records queued to that destination to equal or exceed nn records. QPF checks record counts on the short and long message queue data sets independently, and it only checks for queue data sets that exceed the Protect Phase threshold.</td>
</tr>
</tbody>
</table>

Normally, you can use multiple ENFORCE statements to take different actions as the situation gets worse. With OTMA, this approach might not work as expected. Statements with the #MSGS or #RECORDS keywords do not match until a message is enqueued. (They do not match when a record is inserted.) For a non-OTMA message, when a statement matches on an enqueue, the action remains pending until the next insert, when the action is taken once. For an OTMA message, the action remains pending and is taken on all subsequent inserts from the same TPIPE. In most cases, this processing prevents the TPIPE from ever performing another enqueue, and thus prevents it from matching any other ENFORCE statement with #MSGS or #RECORDS.
Example

ENFORCE TYPE=OTMA,#REC=2000,ACTION=IWAIT
ENFORCE TYPE=OTMA,#REC=1000,ACTION=FAIL

In this example, when the destination reaches 1000 records, QPF starts taking the FAIL action and takes it on all future inserts, preventing any more enqueues for the TPIPE. If the destination reaches 2000 records (from other sources), QPF continues to take the FAIL action for inserts for the TPIPE. The first ENFORCE statement can match other TPIPEs, but cannot match any TPIPE that is already failing. Because the ENFORCE statement only matches on enqueues, the statement does not match. However, you can use an IPOST action to cancel the FAIL action, after which the enqueues will resume, and the IWAIT will match. Also, if the first action is WARN (which does not stop the enqueue from occurring), QPF takes the next action as expected.

%USED keyword

The ENForce statement requires one of the following keywords: #Msgs, #RECords, %USED, or TEMp_records.

Table 94 on page 122 lists the %USED keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>the ENForce statement applies when an application program attempts to insert a message to its temporary queue (between commit points), and the insert would cause the number of records on either the short or long message queue data set FOR THE TEMPORARY QUEUE to equal or exceed the specified percentage for the entire queue data set. The ENForce statement also applies when an application program or input device enqueues one or more messages to a destination, and the enqueue would cause the number of records on either the short or long message queue data set FOR THAT DESTINATION to equal or exceed the specified percentage for the entire queue data set. QPF only checks queue data sets (short, long, or both) that exceed the Protect Phase threshold.</td>
</tr>
</tbody>
</table>

PROCess statement keywords

The PROCess statement defines the actions that QPF implements during the Protect Phase to reduce the number of messages on the message queues. The statement applies to destinations with queued messages.

The order of the PROCess statements is meaningful. When the message queues reach the Protect Phase threshold, QPF analyzes the destinations individually. The analysis begins with the destination with the highest queue usage. For each destination, QPF
scans the PROCess statements until it finds a match, and then takes the action indicated on the statement. Therefore, you should place the most restrictive statement with the highest threshold first.

You must specify one or two destination-related keywords. You can choose from the following keyword pairs:

- DESTination and DESTYpe
- TPName and LUname
- TMEMBER and TPIPE

Also, the PROCess statement requires at least one of the #Msgs, #RECords, or %USED keywords.

To dequeue a master or secondary master terminal, specify the DESTination keyword as the explicit name of the terminal, and then specify \texttt{FORCE=YES}. Do not specify \texttt{ACTion=STOP}. \texttt{STOP} is not supported for master terminals.

### ACTion keyword

ACTion is an optional keyword for the PROCess statement. The ACTion keyword determines what action QPF takes for a destination to reduce the number of messages on the message queues.

**Note**

You can specify multiple actions, but you cannot specify both \texttt{DEQueue} and \texttt{UNLoad_DEQueue}.

**Table 95 on page 123** lists the ACTion keyword parameters.

**Table 95: ACTion keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WARN</td>
<td>QPF sends a message to the recipients specified for the NOTIFY keyword. The message includes the destination name, the relative number of the matching PROCess statement, and the status of the actions. All other actions (except NONe) imply WARN.</td>
</tr>
<tr>
<td>DEQueue</td>
<td>QPF deletes queued messages; QPF stops the destination and node, performs the DEQueue, and restarts (unless STOP is also specified)</td>
</tr>
<tr>
<td>NONe</td>
<td>QPF does not take action during the Protect Phase. <strong>Note:</strong> You can specify a subcommand with ACTion=NONE to &quot;protect&quot; critical destinations, although BMC Software recommends ACTion=WARN so that notification occurs.</td>
</tr>
</tbody>
</table>
### Parameter Description

**STOP**
- if the destination is not already stopped, QPF issues an IMS /STOP command to stop the node or LTERM
- You can include STOP in addition to DEQueue or UNLoad_DEQueue.

**UNLoad_DEQueue**
- QPF unloads and deletes messages
- If you specify the UNLoad_DEQueue action, you must also include the UNLoad_dsn subcommand.
- Note: You can shorten UNLoad_DEQueue to UNLoad (UNLoad_DEQueue is usually referred to as UNLoad). QPF does not contain an "unload only" action, since that would not resolve any message queue problems.

---

#### DESTination keyword

DESTination is an optional keyword for the PROCess statement.

Table 96 on page 124 lists the DESTination keyword parameters.

**Table 96: DESTination keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| mm...mm or (mm...mm,... ) up to 64 characters for TPNames | a destination name or mask, or a list of destination names or masks enclosed in parentheses  
For APPC destinations, the name is the TPName. For OTMA destinations, the name is the TMember.  
The PROCess statement applies only to the destination(s) specified. |

#### DESType/DTYPe keyword

DESType is an optional keyword for the PROCess statement.

You can use DTYPe as an alias for DESType.

Some of the DESType parameters include other DESType parameters:

| LTERM includes: | CNT includes: | STATic  
DYNAMic  
DEADQ |
|-----------------|---------------|-----------------|
| MSName  
VSPcnt  
APPC  
OTMA |

---

Message Advisor for IMS Reference Manual

124
**TRANSACTION** includes:

**SMB** includes:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All/A</td>
<td>a destination type, or a list of destination types</td>
</tr>
</tbody>
</table>
| BMP       | The PROCess statement applies only to the destination type(s) specified.  
You can specify a list of values. For example, DESTYpe=(APPC,OTMA) matches for either APPC or OTMA destinations.  
To exclude a type (except for All), add NO in front of the applicable parameter. For example, DESTYpe=NOBMP. |
| CNT       | |
| DEADQ     | |
| DYNAMic/VIRTual/ETO/VTF/CNT-E/CNT-V | |
| LTERMsa  | |
| MPP       | |
| MSName    | |
| OTMA      | |
| RSMB      | |
| SMB       | |
| STATic   | |
| SYSTEM    | |
| TPName/APPc | |
| TRANSa|ction | |
| VSPcnt    | |

**FORCE** keyword

FORCE is an optional keyword for the PROCess statement.

**Table 98 on page 125** lists the FORCE keyword parameters.

**Table 98: FORCE keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| No        | QPF stops and quiesces the destination before performing a DEQueue or UNLoad action  
If QPF cannot stop or quiesce the destination, it does not perform the action and issues a DEST NOT STOPPED error message. |
Parameter | Description
--- | ---
AT_SHUTdown | QPF forces the action only if IMS reaches the point where system tasks can placed in an IWAIT. Use the Overflow Phase SYSTEM_WAIT keyword to specify the shutdown point. The SYSTEM_WAIT default is to wait the action upon reaching the IMS-defined shutdown limit; however, you can specify an earlier shutdown point. When IMS reaches the shutdown point, you can no longer issue IMS commands, so you cannot stop active destinations. Unless you specify FORCE=AT_SHUTdown, QPF only performs DEQueue and UNLoad actions on destinations that are already stopped.

| Yes | QPF performs DEQueue and UNLoad actions without stopping the destination |

**WARNING**

Specifying `FORCE=Yes` for an active destination may result in the following situations:

- the action may fail
- you may receive IMS message DFS1959E
- in rare cases, IMS may abend with a U0757

**LAST_Used keyword**

LAST_Used is an optional keyword for the PROCess statement.

Table 99 on page 126 lists the LAST_Used keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnn</td>
<td>the PROCess statement applies only to destinations for which the last enqueue occurred at least nnn hours ago. <strong>Note</strong>: The LAST_Used keyword requires more overhead than other keywords. If you specify the LAST_Used keyword, BMC Software recommends that you combine it with other keywords that restrict the matches as much as possible (DESTination, #RECords).</td>
</tr>
</tbody>
</table>

**LUname keyword**

LUname is an optional keyword for the PROCess statement. Consider the following when specifying the LUname keyword:

- You can specify a list of LU names or patterns for the LUname keyword value.
- You can use LUname and TPName together to identify a specific APPC destination.

- Message Advisor assumes DESType=TPName when you specify the LUname keyword.

- An LU name can include a network ID (netid.luname).

Table 100 on page 127 lists the LUname keyword parameter.

### Table 100: LUname keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmmmm. mmmmmmmmm,...)</td>
<td>a name or mask, or a list of names or masks enclosed in parentheses. The PROCess statement applies only to APPC destinations that match the LUname(s) specified. If you specify the LUname keyword, QPF assumes DESType=TPName.</td>
</tr>
</tbody>
</table>

**NOTIFY keyword**

NOTIFY is an optional keyword for the PROCess statement.

The default is the value that was specified for the NOTIFY keyword on the PROTECT_PHASE subcommand.

Table 101 on page 127 lists the NOTIFY keyword parameters.

### Table 101: NOTIFY keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTO</td>
<td>QPF sends messages to the IMS master terminal when the PROCess statement takes effect</td>
</tr>
<tr>
<td>NONE</td>
<td>QPF does not issue messages</td>
</tr>
<tr>
<td>WTO</td>
<td>QPF sends messages to the MVS console when the PROCess statement takes effect</td>
</tr>
<tr>
<td>tso-userids</td>
<td>QPF sends messages to the specified TSO users when the PROCess statement takes effect. <strong>Note:</strong> If you specify a user more than once, the user receives more than one copy of each message. TSO users only receive messages if they are logged on and terminal output buffers are available.</td>
</tr>
</tbody>
</table>
Queue keyword

Queue is an optional keyword for the PROCess statement. If you specify the Queue keyword, the PROCess statement applies only to destinations with messages on the queues that you indicate.

You can list more than one parameter for the Queue keyword value. To exclude a parameter, use ¬ (for example, Queue=¬2).

The #_msgs, #records, and %used keywords apply to all queues, not just the selected queues. The Queue keyword restricts any deQueue or UnLoad_deQueue actions resulting from the PROCess statement to the queues indicated for the keyword. The Queue keyword allows you to deQueue all messages on the selected queues first. Then, if the destination remains on the problem list, the PROCess statement will not match. You can use another PROCess statement to deQueue the remaining queues.

Table 102 on page 128 lists the Queue keyword parameters.

Table 102: Queue keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>QPF deQueue messages on all queues</td>
</tr>
<tr>
<td>1</td>
<td>QPF deQueue messages on queue 1, which contains replies to response or conversational-type messages</td>
</tr>
<tr>
<td>2</td>
<td>QPF deQueue messages on queue 2, which contains replies to transactions from a terminal in exclusive mode; used with the /set command</td>
</tr>
<tr>
<td>3</td>
<td>QPF deQueue messages on queue 3, which contains IMS system messages</td>
</tr>
<tr>
<td>4</td>
<td>QPF deQueue messages on queue 4, which contains message switches, alternate PCB output, and other traffic</td>
</tr>
<tr>
<td>S</td>
<td>QPF deQueue messages on the transaction suspend queue</td>
</tr>
</tbody>
</table>

STATus keyword

STATus is an optional keyword for the PROCess statement. If you specify the STATus keyword, the PROCess statement applies only to destinations with the status that you indicate.

You can specify more than one parameter for the STATus keyword value.

Table 103 on page 129 lists the STATus keyword parameters.
Table 103: STATus keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>the PROCess statement matches any LTERM</td>
</tr>
<tr>
<td>IOERROR</td>
<td>the PROCess statement matches only destinations that show a queue I/O error</td>
</tr>
<tr>
<td>LOCKED</td>
<td>the PROCess statement matches only locked destinations</td>
</tr>
<tr>
<td>NOTALLOC</td>
<td>the PROCess statement matches only LTERMs that are not currently allocated</td>
</tr>
<tr>
<td>STOPPED</td>
<td>the PROCess statement matches only stopped LTERMs</td>
</tr>
</tbody>
</table>

**THReshold% keyword**

THReshold% is an optional keyword for the PROCess statement.

The following table lists the THReshold% keyword parameter.

Table 104: THReshold% keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(nn)</td>
<td>the PROCess statement applies only when a message queue reaches the percentage value that you indicate</td>
</tr>
</tbody>
</table>

**Note:** If the PROCess statement THReshold% is less than the PROTECT_Phase subcommand THReshold%, QPF ignores the PROCess THReshold%. If the PROCess statement THReshold% is greater than the PROTECT_Phase subcommand THReshold%, QPF will not initiate the PROCess statement actions until the message queue reaches the PROCess THReshold%.

**TMEMBER/MEMBER keyword**

TMEMBER is an optional keyword for the PROCess statement.

You can use MEMBER as an alias for TMEMBER.

Table 105 on page 130 lists the TMEMBER keyword parameter.
Table 105: TMEMBER keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mmmmmmmmmmmmmmmmmmmmmmmmmmm or (mmmmmmmmmmmmmmmmmmmmmmmmmm,...) up to 16 characters per TMEMBER</td>
<td>a name or mask, or a list of names or masks enclosed in parentheses. If you specify the TMEMBER keyword, QPF assumes DESTYpe=OTMA. Note: mmmmmmmmmmmmmmmmm is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

**TPIPE/PIPE keyword**

TPIPE is an optional keyword for the PROCess statement.

You can use PIPE as an alias for TPIPE.

Table 106 on page 130 lists the TPIPE keyword parameter.

Table 106: TPIPE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mmmmmmmmmmmm or (mmmmmmmmmmmm,...)</td>
<td>a name or mask, or a list of names or masks enclosed in parentheses. The PROCess statement applies only to OTMA destinations that match the TPIPE(s) specified. If you specify the TPIPE keyword, QPF assumes DESTYpe=OTMA. Note: mmmmmmmmmmm is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

**TPName keyword**

TPName is an optional keyword for the PROCess statement. You can use the TPName keyword in place of the DESTination keyword to specify a TPName.

Table 107 on page 130 lists the TPName keyword parameter.

Table 107: TPName keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm...mm or (mm...mm,...)</td>
<td>a name or mask, or a list of names or masks enclosed in parentheses. If you specify the TPName keyword, QPF assumes DESTYpe=TPName. Note: mm...mm is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>
**#Msgs keyword**

The PROCess statement requires one of the following keywords: #Msgs, #RECords, or %USED. If you specify more than one of the keywords, the PROCess statement applies even if only one of the keywords matches. That is, there is a logical OR between the three keywords. All other keywords must match for the statement to apply. That is, there is a logical AND between all other selection and utilization keywords.

Table 108 on page 131 lists the #Msgs keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| nn        | the PROCess statement applies only to destinations for which the number of messages queued equals or exceeds the number indicated (regardless of the data set)  
            The lowest value you can specify for the #Msgs keyword is 10. |

**#RECords keyword**

The PROCess statement requires one of the following keywords: #Msgs, #RECords, or %USED. If you specify more than one of the keywords, the PROCess statement applies even if only one of the keywords matches. That is, there is a logical OR between the three keywords. All other keywords must match for the statement to apply. That is, there is a logical AND between all other selection and utilization keywords.

Table 109 on page 131 lists the #RECords keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| nn        | the PROCess statement applies only to destinations for which the number of records in use on the message queue (short, long, or both) equals or exceeds the number indicated, and the queue is above the Protect Phase threshold  
            The lowest value you can specify for the #RECords keyword is 10.  
            **Note:** The record count is not available when monitoring first starts. The count only becomes available as time permits. To allow a match on either the message count or the record count, you can include a #Msgs keyword. |
%USED keyword

The PROCess statement requires one of the following keywords: #Msgs, #RECords, or %USED. If you specify more than one of the keywords, the PROCess statement applies even if only one of the keywords matches. That is, there is a logical OR between the three keywords. All other keywords must match for the statement to apply. That is, there is a logical AND between all other selection and utilization keywords.

Table 110 on page 132 lists the %USED keyword parameter.

Table 110: %USED keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( nn )</td>
<td>the PROCess statement applies only to destinations that are using the specified percentage of the message queue (short, long, or both) when the queue is above the Protect Phase threshold You must specify a value greater than zero. Note: The percentage used is not available when monitoring first starts. The percentage only becomes available as time permits. To allow a match on either the message count or the percentage used, you can include a #Msgs keyword.</td>
</tr>
</tbody>
</table>

OVERFLOW_PHASE keywords

Before you can set or replace the QPF options, the phase subcommands listed below must be present in the QPF_OPTIONS command set:

- MONITOR_PHASE
- PROTECT_PHASE
- OVERFLOW_PHASE

Note

IMS must be active before you can specify the QPF_OPTIONS command.

The keywords in this section apply to the Overflow Phase.

ACTion keyword

ACTion is an optional keyword for the OVERFLOW_PHASE subcommand. The ACTion keyword determines what action QPF takes when the Overflow Phase starts.
You can specify a list of actions for the ACTion keyword value.

Table 111 on page 133 lists the ACTion keyword parameters.

### Table 111: ACTion keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWAIT/WAIT</td>
<td>QPF waits any ITASK (except for system components) that tries to insert a message into the message queue. The task remains in a wait until the Overflow Phase terminates. See the SYSTEM_WAIT keyword for information about pausing system tasks. <strong>Note:</strong> You can use ACTion=WAIT as an alias for ACTion=IWAIT.</td>
</tr>
<tr>
<td>DUMPq</td>
<td>QPF starts an IMS DUMPQ shutdown</td>
</tr>
<tr>
<td>NONe</td>
<td>QPF does not take action during the Overflow Phase</td>
</tr>
</tbody>
</table>
| SNAPQ           | QPF starts an IMS SNAPQ. Consider the following issues if you specify the SNAPQ action:  
- QPF suspends ENForce and PROTECT_PHASE processing until the SNAPQ completes.  
- With extremely high message queue utilization and extremely large message queue data sets, the SNAPQ may require a considerable amount of time to complete.  
- After the SNAPQ completes, QPF could reenter the Overflow Phase and initiate another SNAPQ if the RESET% value is close to the THReshold% value, and/or the RESET_TIME value is small. |
| STOP            | QPF stops all message processing regions (BMPs and MPPs) |
| OTMA_STOP       | QPF stops input from all OTMA members when the overflow phase starts. QPF issues a /STOP TMEMBER ALL command to stop the input. When the overflow phase ends, QPF restarts the OTMA members. If all members were started when QPF issued the stop command, QPF issues a /STA TMEMBER ALL command to restart all members. If only some members were started when QPF issued the stop command, QPF issues a /STA TMEMBER command for those members only. The OTMA_STOP keyword is used by default (along with IWAIT) if you do not specify the ACTION keyword. |

### AUTO_cmd keyword

AUTO_cmd is an optional keyword for the OVERFLOW_Phase subcommand. The AUTO_cmd keyword allows further flexibility in performing automated actions. You can use this keyword to specify an MVS operator command that QPF will issue each time the phase starts.
Table 112 on page 134 lists the AUTO_cmd keyword parameter.

Table 112: AUTO_cmd keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'cmd'</td>
<td>QPF issues the specified MVS operator command each time the Overflow Phase starts</td>
</tr>
</tbody>
</table>

**NOTIFY keyword**

NOTIFY is an optional keyword for the OVERFLOW_PHASE subcommand and indicates the recipients of informational messages that QPF issues when the Overflow Phase starts and stops. The message includes the phase name and information regarding message queue data set usage.

The default is the value that was specified for the NOTIFY keyword on the PROTECT_PHASE subcommand.

Table 113 on page 134 lists the NOTIFY keyword parameters.

Table 113: NOTIFY keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTO</td>
<td>QPF sends messages to the IMS master terminal when the Overflow Phase starts and stops</td>
</tr>
<tr>
<td>NONE</td>
<td>QPF does not issue messages</td>
</tr>
<tr>
<td>WTO</td>
<td>QPF sends messages to the MVS console when the Overflow Phase starts and stops</td>
</tr>
<tr>
<td>tso-userids</td>
<td>QPF sends messages to the specified TSO users when the Overflow Phase starts and stops</td>
</tr>
</tbody>
</table>

*Note:* If you specify a user more than once, the user receives more than one copy of each message. TSO users only receive messages if they are logged on and terminal output buffers are available.

**NOTIFY% keyword**

NOTIFY% is an optional keyword for the OVERFLOW_PHASE subcommand.

Table 114 on page 135 lists the NOTIFY% keyword parameters.
Table 114: NOTIFY% keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>QPF reissues the NOTIFY message when the maximum of the percent full values for the message queue data sets changes by the value specified for the Protect Phase NOTIFY% keyword</td>
</tr>
<tr>
<td>0</td>
<td>QPF issues the NOTIFY message only when the Overflow Phase starts and stops, or as specified for the NOTIFY_INTerval keyword</td>
</tr>
<tr>
<td>nn</td>
<td>QPF reissues the NOTIFY message when the maximum of the percent full values for the message queue data sets changes by nn percentage points</td>
</tr>
</tbody>
</table>

**NOTIFY_INTerval keyword**

NOTIFY_INTerval is an optional keyword for the OVERFLOW_PHASE subcommand.

The following table lists the NOTIFY_INTerval keyword parameters.

Table 115: NOTIFY_INTerval keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>QPF issues messages according to the value specified for the Protect Phase NOTIFY_INTerval keyword</td>
</tr>
<tr>
<td>0</td>
<td>QPF issues the NOTIFY message only when the Overflow Phase starts and stops, or as specified for the NOTIFY% keyword</td>
</tr>
<tr>
<td>nn</td>
<td>QPF allows nn minutes between NOTIFY messages; indicates that QPF should issue a NOTIFY message at least this often</td>
</tr>
</tbody>
</table>

**RESET% keyword**

RESET% is an optional keyword for the OVERFLOW_PHASE subcommand. The value is a percentage that indicates when the Overflow Phase should end. The phase ends when both message queue data sets (SHMSG and LGMSG) remain below the RESET% value for the length of time specified for the RESET_TIME keyword. The reset value must be less than or equal to the value specified for the THReshold% keyword.

Table 116 on page 136 lists the RESET% keyword parameters.
Table 116: RESET% keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>the Overflow Phase ends when the message queue data sets remain below the value specified for the THReshold% keyword minus one. The data sets must remain below this value for the length of time specified for the RESET_TIME keyword.</td>
</tr>
<tr>
<td>nn</td>
<td>the Overflow Phase ends when the message queue data sets remain below this percentage value for the length of time specified for the RESET_TIME keyword. The percentage value must be less than or equal to the THReshold% keyword value.</td>
</tr>
<tr>
<td>0</td>
<td>the Overflow Phase remains active until you manually terminate it using the QPF_Options command</td>
</tr>
</tbody>
</table>

RESET_TIME keyword

RESET_TIME is an optional keyword for the OVERFLOW_PHASE subcommand. The value is a number of minutes that is used to determine when the Overflow Phase should end. The phase ends when both message queue data sets (SHMSG and LGMSG) remain below the RESET% value for the number of minutes specified.

Note

It is possible for a phase to reach the reset threshold, begin to shut down, and then reach the start-up threshold before shut down completes. In this case, stopped messages are not received, but all normal start-up processing occurs (including reissuing the AUTO_cmd).

Table 117 on page 136 lists the RESET_TIME keyword parameter.

Table 117: RESET_TIME keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 nn</td>
<td>the Overflow Phase ends when the message queue data sets remain below the RESET% value for the number of minutes specified. Note: When THReshold% equals RESET%, Message Advisor forces the RESET_TIME to at least one minute.</td>
</tr>
</tbody>
</table>

SYSTEM_WAIT keyword

SYSTEM_WAIT is an optional keyword for the OVERFLOW_PHASE subcommand. The keyword indicates whether QPF places IMS system tasks (such as the WTOR processor) in a wait if they try to insert a message when ACTion=IWAIT.
Note
If you specify the SYSTEM_WAIT keyword, you must also specify ACTion=IWAIT.

Table 118 on page 137 lists the SYSTEM_WAIT keyword parameters.

Table 118: SYSTEM_WAIT keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT_SHUTdown</td>
<td>QPF only waits system tasks when IMS reaches the IMS-defined shutdown limit</td>
</tr>
<tr>
<td></td>
<td>Note: If you specify AT_SHUTdown, you must also specify WTOR=Yes.</td>
</tr>
<tr>
<td>AT_OVERflow</td>
<td>QPF always waits system tasks during the Overflow Phase</td>
</tr>
<tr>
<td></td>
<td>Note: If you specify AT_OVERflow, you must also specify WTOR=Yes.</td>
</tr>
<tr>
<td>NEVER</td>
<td>QPF never waits system tasks</td>
</tr>
<tr>
<td></td>
<td>If system tasks cause a message queue to reach the IMS-defined shutdown limit,</td>
</tr>
<tr>
<td></td>
<td>IMS shuts down.</td>
</tr>
</tbody>
</table>

THReshold% keyword

THReshold% is an optional keyword for the OVERFLOW_PHASE subcommand. The value is a percentage that indicates when the Overflow Phase should begin. The phase begins when either message queue data set (SHMSG or LGMSG) reaches the percentage value.

Note
If a phase is not already active, it begins automatically when IMS SHUTDWN value.

Table 119 on page 137 lists the THReshold% keyword parameters.

Table 119: THReshold% keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>the Overflow Phase starts when either message queue data set reaches 99 percent</td>
</tr>
<tr>
<td>nn up to 99</td>
<td>the Overflow Phase starts when either message queue data set reaches this percentage value</td>
</tr>
<tr>
<td>0</td>
<td>the Overflow Phase is continuously active</td>
</tr>
</tbody>
</table>

WTO_Desc keyword

WTO_Desc is an optional keyword for the OVERFLOW_PHASE subcommand.
Table 120 on page 138 lists the WTO_Desc keyword parameter.

Table 120: WTO_Desc keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X’cccc’</td>
<td>the descriptor code to be used when you specify <code>NOTIFY=WTO</code> or <code>WTOR=YES</code>&lt;br&gt;The default is the value that is used by the IMS control region.</td>
</tr>
</tbody>
</table>

**WTO_Routcde keyword**

WTO_Routcde is an optional keyword for the OVERFLOW_PHASE subcommand.

Table 121 on page 138 lists the WTO_Routcde keyword parameter.

Table 121: WTO_Routcde keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X’cccc’</td>
<td>the routing code to be used when you specify <code>NOTIFY=WTO</code> or <code>WTOR=YES</code>&lt;br&gt;The default is the value that is used by the IMS control region.</td>
</tr>
</tbody>
</table>

**WTOR keyword**

WTOR is an optional keyword for the OVERFLOW_PHASE subcommand and indicates whether QPF issues a WTOR when the Overflow Phase starts. If QPF issues a WTOR, you can manually initiate QPF actions.

Table 122 on page 138 lists the WTOR keyword parameters.

Table 122: WTOR keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>if you specified a value for the Protect Phase WTOR keyword, that value remains in effect by default</td>
</tr>
<tr>
<td>No</td>
<td>QPF does not issue a WTOR when the Overflow Phase starts</td>
</tr>
<tr>
<td>Yes</td>
<td>QPF issues a WTOR when the Overflow Phase starts</td>
</tr>
</tbody>
</table>

**UNLoad_dsn keywords**

The UNLoad_dsn subcommand indicates the unload data set(s) to be used for any UNLoad_DEQueue actions. If you include any UNLoad_DEQueue actions on
PROCess statements, or if you want to issue the UNLoad action manually (using the ISPF interface, the QPF_ACTion command, or the operator WTOR), you must also specify the UNLoad_dsn subcommand.

QPF does not allocate unload data sets until an UNLoad_DEQueue action starts. However, once QPF allocates the data sets, they normally remain allocated and available for future UNLoad actions, with the following exception: when each UNLoad action completes, QPF automatically frees any data set name that includes a %DEST symbolic keyword. Otherwise, QPF frees the data set when one of the following occurs:

- the data set is full or receives an I/O error
- the Monitor Phase ends
- a user resets the Monitor Phase using the QPF_OPTIONS RESET keyword

**Note**

Because QPF does not support an UNLoad without a DEQueue, the shortened form of UNLoad is interchangeable with UNLoad_DEQueue. Both commands indicate that QPF unloads and dequeues messages.

### BLKSIZe keyword

BLKSIZe is an optional keyword for the UNLoad_dsn subcommand.

Table 123 on page 139 lists the BLKSIZe keyword parameter.

**Table 123: BLKSIZe keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23476</td>
<td>the blocksize for new unload data sets</td>
</tr>
<tr>
<td>nnnnn</td>
<td>The value also applies to existing data sets for which the blocksize is not defined. You can specify values between 84 and 32767. However, if you specify a value that is less than the LGMSG data set record length plus 16, QPF changes the blocksize to the LGMSG record length plus 16.</td>
</tr>
</tbody>
</table>

### CYLS_PRIM keyword

CYLS_PRIM is a required keyword for the UNLoad_dsn subcommand.

Table 124 on page 140 lists the CYLS_PRIM keyword parameter.
Table 124: CYLS_PRIM keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnn</td>
<td>the number of primary cylinders to allocate for new data sets</td>
<td></td>
</tr>
</tbody>
</table>

**CYLS_SEC keyword**

CYLS_SEC is an optional keyword for the UNLoad_dsn subcommand.

Table 125 on page 140 lists the CYLS_SEC keyword parameter.

Table 125: CYLS_SEC keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnn</td>
<td>the number of secondary cylinders to allocate for new data sets</td>
<td></td>
</tr>
</tbody>
</table>

**DATACLAS keyword**

DATACLAS is an optional keyword for the UNLoad_dsn subcommand.

Table 126 on page 140 lists the DATACLAS keyword parameter.

Table 126: DATACLAS keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccccc</td>
<td>the SMS data class for new data sets</td>
<td></td>
</tr>
</tbody>
</table>

**DISP keyword**

DISP is an optional keyword for the UNLoad_dsn subcommand and indicates the disposition of the data set.

Table 127 on page 140 lists the DISP keyword parameters.

Table 127: DISP keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MOD</td>
<td>if the data set exists, concatenates new unloaded messages to the existing data set; if the data set does not exist, allocates a new data set</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td></td>
</tr>
</tbody>
</table>
| NEW       | allocates a new data set  
The allocation fails if the data set already exists.  

**Note:** BMC Software recommends DISP=MOD, since QPF attempts to reallocate the data set in the following situations:  
- after an IMS restart  
- the Monitor Phase ends and then restarts |

### DSName keyword

DSName is a required keyword for the UNLoad_dsn subcommand. Consider the following when specifying the DSName keyword:

- You can include symbolic keywords when specifying a value for the DSName keyword. A symbolic keyword begins with a percent sign (%) and optionally ends with a percent sign (%). QPF substitutes the appropriate value when it allocates the data set.

  **Note**  
  To perform retries, you must specify a variable symbolic keyword or a GDG designation.

  A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the remaining characters are concatenated to the symbolic value.

- You can include system symbols when specifying a value for the DSName keyword. A system symbol begins with an ampersand (&) and optionally ends with a period (.). The system will substitute a value for valid symbols. You will receive an allocation error if you use an undefined system symbol or if the substitution results in an invalid data set name. See the JCL Reference Manual for more information about system symbols.

See “Command syntax considerations” on page 37 for additional items to consider when specifying the DSName keyword value.  
**Table 128 on page 142** lists the DSName keyword parameters.
### Table 128: DSName keyword parameters

<table>
<thead>
<tr>
<th>Symbolic Keyword</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| N/A              | dsn       | the data set name  
You can specify a GDG with (+1) to indicate that QPF is to create a new generation in the GDG. Also, you can include symbolic keywords in the data set name. QPF substitutes the appropriate value when it allocates the data set. |
| %DATE            | yyddd     | the day of the year |
| %DATE4           | yyyyddd   | the day of the year |
| %DEST            | 1 to 8 characters | the destination or origin being unloaded  
The following naming conventions apply:  
— for APPC, QPF uses the LUname  
— for OTMA, QPF uses the TPIPE  
— QPF changes any non-alphanumeric-national characters to #  
— QPF changes the first character to # if it is numeric |
| %DSNNO           | nnnnnnn   | a six-digit number that is incremented for each data set allocated |
| %IDATE           | yyddd     | the start date of the active IMS system |
| %IMSID           | cccc      | the IMSID of the active system |
| %ITIME           | hhmmssst  | the start time of the active IMS system |
| %TIME            | hhmmss    | the current time of day |
| %TIMET           | hhmmss    | the current time of day |

### MAX_DSNS keyword

MAX_DSNS is an optional keyword for the UNLoad_dsn subcommand and limits the total number of data sets that QPF can allocate while the IMS system is active.

Table 129 on page 143 lists the MAX_DSNS keyword parameter.
Table 129: MAX_DSNS keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| 999999 nnnnnn | when the number of allocated data sets reaches this number, all further allocation attempts will fail until one of the following occurs:  
- new options are loaded with a higher limit  
- IMS is restarted  
You can specify any value from 1 to 999999. |

**MGMTCLAS keyword**

MGMTCLAS is an optional keyword for the UNLoad_dsn subcommand.

Table 130 on page 143 lists the MGMTCLAS keyword parameter.

Table 130: MGMTCLAS keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccccc</td>
<td>the SMS management class for new data sets</td>
</tr>
</tbody>
</table>

**MODEL_Dsname keyword**

MODEL_Dsname is an optional keyword for the UNLoad_dsn subcommand.

Table 131 on page 143 lists the MODEL_Dsname keyword parameter.

Table 131: MODEL_Dsname keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dsn</td>
<td>the data set containing model DSCB information for GDGs</td>
</tr>
</tbody>
</table>

**RETRY keyword**

RETRY is an optional keyword for the UNLoad_dsn subcommand.

Table 132 on page 144 lists the RETRY keyword parameter.
Table 132: RETRY keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>the maximum number of allocation retries QPF attempts before the allocation is considered a failure. The RETRY keyword is only allowed when you specify a variable or variable destination data set name. You can specify any value between 0 and 99. <strong>Note:</strong> If an error occurs after you specify the UNLoad_dsn keyword, you must reset the error condition status to continue using this data set. To reset the error condition status, change the UNLoad_dsn RETRY keyword value, and then submit the command set with <code>TYPE=SET</code>.</td>
</tr>
</tbody>
</table>

**STORCLAS keyword**

STORCLAS is an optional keyword for the UNLoad_dsn subcommand.

Table 133 on page 144 lists the STORCLAS keyword parameter.

Table 133: STORCLAS keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccccc</td>
<td>the SMS storage class for new data sets</td>
</tr>
</tbody>
</table>

**TYPE keyword**

TYPE is an optional keyword for the UNLoad_dsn subcommand.

Table 134 on page 144 lists the TYPE keyword parameters.

Table 134: TYPE keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNCH_BLK</td>
<td>QPF updates the unload data set only after it dequeues each block of messages</td>
</tr>
<tr>
<td>SYNCH_MSG</td>
<td>QPF updates the unload data set after it dequeues each message</td>
</tr>
</tbody>
</table>

**UNIT keyword**

UNIT is an optional keyword for the UNLoad_dsn subcommand.

Table 135 on page 145 lists the UNIT keyword parameter.
Table 135: UNIT keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccc</td>
<td>the generic unit name that QPF uses when it allocates a new data set. If you specify DISP=NEW or MOD, you must also specify the UNIT keyword. QPF ignores the UNIT keyword when allocating an existing data set. All existing data sets that QPF uses must be cataloged.</td>
</tr>
</tbody>
</table>

**VOLser keyword**

VOLser is an optional keyword for the UNLoad_dsn subcommand.

Table 136 on page 145 lists the VOLser keyword parameter.

Table 136: VOLser keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccc</td>
<td>the volume serial number that QPF uses when it allocates a new data set. All existing data sets that QPF uses must be cataloged.</td>
</tr>
</tbody>
</table>
CUSTOMIZE command

This chapter describes the Message Advisor for IMS CUSTOMIZE command set, and also provides a syntax diagram and sample command sets. For more information about building and executing a CUSTOMIZE command set, see the installation guide.

This chapter also includes a discussion of the CUSTOMIZE command’s virtual LTERM creation feature and its uppercase-only character display feature.

Overview

The CUSTOMIZE command lets you set and change Message Advisor options so that the product meets your site’s needs.

The CUSTOMIZE command set consists of the following subcommands and their associated keywords:

- IMS_OPTIONS subcommand sets Message Advisor customization options for each IMS

  **Note**
  Repeat the IMS_OPTIONS subcommand and its keywords for each IMS system that is running Message Advisor.

- LIST_OPTIONS subcommand displays the Message Advisor customization options for each of your IMS systems and Message Advisor Servers

  **Note**
  You can use the LIST_OPTIONS subcommand and its keywords to display a summary of the IMS systems that have been defined to the Message Advisor Server or to display details of Message Advisor Server or IMS options.

- SERVER_OPTIONS subcommand sets the customization options for each of your Message Advisor Servers
Note
Repeat the SERVER_OPTIONS subcommand and its keywords for each Message Advisor Server.

For more information about the CUSTOMIZE command set, see the installation guide.

Virtual LTERM creation feature

If you use Message Advisor with any of the following products, you can activate the CUSTOMIZE command’s virtual LTERM creation feature to dynamically create logical terminals (LTERMs) that are requested by requeue tasks:

- DELTA IMS VIRTUAL TERMINAL (BMC Software product)
- EXTENDED TERMINAL ASSIST PLUS (BMC Software product)
- EXTENDED TERMINAL OPTION (IBM product)

The virtual LTERM creation feature increases requeue accuracy and creates virtual LTERMs only for messages which are to be requeued.

Note
If you disable the virtual LTERM creation feature, you can requeue messages to existing LTERMs. You can requeue messages to LTERMs that do not exist only if the LTERMs appear in the VPRINTER, unsolicited output, or remote TSS table.

The VIRTUAL_CREATE keyword for the IMS_OPTIONS subcommand activates the virtual LTERM creation feature.

Uppercase-only feature

You can display Message Advisor reports in uppercase and lowercase characters. The uppercase-only feature lets you display and print Message Advisor reports in all uppercase characters. To activate the feature, specify UPPERCASE=YES for the SERVER_OPTIONS subcommand.

Note
If you use advanced program-to-program communications (APPC) TPNAMEs containing lowercase characters, do not specify UPPERCASE=YES. If you specify UPPERCASE=YES in this situation, you cannot use ISPF options 2 and 8 to process the TPNAMEs, and the TPNAMEs are displayed in uppercase characters on reports.
For more information about customization, see the installation guide.

**CUSTOMIZE command set syntax**

The following figures show the CUSTOMIZE command set syntax.

*Figure 16: CUSTOMIZE command set syntax (part 1 of 3)*
Figure 17: CUSTOMIZE command set syntax (part 2 of 3)
Sample CUSTOMIZE command sets

The following figure shows a command set for listing the Message Advisor Servers and IMS systems for which options have been defined.

Figure 19: Command set: List Message Advisor servers and IMS systems for which options have been defined

CUSTOMIZE
  LIST_OPTIONS
END
Figure 20 on page 152 shows a command set for listing detailed options for a specific IMS ID.

**Figure 20: Command set: List options for a specific IMS ID**

```
CUSTOMIZE
  LIST_OPTIONS IMSID=IMSA
END
```

Figure 21 on page 152 shows a command set for listing detailed options for a specific Message Advisor Server.

**Figure 21: Command set: List options for a specific Message Advisor server**

```
CUSTOMIZE
  LIST_OPTIONS SERVER_NAME=QMSJ
END
```

Figure 22 on page 152 shows a command set for activating the automatic requeue feature. The command set requests that:

- after cold-starting IMS A12B following a normal shutdown, a prompt asking whether to perform a `REQUEUE TYPE=COLD` be issued; if the response is yes, the command that is specified for the AUTO_REQ_CMD keyword is executed (in this case, `S IMDRDR, MBR=AUTONRE`)

- after starting IMS A12B with `/ERE COLDCOM` or `COLD SYS` following an IMS failure, a prompt asking whether to perform a `REQUEUE TYPE=EREFAIL` be issued; if the response is yes, the command that is specified for the AUTO_REQ_CMD keyword is executed (in this case, `S IMDRDR, MBR=AUTONRE`)

**Figure 22: Command set: Activate automatic requeue**

```
CUSTOMIZE
  IMS_OPTIONS
    IMSID=A12B,
    AUTO_REQ=PROMPT,
    AUTO_REQ_CMD='S IMDRDR, MBR=AUTONRE',
    AUTO_REQ_COLDCOM=PROMPT,
    AUTO_CMD_COLDCOM='S IMDRDR, MBR=AUTOERE',
    AUTO_REQ_COLDSYS=PROMPT,
    AUTO_CMD_COLDSYS='S IMDRDR, MBR=AUTOERE'
END
```

For more information about using the CUSTOMIZE command, see the installation guide.

**CUSTOMIZE command set description**

The CUSTOMIZE command set consists of the following items:

- CUSTOMIZE primary command
- subcommands with associated keywords and parameters
- END command

--- Note ---
Keyword and parameter descriptions for each subcommand are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

You can change the defaults by adding a special options entry. For the IMS_Options subcommand, add an entry with IMSid=@@@@. For the SERVER_Options subcommand, add an entry with SERVER_name=@@@@@@@@. See the IMSid and SERVER_name keyword descriptions for more information.

--- IMS_Options subcommand keywords ---
You can include an IMS_Options subcommand within the CUSTomize command set in order to specify Message Advisor customization options for each of your IMS systems.

Repeat the subcommand and its keywords for each IMS system that is running Message Advisor.

--- Note ---
To use a single IMS_Options subcommand to provide generic options for multiple IMS systems, specify a pattern for the IMSid keyword and use the %IMSID symbolic variable in data set names as necessary.

--- AUTO_CMD_COLDCOM keyword ---
AUTO_CMD_COLDCOM is an optional keyword for the IMS_Options subcommand. AUTO_CMD_COLDCOM executes a site-defined multiple virtual storage (MVS) command, which initiates a Message Advisor requeue. The MVS command executes following an IMS /ERE COLDCOMM start, if AUTO_REQ_COLDCOM=Yes or PROMPT. You must create the MVS task before initiating an automatic requeue.

Table 137 on page 154 lists the AUTO_CMD_COLDCOM keyword parameter.
Table 137: AUTO_CMD_COLDCOM keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>up to 88 characters</td>
<td>For a string to continue across multiple lines, the string must extend through the last valid column on the current line (column 71 for a standard, 80-byte fixed-length record). The first non-blank character on the next line must be the delimiter that you typed at the beginning of the string (normally a quote). To use quotes in a string, type C and another character as the delimiter (C/S X,A='123'/). The AUTO_CMD_COLDCOM keyword value can include the symbolic keyword %IMSID. The current IMS ID replaces %IMSID. A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value. System symbols may be used as part of the string. System symbols are resolved either at IMS restart or completion of the next checkpoint SNAPQ.</td>
</tr>
</tbody>
</table>

**AUTO_CMD_COLDSYS keyword**

AUTO_CMD_COLDSYS is an optional keyword for the IMS_Options subcommand. AUTO_CMD_COLDSYS executes a site-defined MVS command, which initiates a Message Advisor requeue. The MVS command executes following an IMS /ERE COLDSYS start, if AUTO_REQ_COLDSYS=Yes or PROMPT. You must create the MVS task before initiating an automatic requeue.

Table 138 on page 154 lists the AUTO_CMD_COLDSYS keyword parameter.

Table 138: AUTO_CMD_COLDSYS keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>up to 88 characters</td>
<td>For a string to continue across multiple lines, the string must extend through the last valid column on the current line (column 71 for a standard, 80-byte fixed-length record). The first non-blank character on the next line must be the delimiter that you typed at the beginning of the string (normally a quote). To use quotes in a string, type C and another character as the delimiter (C/S X,A='123'/). The AUTO_CMD_COLDSYS keyword value can include the symbolic keyword %IMSID. The current IMS ID replaces %IMSID. A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value. System symbols may be used as part of the string. System symbols are resolved either at IMS restart or completion of the next checkpoint SNAPQ.</td>
</tr>
</tbody>
</table>
**AUTO_REQ keyword**

AUTO_REQ is an optional keyword for the IMS_Options subcommand and indicates whether Message Advisor automatically initiates a requeue following a normal IMS cold start (/NRE CHKPT 0).

Table 139 on page 155 lists the AUTO_REQ keyword parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Message Advisor does not automatically initiate a requeue after a normal IMS cold start (/NRE CHKPT 0).</td>
</tr>
<tr>
<td>Yes</td>
<td>Message Advisor automatically initiates a requeue after a normal IMS cold start (/NRE CHKPT 0).</td>
</tr>
<tr>
<td>PROMPT</td>
<td>Message Advisor issues a WTOR before executing the MVS command specified for the AUTO_REQ_CMD keyword. The prompt lets you prevent Message Advisor from executing the MVS command. <strong>WARNING:</strong> If you specify AUTO_REQ=PROMPT, you must reply to the WTOR because the IMS restart process waits for a reply. <strong>Note:</strong> BMC Software recommends that, if you specify AUTO_REQ=PROMPT, you also specify REQ_PRompt=Yes for the SERVER_Options subcommand or for the REQueue command.</td>
</tr>
</tbody>
</table>

**AUTO_REQ_CMD keyword**

AUTO_REQ_CMD is an optional keyword for the IMS_Options subcommand. AUTO_REQ_CMD executes a site-defined MVS command, which initiates a Message Advisor requeue. The MVS command executes following a normal IMS cold start (/NRE CHKPT 0) if AUTO_REQ=Yes or PROMPT. You must create the MVS task before initiating an automatic requeue.

Table 140 on page 156 lists the AUTO_REQ_CMD keyword parameter.
Table 140: AUTO_REQ_CMD keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>up to 88 characters</td>
<td>For a string to continue across multiple lines, the string must extend through the last valid column on the current line (column 71 for a standard, 80-byte fixed-length record). The first non-blank character on the next line must be the delimiter that you typed at the beginning of the string (normally a quote). To use quotes in a string, type C and another character as the delimiter (C/S X,A='123'/). The AUTO_REQ_CMD keyword value can include the symbolic keyword %IMSID. The current IMS ID replaces %IMSID. A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value. System symbols may be used as part of the string. System symbols are resolved either at IMS restart or completion of the next checkpoint SNAPQ.</td>
</tr>
</tbody>
</table>

AUTO_REQ_COLDCOM keyword

AUTO_REQ_COLDCOM is an optional keyword for the IMS_Options subcommand and indicates whether Message Advisor automatically initiates a requeue following an IMS /ERE COLDCOMM start.

Table 141 on page 156 lists the AUTO_REQ_COLDCOM keyword parameters.

Table 141: AUTO_REQ_COLDCOM keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Message Advisor does not automatically initiate a requeue following an IMS /ERE COLDCOMM start.</td>
</tr>
<tr>
<td>Yes</td>
<td>Message Advisor automatically initiates a requeue following an IMS /ERE COLDCOMM start.</td>
</tr>
<tr>
<td>PROMPT</td>
<td>Message Advisor issues a WTOR before executing the MVS command specified for the AUTO_CMD_COLDCOM keyword. The prompt allows you to prevent Message Advisor from executing the MVS command. <strong>WARNING:</strong> If you specify AUTO_REQ_COLDCOM=PROMPT, you must reply to the WTOR because the IMS /ERE COLDCOMM process waits for a reply. <strong>Note:</strong> BMC recommends that, if you specify AUTO_REQ_COLDCOM=PROMPT, you also specify REQ_PROMPT=Yes for the SERVER_Options subcommand or for the REQueue command.</td>
</tr>
</tbody>
</table>
**AUTO_REQ_COLDSYS keyword**

AUTO_REQ_COLDSYS is an optional keyword for the IMS_Options subcommand and indicates whether Message Advisor automatically initiates a requeue following an IMS /ERE COLDSYS start.

Table 142 on page 157 lists the AUTO_REQ_COLDSYS keyword parameters.

**Table 142: AUTO_REQ_COLDSYS keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Message Advisor does not automatically initiate a requeue following an IMS /ERE COLDSYS start.</td>
</tr>
<tr>
<td>Yes</td>
<td>Message Advisor automatically initiates a requeue following an IMS /ERE COLDSYS start.</td>
</tr>
<tr>
<td>PROMPT</td>
<td>Message Advisor issues a WTOR before executing the MVS command specified for the AUTO_CMD_COLDSYS keyword. The prompt allows you to prevent Message Advisor from executing the MVS command. <strong>WARNING:</strong> If you specify AUTO_REQ_COLDSYS=PROMPT, you must reply to the WTOR because the IMS /ERE COLDSYS process waits for a reply. <strong>Note:</strong> BMC recommends that, if you specify AUTO_REQ_COLDSYS=PROMPT, you also specify REQ_PRompt=Yes for the SERVER_Options subcommand or for the REQueue command.</td>
</tr>
</tbody>
</table>

**CHEckpoint_dsn keyword**

CHEckpoint_dsn is an optional keyword for the IMS_Options subcommand.

Table 143 on page 157 lists the CHEckpoint_dsn keyword parameter.

**Table 143: CHEckpoint_dsn keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCNODE.%IMSID.CHKPT up to 44 characters</td>
<td>the name of the data set that tracks recent IMS start, stop, and SNAPQ checkpoints. The name should match the name of the Message Advisor Checkpoint Tracking data set allocated during installation. The CHEckpoint_dsn keyword value can include the symbolic keyword %IMSID. The current IMS ID replaces %IMSID. A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value.</td>
</tr>
</tbody>
</table>
**DEFAULT_SERVER keyword**

DEFAULT_SERVER is an optional keyword for the IMS_Options subcommand and indicates the default Message Advisor Server to be used for an IMS when only the IMS ID is known.

Table 144 on page 158 lists the DEFAULT_SERVER keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default_server</td>
<td>the name of the Message Advisor Server to be used to process requests for a specific IMS ID</td>
</tr>
</tbody>
</table>

**DELeete keyword**

DELeete is an optional keyword for the IMS_Options subcommand and indicates whether to remove an IMS environment from the options module.

Table 145 on page 158 lists the DELeete keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>removes an IMS environment from the options module</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>If you include the DELeete keyword with the IMS_Options subcommand, it must be the only keyword (other than the IMSid keyword that identifies the IMS environment).</td>
</tr>
</tbody>
</table>

**EXTract_dsn keyword**

EXTract_dsn is an optional keyword for the IMS_Options subcommand.

Table 146 on page 159 lists the EXTract_dsn keyword parameter.
Table 146: EXTract_dsn keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCNODE.%IMSID.EXTRACT</td>
<td>the name of the data set that holds messages extracted from the IMS log file for later requeue</td>
</tr>
<tr>
<td></td>
<td>You can use the EXTRACT subcommand on the REQUEUE command to override the EXTract_dsn keyword.</td>
</tr>
<tr>
<td></td>
<td>The EXTract_dsn keyword value can include the symbolic keywords %IMSID and %USER. The current IMS ID replaces %IMSID; the requestor’s RACF user ID replaces %USER. The requestor is the user that submits the REQUEUE request.</td>
</tr>
<tr>
<td></td>
<td>A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value.</td>
</tr>
</tbody>
</table>

**IMSid keyword**

IMSid is an optional keyword for the IMS_Options subcommand.

Table 147 on page 159 lists the IMSid keyword parameters.

Table 147: IMSid keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccc</td>
<td>the IMS system to which this IMS_Options subcommand applies</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> cccc is a value that you specify.</td>
</tr>
<tr>
<td></td>
<td><strong>IMSid masking feature:</strong></td>
</tr>
<tr>
<td></td>
<td>Masking characters (*) and (?) are allowed for the IMSid keyword. When you use these characters, a generic options entry is added, updated, or deleted for the specified IMS ID pattern.</td>
</tr>
<tr>
<td></td>
<td>If no IMS options exist for the specified IMS ID, the first matching generic entry is used when options are required. The options are sorted such that the asterisk (*) sorts last and the question mark (?) sorts next to last.</td>
</tr>
<tr>
<td></td>
<td><strong>WARNING:</strong> If generic options are added for IMSid=*, Message Advisor executes all command sets because it always finds a matching set of options.</td>
</tr>
<tr>
<td>@@@@@</td>
<td>special IMS ID that provides defaults when building new options for other IMS IDs</td>
</tr>
<tr>
<td></td>
<td>Specify IMSid=@@@@ to reset current default keyword values to different values.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Message Advisor uses the @@@@@ defaults only when executing the CUSTOMize command set to build options for a new IMS ID.</td>
</tr>
</tbody>
</table>
**LOGcode keyword**

LOGcode is an optional keyword for the IMS_Options subcommand.

Table 148 on page 160 lists the LOGcode keyword parameter.

**Table 148: LOGcode keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| EF cc     | prefix that identifies diagnostic and error records that Message Advisor writes to the IMS log  
**Note:** cc is a value that you specify.  
Ensure that the code is unique to your IMS system. |

**NRE_SNAPQ keyword**

NRE_SNAPQ is an optional keyword for the IMS_Options subcommand and indicates the conditions under which Message Advisor for IMS takes a /CHE SNAPQ following a successful /NRE or /NRE CHKPT 0.

Table 149 on page 160 lists the NRE_SNAPQ keyword parameters.

**Table 149: NRE_SNAPQ keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>SNAPQ never occurs following /NRE.</td>
</tr>
<tr>
<td>ALL</td>
<td>SNAPQ occurs following both /NRE CHKPT 0 and /NRE.</td>
</tr>
<tr>
<td>COLD</td>
<td>SNAPQ only occurs following /NRE CHKPT 0.</td>
</tr>
<tr>
<td>WARM</td>
<td>SNAPQ only occurs following /NRE.</td>
</tr>
</tbody>
</table>

**RATE keyword**

RATE is an optional keyword for the IMS_Options subcommand.

Table 150 on page 161 lists the RATE keyword parameter.
Table 150: RATE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>the number of messages or destinations that Message Advisor requeues, dequeues, unloads, or displays per minute (the message queue activity rate) Specify 0 if you do not desire rate control. You can use the RATE keyword to override a 0 value when executing a command set. <strong>Note:</strong> <code>nnnnnnnn</code> is a value that you specify. One message per second is the slowest rate that can be achieved with RATE values 1-60.</td>
</tr>
</tbody>
</table>

**RECON1_dsn keyword**

RECON1_dsn is an optional keyword for the IMS_Options subcommand.  

---

**Note**  
If you specify a value for the RECON1_dsn keyword, the value overrides any value that you specify for the RECON_DYNALLOC keyword.

---

Table 151 on page 161 lists the RECON1_dsn keyword parameters.

Table 151: RECON1_dsn keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSVS.%IMSID.RECON1 up to 44 characters</td>
<td>the name of the RECON1 data set that stores the requeue log input for this IMS ID Message Advisor will examine up to three RECON data sets, but will use only the RECON data set marked Copy1. The RECON1_dsn keyword value can include the symbolic keyword %IMSID. The current IMS ID replaces %IMSID. A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value.</td>
</tr>
<tr>
<td>DBRC</td>
<td>if you specify RECON1_dsn=DBRC, Message Advisor uses the open RECON data set names located in the DBRC address space to determine the Copy1 RECON data set; Message Advisor ignores any data sets you specify for the RECON2_dsn and RECON3_dsn keywords</td>
</tr>
</tbody>
</table>

**RECON2_dsn keyword**

RECON2_dsn is an optional keyword for the IMS_Options subcommand.  

Table 152 on page 162 lists the RECON2_dsn keyword parameter.
Table 152: RECON2_dsn keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| IMSVS.%IMSID.RECON2 up to 44 characters | the name of the RECON2 data set for this IMS ID  
The RECON2_dsn keyword value can include the symbolic keyword %IMSID. The current IMS ID replaces %IMSID.  
A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value. |

**RECON3_dsn keyword**

RECON3_dsn is an optional keyword for the IMS_Options subcommand.

Table 153 on page 162 lists the RECON3_dsn keyword parameter.

Table 153: RECON3_dsn keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| IMSVS.%IMSID.RECON3 up to 44 characters | the name of the RECON3 data set for this IMS ID  
If you do not use RECON3, specify a null entry (RECON3_dsn=,).  
The RECON3_dsn keyword value can include the symbolic keyword %IMSID. The current IMS ID replaces %IMSID.  
A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value. |

**RECON_DYNALLOC keyword**

RECON_DYNALLOC is an optional keyword for the IMS_Options subcommand.

Table 154 on page 163 lists the RECON_DYNALLOC keyword parameter.
Table 154: RECON_DYNALLOC keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dsname</td>
<td>the name of the IMS RESLIB or an APF-authorized library that contains the RECON dynamic allocation modules for this IMS ID. The RECON_DYNALLOC keyword value can include the symbolic keyword %IMSID. The current IMS ID replaces %IMSID. A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value. Note: If you specify a value for the RECON1_dsn keyword, the value overrides any value that you specify for the RECON_DYNALLOC keyword.</td>
</tr>
</tbody>
</table>

**SCRap_dsn keyword**

SCRap_dsn is an optional keyword for the IMS_Options subcommand. Table 155 on page 163 lists the SCRap_dsn keyword parameter.

Table 155: SCRap_dsn keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCNODE. %IMSID.SCRAP</td>
<td>the name of the data set for holding messages that cannot be requeued using the REQueue command. Message Advisor uses the data set when you specify SCRAP= YES for the target IMS ID. You can use the SCRAP subcommand on the REQUEUE command to override the SCRap_dsn keyword. The SCRap_dsn keyword value can include the symbolic keywords %IMSID and %USER. The current IMS ID replaces %IMSID; the requestor’s RACF user ID replaces %USER. A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value.</td>
</tr>
</tbody>
</table>

**SPILL1_dsn keyword**

SPILL1_dsn is an optional keyword for the IMS_Options subcommand. Table 156 on page 164 lists the SPILL1_dsn keyword parameter.
Table 156: SPILL1_dsn keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| BMCNODE.%IMSID.SPILL1   | the name of the working data set to use if a requeue requires overflow from main storage  
 You can use the SPILL subcommand on the REQUEUE command to override the SPILL1_dsn keyword.  
 If Message Advisor runs out of extended private storage during the requeue process, SPILL1_dsn is required for TYPE=COLD, REPRocess, or EREfail.  
 The SPILL1_dsn keyword value can include the symbolic keyword %IMSID. The current IMS ID replaces %IMSID.  
 A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value. |
| up to 44 characters      |                                                                                                                                                                                                                                                                                                                                            |

**SPILL2_dsn keyword**

SPILL2_dsn is an optional keyword for the IMS_Options subcommand.

Table 157 on page 164 lists the SPILL2_dsn keyword parameter.

Table 157: SPILL2_dsn keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| BMCNODE.%IMSID.SPILL2   | the name of the working data set to use for data extracted from log records  
 You can use the SPILL subcommand on the REQUEUE command to override the SPILL2_dsn keyword.  
 The data is processed by sort (TYPE=EREfail and TYPE=REPRocess).  
 The SPILL2_dsn keyword value can include the symbolic keyword %IMSID. The current IMS ID replaces %IMSID.  
 A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value. |
| up to 44 characters      |                                                                                                                                                                                                                                                                                                                                            |

**SPILL3_dsn keyword**

SPILL3_dsn is an optional keyword for the IMS_Options subcommand.

Table 158 on page 165 lists the SPILL3_dsn keyword parameter.
### Table 158: SPILL3_dsn keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCNODE.%IMSID.SPILL3 up to 44 characters</td>
<td>the name of the working data set to use when a SNAPQ checkpoint begins requeue processing (TYPE=EREfail and TYPE=REPRocess) and overflow from main storage occurs. You can use the SPILL subcommand on the REQUEUE command to override the SPILL3_dsn keyword. The SPILL3_dsn keyword value can include the symbolic keyword %IMSID. The current IMS ID replaces %IMSID. A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value.</td>
</tr>
</tbody>
</table>

### SPILL4_dsn keyword

SPILL4_dsn is an optional keyword for the IMS_Options subcommand. 

Table 159 on page 165 lists the SPILL4_dsn keyword parameter.

### Table 159: SPILL4_dsn keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCNODE.%IMSID.SPILL4 up to 44 characters</td>
<td>the name of the working data set to use when overflow from main storage occurs while processing 01 and 03 log records (TYPE=EREfail and TYPE=REPRocess). You can use the SPILL subcommand on the REQUEUE command to override the SPILL4_dsn keyword. The SPILL4_dsn keyword value can include the symbolic keyword %IMSID. The current IMS ID replaces %IMSID. A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value.</td>
</tr>
</tbody>
</table>

### UNLoad_dsn keyword

UNLoad_dsn is an optional keyword for the IMS_Options subcommand. 

Table 160 on page 166 lists the UNLoad_dsn keyword parameter.
Table 160: UNLoad_dsn keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCNODE.%IMSID.UNLOAD</td>
<td>the name of the data set that holds messages unloaded from the message queues by the UNLoad command or the DEQueue command when MODE=UNLoad_DEQueue</td>
</tr>
<tr>
<td></td>
<td>You can use the OUTPUT subcommand on the UNLOAD or DEQUEUE commands to override the UNLoad_dsn keyword.</td>
</tr>
<tr>
<td></td>
<td>The UNLoad_dsn keyword value can include the symbolic keywords %IMSID and %USER. The current IMS ID replaces %IMSID; the requestor’s RACF user ID replaces %USER.</td>
</tr>
<tr>
<td></td>
<td>A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the following characters are concatenated to the symbolic value.</td>
</tr>
</tbody>
</table>

VIRTUAL_create keyword

VIRTUAL_create is an optional keyword for the IMS_Options subcommand.

Table 161 on page 166 lists the VIRTUAL_create keyword parameters.

Table 161: VIRTUAL_create keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>activates the Message Advisor virtual LTERM creation feature</td>
</tr>
<tr>
<td></td>
<td>When you use Message Advisor with DELTA IMS VIRTUAL TERMINAL, EXTENDED TERMINAL ASSIST PLUS, or EXTENDED TERMINAL OPTION (ETO), the virtual LTERM creation feature allows Message Advisor to recreate deleted virtual LTERM that are required for a requeue task.</td>
</tr>
<tr>
<td>No</td>
<td>deactivates the Message Advisor virtual LTERM creation feature</td>
</tr>
</tbody>
</table>

WTO_CMDS keyword

WTO_CMDS is an optional keyword for the IMS_Options subcommand.

Table 162 on page 167 lists the WTO_CMDS keyword parameters.
Table 162: WTO_CMDS keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Message Advisor issues a WTO to the IMS joblog whenever a /START or /STOP command is issued during a DEQUEUE operation. This option should be chosen very carefully because a considerable number of WTO messages may be written during the DEQUEUE command.</td>
</tr>
<tr>
<td>No</td>
<td>This default option specifies that Message Advisor should <em>not</em> issue a WTO to the IMS joblog whenever a /START or /STOP command is issued during a DEQUEUE operation.</td>
</tr>
</tbody>
</table>

LIST_Options subcommand keywords

You can include a LIST_Options subcommand within the CUStomize command set in order to display Message Advisor customization options for each of your IMS systems and Message Advisor Servers. Repeat the subcommand and its keywords for each IMS system that is running Message Advisor, and for each Message Advisor Server.

**DETail keyword**

DETail is an optional keyword for the LIST_Options subcommand. If you specify the DETail keyword, you cannot specify the IMSid or SERVER_name keywords.

Table 163 on page 167 lists the DETail keyword parameters.

Table 163: DETail keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>specify DETail=No or use the default to create a list of IMS IDs and servers for which options have been defined</td>
</tr>
<tr>
<td>Yes</td>
<td>provides a detailed list of all option parameters for all IMS IDs and servers which have been defined</td>
</tr>
</tbody>
</table>

**IMSid keyword**

IMSid is an optional keyword for the LIST_Options subcommand.

Table 164 on page 168 lists the IMSid keyword parameter.
Table 164: IMSid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| All ccccc | the IMS system to which this LIST_Options command set applies  
**Note:** ccccc is a value that you specify.  
**IMSid masking feature:** 
Masking characters (* and ?) are allowed for the IMSid keyword. When you use these characters, the generic options entry for the specified IMS ID mask is listed. Note that only one entry is listed. If you wish to see all of the entries, you must specify IMSid=All. |

**SERVER_name keyword**

SERVER_name is an optional keyword for the LIST_Options subcommand.

Table 165 on page 168 lists the SERVER_name keyword parameter.

Table 165: SERVER_name keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| All ccccccccc | the name of the Message Advisor Server to which this LIST_Options subcommand applies  
**Note:** ccccccccc is a value that you specify. |

**SERVER_Options subcommand keywords**

You can include a SERVER_Options subcommand within the CUStomize command set in order to specify customization options for each Message Advisor Server. Repeat the subcommand and its keywords for each Message Advisor Server.

**DELete keyword**

DELete is an optional keyword for the SERVER_Options subcommand and indicates whether to remove a Message Advisor Server environment from the options module.

Table 166 on page 169 lists the DELete keyword parameter.
Table 166: DELete keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Yes       | removes a Message Advisor Server environment from the options module  
  **Note:** If you use the DELete keyword, it must be the only keyword (other than the SERVER_name keyword that identifies the server environment). |

**GRS_qname keyword**

GRS_qname is an optional keyword for the SERVER_Options subcommand.

Table 167 on page 169 lists the GRS_qname keyword parameter.

Table 167: GRS_qname keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| QMANAGER  | the global resource sharing (GRS) or equivalent name that, when enqueued with a systems-type enqueue, applies to all CPUs in the GRS environment  
  **Note:** cccccccc is a value that you specify.  
  The GRS name must be the same for all Message Advisor and Q:MANAGER IMS Servers.  
  System symbols may be used as part of the string. System symbols are resolved the next time the Message Advisor Server is started. |

**REQ_PRompt keyword**

REQ_PRompt is an optional keyword for the SERVER_Options subcommand.

Table 168 on page 170 lists the REQ_PRompt keyword parameters.
Table 168: REQ_PRompt keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Yes       | Message Advisor issues a WTOR prompt before executing a REQueue TYPE=COLD. The WTOR indicates the checkpoint that Message Advisor will use for the requeue. **Note:** If you specified any of the following for the IMS_Options subcommand, BMC Software recommends that you also specify REQ_PRompt=Yes:  
  - AUTO_REQ=Yes  
  - AUTO_REQ=PROMPT  
  - AUTO_REQ_COLDCOM=Yes  
  - AUTO_REQ_COLDCOM=PROMPT  
  - AUTO_REQ_COLDSYS=Yes  
  - AUTO_REQ_COLDSYS=PROMPT |
| No        | Message Advisor does not issue a WTOR prompt before executing a REQueue TYPE=COLD |

**REQUEUE_WTOMSG keyword**

REQUEUE_WTOMSG is an optional keyword for the SERVER_Options subcommand. REQUEUE_WTOMSG generates the WTO message BMC43254I to the console and to the Message Advisor server job log reporting how many messages were successfully requeued for the requested REQUEUE job.

This option can be used to report REQUEUE job messages that were successfully requeued for a situation where an ISPF session that generated the REQUEUE job is no longer available, due to a possible TSO timeout.

The subcommand has the following keyword:

REQUEUE_WTOMSG = NO|YES

Table 169: REQUEUE_WTOMSG keyword

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO (default)</td>
<td>Message Advisor does not report (through BMC43254I) how many messages were successfully requeued.</td>
</tr>
<tr>
<td>YES</td>
<td>Message Advisor reports how many messages were successfully requeued.</td>
</tr>
</tbody>
</table>

**SERVER_APPLid keyword**

SERVER_APPLid is an optional keyword for the SERVER_Options subcommand.
Table 170 on page 171 lists the SERVER_APPLid keyword parameter.

Table 170: SERVER_APPLid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMRP</td>
<td>the APPL ACBNAME assigned to Message Advisor running in this VTAM domain. The name must be unique throughout your network.</td>
</tr>
<tr>
<td>cccccccc</td>
<td>Note: cccccccc is a value that you specify.</td>
</tr>
</tbody>
</table>

SERVER_name keyword

SERVER_name is an optional keyword for the SERVER_Options subcommand.

Table 171 on page 171 lists the SERVER_name keyword parameters.

Table 171: SERVER_name keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMRP</td>
<td>the name of a Message Advisor environment (normally associated with a CPU or a VTAM domain) that will service IMS DC/TM systems requiring Message Advisor services. One Message Advisor address space per CPU can support all IMS address spaces on that CPU, and as many ISPF users as necessary (from any domain within your network).</td>
</tr>
<tr>
<td>cccccccc</td>
<td>Note: cccccccc is a value that you specify. BMC Software recommends that the SERVER_name keyword value match the SERVER_APPLid keyword value.</td>
</tr>
<tr>
<td>@@@@@@@@@@</td>
<td>the default Message Advisor Server for options customization, used to initialize fields in new Message Advisor Server definitions and provide starting points for Message Advisor address spaces that you are activating for the first time. Specify SERVER_name=@@@@@@@@@@ to reset current default keyword values to different values.</td>
</tr>
<tr>
<td>@@@@@@@@@@</td>
<td>Note: Message Advisor uses the @@@@@@@@@@ defaults only when executing the CUStomize command set to build options for a new server.</td>
</tr>
</tbody>
</table>

SORTcore keyword

SORTcore is an optional keyword for the SERVER_Options subcommand.

Table 172 on page 172 lists the SORTcore keyword parameter.
Table 172: SORTcore keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4096K nnnnc</td>
<td>reserves extended private area (EPVT) memory above the 16 MB line in order to perform incore sorting. Message Advisor uses EPVT when performing a REQueue TYPE=EREfail or TYPE=REPRocess. The SORTcore keyword value must match your sort product’s core reserve parameter. For SyncSort OS, the parameter is CORE; for DFSORT, the parameter is SIZE. A value larger than 4096K reduces the EPVT storage available for Message Advisor and may affect performance. Enter K or M after the sort core value to specify kilobytes or megabytes; otherwise, Message Advisor reserves core in bytes. <strong>Note:</strong> nnnnc is a value that you specify.</td>
</tr>
</tbody>
</table>

**SSCTname keyword**

SSCTname is an optional keyword for the SERVER_Options subcommand.

Table 173 on page 172 lists the SSCTname keyword parameter.

Table 173: SSCTname keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMR ccc</td>
<td>the MVS subsystem control table (SSCT) name prefix for Message Advisor. The subsystem entry is dynamically added. Normally, the name should not be changed unless it conflicts with another product’s subsystem name. <strong>Note:</strong> ccc is a value that you specify. The value for the SSCTname keyword must be the same for all Message Advisor and Q:MANAGER IMS Servers. System symbols may be used as part of the string. System symbols are resolved the next time the Message Advisor Server is started.</td>
</tr>
</tbody>
</table>

**SWAp keyword**

SWAp is an optional keyword for the SERVER_Options subcommand.

Table 174 on page 173 lists the SWAp keyword parameters.
Table 174: SWAp keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>indicates that the Message Advisor Server will not run as an address space that allows swapping</td>
</tr>
<tr>
<td>Yes</td>
<td>indicates that the Message Advisor Server will run as an address space that allows swapping</td>
</tr>
</tbody>
</table>

**TITLE keyword**

TITLE is an optional keyword for the SERVER_Options subcommand.

Table 175 on page 173 lists the TITLE keyword parameter.

Table 175: TITLE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Advisor up to 16 characters</td>
<td>the title displayed on WTORs System symbols may be used as part of the string. System symbols are resolved the next time the Message Advisor Server is started.</td>
</tr>
</tbody>
</table>

**TRACE keyword**

TRACE is an optional keyword for the SERVER_Options subcommand.

Table 176 on page 173 lists the TRACE keyword parameter.

Table 176: TRACE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000 $nnnnc</td>
<td>the number of internal trace table entries the Message Advisor address space maintains in extended ECSA. Each entry requires 100 bytes Note: BMC Software recommends that the Trace facility remain active at all times, with a trace table allocation of at least 2000. $nnnnc is a value that you specify.</td>
</tr>
</tbody>
</table>

**UPpercase keyword**

UPpercase is an optional keyword for the SERVER_Options subcommand.

Table 177 on page 174 lists the UPpercase keyword parameters.
Table 177: UPpercase keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>deactivates the uppercase-only customization feature</td>
</tr>
<tr>
<td>Yes</td>
<td>activates the uppercase-only customization feature so that reports print in uppercase</td>
</tr>
<tr>
<td></td>
<td>Uppercase allows you to print and display reports on printers that do not have lowercase capability.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If you use advanced program-to-program communications (APPC) TPNAMEs containing lowercase characters, do not specify UPpercase=Yes. If you specify UPpercase=Yes in this situation, you cannot use ISPF options 2 and 8 to process the TPNAMEs, and the TPNAMEs will display in all uppercase on reports.</td>
</tr>
</tbody>
</table>

**VTAM_LOGmode keyword**

VTAM_LOGmode is an optional keyword for the SERVER_Options subcommand.

Table 178 on page 174 lists the VTAM_LOGmode keyword parameter.

Table 178: VTAM_LOGmode keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMANAGER ccccccc</td>
<td>the name of the VTAM logmode that Message Advisor uses to access the LU 6.2 link</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> ccccccc is a value that you specify.</td>
</tr>
</tbody>
</table>

**VTAM_LU_prefix keyword**

VTAM_LU_prefix is an optional keyword for the SERVER_Options subcommand.

Table 179 on page 174 lists the VTAM_LU_prefix keyword parameter.

Table 179: VTAM_LU_prefix keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMRU cccc</td>
<td>the VTAM LU name prefix (first four characters) assigned to Message Advisor user sessions</td>
</tr>
<tr>
<td></td>
<td>The name must be unique in a domain. Only specify four characters, since Message Advisor appends a sequential number (starting with 0001, 0002,... nnnn) until a number is available for a session or an invalid LU name is detected.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> cccc is a value that you specify.</td>
</tr>
</tbody>
</table>
**WTO_DESC keyword**

WTO_DESC is an optional keyword for the SERVER_Options subcommand.

Table 180 on page 175 lists the WTO_DESC keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0600'</td>
<td>the WTOR message description code</td>
</tr>
</tbody>
</table>
| X'cccc'   | The code is specified in hexadecimal and the default equates to description codes 6,7.  
  **Note:** cccc is a value that you specify. |

**WTO_ROUTcde keyword**

WTO_ROUTcde is an optional keyword for the SERVER_Options subcommand.

Table 181 on page 175 lists the WTO_ROUTcde keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'4060'</td>
<td>the routing code for messages that Message Advisor issues</td>
</tr>
</tbody>
</table>
| X'cccc'   | The code is specified in hexadecimal and the default equates to route codes 2,10,11.  
  **Note:** cccc is a value that you specify. |

**WTOR_HIDE keyword**

WTOR_HIDE is an optional keyword for the SERVER_Options subcommand.

Table 182 on page 175 lists the WTO_HIDE keyword parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| No        | indicates that the Message Advisor Server will issue a WTOR when initialization completes  
The outstanding WTOR allows users to enter PRD commands through the MVS console. |
| Yes       | indicates that the Message Advisor Server will not issue a WTOR when initialization completes |
This chapter describes the Message Advisor for IMS DEQUEUE command set, and also provides a syntax diagram and sample command sets. For more information about building and executing a DEQUEUE command set, see the Message Advisor for IMS User Guide.

Overview

The DEQUEUE command lets you dequeue (delete) messages from the message queues. You can use the DEQUEUE command to perform the following tasks:

- dequeue all messages from a destination
- dequeue a specific message from a destination
- exit conversations and dequeue the associated messages
- unload messages to a sequential data set and dequeue them
- validate that a dequeue will be successful without actually performing the dequeue
- select or reject messages for dequeue based on the following criteria:
  - date and time ranges
  - destinations (with masking allowed)
  - destination type
  - origins (with masking allowed)
  - other message prefix and message data selection criteria
- create report data
Message Advisor allows the following masking characters (m) for most selection keywords:

- ? replaces exactly one character in a multicharacter string.
- * replaces any number of characters.
- Masking characters can be embedded in the string.

By using the DEQUEUE command to control message queue contents, you can prevent IMS outages caused by overflowing message queues.

The DEQUEUE command set consists of the DEQUEUE primary command and its associated keywords and parameters, with subcommands and their associated keywords and parameters. The DEQUEUE subcommands are as follows:

- INTERVAL subcommand dequeues messages according to a specified time range
  For more information, see “INTERVAL subcommand” on page 293.

- OUTPUT subcommand identifies the file to which Message Advisor unloads messages when MODE=UNLOAD_DEQUEUE
  For more information, see “OUTPUT, EXTRACT, and SCRAP subcommands” on page 303.

- SELECT subcommand specifies criteria to select specific destinations and/or messages to be dequeued
  For more information, see “SELECT and REJECT subcommands” on page 313.

- REJECT subcommand specifies criteria to exclude specific destinations and/or messages from being dequeued (the messages remain on the queue)
  For more information, see “SELECT and REJECT subcommands” on page 313.

For more information about building and executing a DEQUEUE command set, see the Message Advisor for IMS User Guide.

Assigning conditional steps

You can exercise additional control over a Message Advisor request that contains multiple command sets by using one or more COND pseudo command sets to insert conditional steps. Conditional steps let you perform the following tasks:

- request that Message Advisor execute a command set only if it successfully executes a previous command set
request that Message Advisor execute a command set only if it does not successfully execute a previous command set

**Note**
A COND step can (as an option) include the LABEL, IFCOND, MAXRC, and BRANCH keywords.

When you insert a COND step, a return code of 8 or higher in any command set no longer results in the remaining command sets being flushed with a BMC43111W message. Instead, Message Advisor executes all command sets (unless the COND step successfully executes a BRANCH keyword).

For more information about building and executing a COND pseudo command, see “COND pseudo command” on page 399, and see the Message Advisor for IMS User Guide.

**DEQUEUE command set syntax**

The following figure shows the DEQUEUE command set syntax.
Note

The SCOPE keyword applies to shared queues only.

Figure 23: DEQUEUE command set syntax

Sample DEQUEUE command sets

Message Advisor provides sample DEQUEUE command sets that can serve as templates when building your own DEQUEUE command sets.

The MAQSAMP library contains the sample command sets.
Table 183 on page 181 identifies the library members that contain the sample DEQUEUE command sets. MAQSAMP also contains an index in member QMR@INDX that lists all sample command sets provided with Message Advisor.

Table 183: Sample DEQUEUE command sets

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMR@DEQ1</td>
<td>dequeues all messages from a single destination</td>
</tr>
<tr>
<td>QMR@DEQ2</td>
<td>dequeues selected messages from multiple destinations</td>
</tr>
<tr>
<td>QMR@DEQ3</td>
<td>dequeues a specific message</td>
</tr>
<tr>
<td>QMR@DEQ4</td>
<td>dequeues messages based on selection criteria</td>
</tr>
</tbody>
</table>

**Note**

Many of the samples include field-level prompts (IMSID=?). When you execute a command set through the Message Advisor Interactive System Productivity Facility (ISPF) interface, you are prompted to provide a valid value for any field that contains a prompt.

If you execute a command set outside of the Message Advisor ISPF interface, you are not prompted to replace ?. You must replace all ? with a valid value before submitting the request.

The following sample command sets are similar, but not identical to, the samples provided in MAQSAMP.

**Figure 24 on page 181** shows a command set for dequeueing all messages in all queues from a specific destination. The command dequeues all messages from destination LTRM2900. Because the command set does not include **FORCE=YES**, the genned node to which the destination is assigned must be stopped.

MAQSAMP(QMR@DEQ1) contains a sample command set similar to **Figure 24 on page 181**.

**Figure 24: Command set: Dequeue from one destination**

DEQUEUE IMSID=R61P
SELECT DESTINATION=LTRM2900,QUEUE=ALL
END

**Figure 25 on page 182** shows a command set for unloading and dequeueing all messages from multiple destinations.

The sample includes the **TYPE=SYNCH_BLK** option associated with **MODE=UNLOAD_DEQUEUE**, which is the default. The SYNCH_BLK option runs significantly faster than the SYNCH_MSG option, which updates the unload file each time Message Advisor dequeues a message from the IMS system.

The command set affects destination SMB TRAN2800, and any destination starting with LTRM29. The command set unloads messages to the data set specified on the
OUTPUT statement. The messages can be requeued later by using REQUEUE TYPE=FILE.

Figure 25: Command set: Dequeue from multiple destinations

```
DEQUEUE IMSID=R61P,
  MODE=UNLOAD_DEQUEUE,TYPE=SYNCH_BLK
SELECT DESTINATION=LTRM29*, QUEUE=ALL
SELECT DESTINATION=TRAN2800, QUEUE=ALL, DESTYPE=SMB
OUTPUT DSNAME=R61IMS.UNLDEQ.WORK1
END
```

For more information about building and executing a DEQUEUE command set, see the *Message Advisor for IMS User Guide*.

**DEQUEUE command set description**

The DEQUEUE command set consists of the following items:

- DEQUEUE primary command with associated keywords and parameters
- subcommands with associated keywords and parameters
- END command

---

**Note**

Keyword and parameter descriptions for the primary command are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

Keyword and parameter descriptions for the subcommands are located in the following chapters:

- “INTERVAL subcommand” on page 293
- “OUTPUT, EXTRACT, and SCRAP subcommands” on page 303
- “SELECT and REJECT subcommands” on page 313

---

**Note**

IMS must be active before you can specify the DEQueue command.
CONVersations keyword

CONVersations is an optional keyword for the DEQueue command.

The CONVersations keyword provides the default value for the CONVID keyword for the SELECT and REJECT subcommands and can be overridden for specific selections.

Consider the following when specifying the CONVersations keyword:

- For shared queues systems, Message Advisor processes all conversational filtering at the message level. Message Advisor searches all destinations. A message that contains a conversation prefix and SPA is considered conversational. A message that does not contain a conversation prefix and SPA is considered non-conversational.

- For non-shared queues systems, the ONLY and NONE keywords provide additional filtering at the destination level. ONLY processes only conversational SMBs and CNTs that are in conversation. NONE processes non-conversational SMBs, but processes all CNTs because CNTs in conversation might also have non-conversational messages queued.

A message is considered conversational if it meets all the following requirements:

- contains a conversation prefix and SPA
- is queued to a conversational transaction or an LTERM in conversation

A message is considered non-conversational if it meets any of the following requirements:

- is queued to a non-conversational transaction or an LTERM not in conversation (even if the message contains an SPA)
- does not contain an SPA (even if the conversation is still active)

- Conversational messages will often be found on the backup queue for LTERMs. The default value for the Queue keyword for SELECT and REJECT subcommands for the DEQueue command does not include the backup queue or the suspend queue. If conversational messages on these queues are to be included in processing, you must specify Q=ALL, Q=B, or Q=S in addition to CONV=ALL or CONV=ONLY.

- Message Advisor dequeues conversational messages using the IMS /EXIT command. For dynamic terminals when the associated user is not active, only the IMS /EXIT command is needed to dequeue conversational messages. For static terminals and active dynamic terminals, the associated user must be stopped before the IMS /EXIT command can be issued to dequeue conversational messages.
The FORCE keyword controls whether the user will be stopped and if it will be restarted. If the FORCE keyword does not allow the user to be stopped, an error status code will be issued for the conversation.

IMS will exit a conversation when the associated node or user is stopped and restarted (even when the IMS /EXIT command is not issued). In some cases, held conversations might also be exited. Because the FORCE keyword indicates whether Message Advisor is allowed to stop a destination, special considerations occur when an active conversational message exists but is not selected or is rejected.

FORCE=RESETALL and FORCE=STOPAll will stop destinations without checking to see if conversations exist. This action might cause active or held conversations to be exited even when not selected.

FORCE=YES will dequeue non-conversational messages without stopping, and will stop destinations (if necessary) to dequeue conversational messages. This action might cause active or held conversations to be exited even when not selected.

FORCE=NO will not stop destinations. An error status code will be issued if an active conversation exists and the destination is not stopped.

FORCE=RESET, FORCE=RESETACT, and FORCE=STOP will process messages based on the DESTYPE keyword. When a DESTYPE is used that allows multiple types to be processed (such as DESTYPE=ALL), the user might be stopped and conversations might be exited even when not selected. If a specific DESTYPE value is used, an error status code will be issued.

For example, if both an active and held conversation exist for a static LTERM and a command set is created to dequeue the held conversation, DESTYPE=HELD will cause an error status code to be issued because stopping and restarting the user would also dequeue the active conversation. If any other DESTYPE value is used, the user will be stopped even though that action will exit the active conversation which may not have been selected.

Another example involves a dynamic LTERM that has an active conversational message queued to a transaction and has other non-conversational messages queued to the LTERM. Using DESTYPE=LTERM to dequeue the non-conversational messages will cause an error status code to be issued. Using DESTYPE=ALL allows the dequeue to be processed even thought the active conversation might be exited.

Table 184 on page 184 lists the CONVversions keyword parameters.

Table 184: CONVersations keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Message Advisor does not exit conversations. Non-conversational messages are dequeued. This value is the default value.</td>
</tr>
<tr>
<td>All/Yes</td>
<td>Message Advisor exits conversations for the selected destinations and dequeues non-conversational messages.</td>
</tr>
</tbody>
</table>


**Parameter** | **Description**
---|---
ONLY | Message Advisor exits conversations for the selected destinations, but does *not* dequeue non-conversational messages.

**FORCE keyword**

FORCE is an optional keyword for the DEQueue command. Consider the following when specifying the FORCE keyword:

- FORCE applies to LTERM$s and conversational SMBs only. Message Advisor always processes non-conversational SMBs, regardless of the FORCE setting.

- For conversational SMBs, FORCE indicates how Message Advisor processes users in conversation with the SMB. **FORCE=No** and **FORCE=Reset** are the same and indicate that Message Advisor does not dequeue messages for active users. Message Advisor only dequeues conversational messages for inactive users. Any other FORCE value indicates that Message Advisor forces active users off so that the conversation can be exited.

- For shared queues, **FORCE=Yes** forces direct deletions rather than using the standard CQS interface. Direct deletions are also used if IN Terral, SE lect, or RE Ject subcommands that include message selection keywords are present.

Table 185 on page 185 lists the FORCE keyword parameters. This table describes how the FORCE keyword parameters apply to LTERM$s. For more information on the effect of the FORCE command on LTERM$s that are in conversation, see “CONVersations keyword” on page 183.

**Table 185: FORCE keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Reset | If necessary, Message Advisor stops inactive LTERM$s before dequeueing messages and restarts them after dequeueing messages. For inactive dynamic LTERM$s, the associated user will also be stopped and restarted. This value is the default value. This only applies to LTERM$s that are inactive (no users logged on), not already stopped, and have messages to be dequeued. If an LTERM is active or Message Advisor cannot stop an LTERM, **FORCE=Reset** fails and Message Advisor does not attempt a dequeue. For additional considerations for LTERM$s in conversation, see “CONVersations keyword” on page 183.  
**Note:** **FORCE=Reset** has no effect if Message Advisor does not dequeue messages. Therefore, Message Advisor does not stop LTERM$s with no matching messages. |
<p>| No | Message Advisor will only dequeue messages from LTERM$s that are already stopped. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESETACTive</td>
<td>Message Advisor stops the LTERM before dequeueing messages and restarts it after dequeueing messages. For dynamic LTERMs, the associated user will also be stopped and restarted. This only applies to LTERMs that are not already stopped, but applies even if the LTERM is active (any user logged on is forced off by the STOP command). If Message Advisor cannot stop the LTERM, FORCE=RESETACTive fails and Message Advisor does not attempt a dequeue. For additional considerations for LTERMs in conversation, see “CONVersations keyword” on page 183.</td>
</tr>
</tbody>
</table>
| RESETALL      | Message Advisor stops and restarts all LTERMs that match SELect and REJect subcommands, even if:  
|               | ■ the LTERMs have no messages to dequeue  
|               | ■ MODE=VALIDate  
|               | ■ the LTERM is active  
|               | ■ the LTERM is in conversation  
|               | Note: IMS cancels active conversations when an LTERM is restarted.  
|               | Note: Message Advisor processes FORCE=RESETALL even if it does not dequeue messages. You can use FORCE=RESETALL with the UNLoad command or with DEQueue MODE=VALIDate to stop LTERMs without dequeueing messages. FORCE=RESETALL also lists all destinations that match (even those without messages). |
| Stop          | If necessary, Message Advisor stops the LTERM before dequeueing messages. For dynamic LTERMs, the associated user will also be stopped. This applies even if the LTERM is active (any user logged on is forced off by the STOP command). If Message Advisor cannot stop the LTERM, it does not dequeue messages. In any case, Message Advisor does not restart the LTERM or the associated user. For additional considerations for LTERMs in conversation, see “CONVersations keyword” on page 183.  
|               | Note: FORCE=Stop has no effect if Message Advisor does not dequeue messages. Therefore, Message Advisor does not stop LTERMs with no matching messages. |
| STOPAll       | Message Advisor stops all LTERMs that match SELect and REJect subcommands, even if:  
|               | ■ the LTERMs have no messages to dequeue  
|               | ■ MODE=VALIDate  
|               | ■ the LTERM is active  
|               | ■ the LTERM is in conversation  
|               | Note: IMS cancels active conversations when an LTERM is restarted.  
|               | Note: Message Advisor processes FORCE=STOPAll even if it does not dequeue messages. You can use FORCE=STOPAll with the UNLoad command and with DEQueue MODE=VALIDate to stop LTERMs without dequeueing messages. FORCE=STOPAll lists all destinations that match (even those without messages). |
IMSid keyword

IMSid is a required keyword for the DEQueue command.

Table 186 on page 187 lists the IMSid keyword parameter.

Table 186: IMSid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| cccc      | the IMS system to which this DEQueue command set applies  

Note: cccc is a value that you specify.

LOG keyword

LOG is an optional keyword for the DEQueue command and indicates whether to log DEQueue activities for shared queues. Message Advisor ignores the LOG keyword for non-shared queues.

Note

The LOG keyword applies to shared queues only.

Table 187 on page 187 lists the LOG keyword parameters.

Table 187: LOG keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>CQS is used to dequeue messages with normal logging</td>
</tr>
<tr>
<td>NO</td>
<td>messages are dequeued directly from the coupling facility structure with no logging, which means that a structure recovery will recover the dequeued messages</td>
</tr>
</tbody>
</table>
MODE keyword

MODE is an optional keyword for the DEQueue command.

Table 188 on page 188 lists the MODE keyword parameters.

Table 188: MODE keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEQueue</td>
<td>dequeues selected messages</td>
</tr>
<tr>
<td>UNLoad_DEQueue</td>
<td>unloads messages to an output file, reads the output file record, deletes the messages from IMS, and updates the output file</td>
</tr>
<tr>
<td>VALIDate</td>
<td>determines whether a dequeue will be successful without actually performing the dequeue</td>
</tr>
<tr>
<td></td>
<td>Note: When MODE=VALIDate, destinations are not stopped if FORCE=Reset, RESETACTIVE, or Stop. If FORCE=RESETALL or STOPAll, destinations are stopped.</td>
</tr>
</tbody>
</table>

RATE keyword

RATE is an optional keyword for the DEQueue command.

Table 189 on page 188 lists the RATE keyword parameter.

Table 189: RATE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnnnnn</td>
<td>the number of selected destinations processed per minute</td>
</tr>
<tr>
<td></td>
<td>Note: nnnnnnnnn is a value that you specify.</td>
</tr>
</tbody>
</table>

REPORT/PRINT keyword

REPORT is an optional keyword for the DEQueue command.

You can list multiple values for the REPORT keyword. If you list multiple values, enclose the values in parentheses. For example, REPORT=(DESTination,SUMmary).

You can use PRINT as an alias for REPORT.

Table 190 on page 189 lists the REPORT keyword parameters.
Table 190: REPORT keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>prints a summary and destination report that contains information regarding each message DRRN that Message Advisor selected and dequeued</td>
</tr>
<tr>
<td>DESTination</td>
<td>prints a destination report that contains a line for each queue from which messages were selected. Separate lines are always printed for each conversational message selected.</td>
</tr>
<tr>
<td>DRRN</td>
<td>prints a destination report that contains a line for each message DRRN that Message Advisor selected. The DRRN option is ignored unless <code>MODE=UNLOAD_DEQUEUE</code> is specified.</td>
</tr>
<tr>
<td>None</td>
<td>Message Advisor does not print report output</td>
</tr>
<tr>
<td>SUMMARY</td>
<td>prints a summary of messages dequeued, total destinations in error, and total destinations selected</td>
</tr>
</tbody>
</table>

**SCOpe keyword**

SCOpe is an optional keyword for the DEQueue command and limits dequeue processing to destinations that match the specified scope. You can specify a list of values for the SCOpe keyword.

---

**Note**
The SCOpe keyword applies to shared queues only.

---

Table 191 on page 189 lists the SCOpe keyword parameters. Figure 26 on page 190 illustrates the queues that are processed for each parameter.

Table 191: SCOpe keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLQ</td>
<td>includes both local and shared queues in dequeue processing</td>
</tr>
</tbody>
</table>
| LOCALQ    | limits dequeue processing to local queues  
  **Note:** You can use LCLQ as an alias for LOCALQ. |
| OVERFLOWQ | limits dequeue processing to the overflow shared queues  
  **Note:** You can use OVFQ as an alias for OVERFLOWQ. |
| PRIMARYQ  | limits dequeue processing to the primary shared queues  
  **Note:** You can use PRIQ as an alias for PRIMARYQ. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| SHAREDQ   | limits dequeue processing to shared queues (both primary and overflow)  
**Note:** You can use SHRQ as an alias for SHAREDQ.  
SHAREDQ is mutually exclusive with PRIMARYQ and OVERFLOWQ. |

**Figure 26: SCOpe keyword parameters and associated queues processed**

SCOPE=

![Diagram showing queues and parameters]

**TYPE keyword**

TYPE is an optional keyword for the DEQueue command.

**Note**
You can only specify the TYPE keyword when MODE=UNLoad_DEQueue.

Table 192 on page 191 lists the TYPE keyword parameters.
Table 192: TYPE keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNCH_BLK</td>
<td>enhances performance when MODE=UNLoad_DEQueue by limiting updates to the unload file</td>
</tr>
<tr>
<td>SYNCH_MSG</td>
<td>when MODE=UNLoad_DEQueue, indicates that Message Advisor is to update the unload file each time it deletes a message from IMS</td>
</tr>
</tbody>
</table>
DISPLAY command

This chapter describes the Message Advisor for IMS DISPLAY command set, and also provides a syntax diagram and sample command sets. For more information about building and executing a DISPLAY command set, see the Message Advisor for IMS User Guide.

Overview

The DISPLAY command lets you display the following information about your IMS message queues:

- number of messages that are in the message queues
- number of messages that are queued to destinations meeting your selection criteria
- summary of messages that are queued to a single destination
- summary of messages that are in temporary queues (for example, messages that the dependent regions have created but not committed)
- message contents

Message Advisor allows the following masking characters \((m)\) for most selection keywords:

- ? replaces exactly one character in a multicharacter string.
- * replaces any number of characters.
- Masking characters can be embedded in the string.

The DISPLAY command set performs different functions based on the TYPE keyword parameter values:

- DESTINATION
Without limiting parameters, the DISPLAY TYPE=DEST_QUEUES command is resource-intensive because Message Advisor scans the IMS message queues in their entirety to satisfy the request. To minimize effects on your IMS system, BMC Software recommends adding the SORT_BY=QCT keyword.

External security (for example, RACF or ACF2) or internal Message Advisor security (module QMRXSMA0) may restrict DISPLAY TYPE=DEST_QUEUES command usage. For more information, see the installation guide.

The DISPLAY command normally processes both permanent destination queues and temporary queues. You can specify SELECT REGION=* to process only temporary queues, or SELECT REGION_ID=0 to process only destination queues.

You can use the DISPLAY command to display message queue data, and then use the information to determine whether to dequeue messages. If the message queues are getting full, you can select or reject messages to unload and dequeue. By controlling message queue contents, you can prevent IMS outages caused by overflowing message queues.

The DISPLAY command set consists of the DISPLAY primary command and its associated keywords and parameters, with subcommands and their associated keywords and parameters. The DISPLAY subcommands are as follows:

- INTERVAL subcommand displays messages according to a specified time range
  For more information, see “INTERVAL subcommand” on page 293.

- SELECT subcommand specifies criteria to select specific destinations and/or messages to display
  For more information, see “SELECT and REJECT subcommands” on page 313.

- REJECT subcommand specifies criteria to exclude specific destinations and/or messages from being displayed
  For more information, see “SELECT and REJECT subcommands” on page 313.

For more information about building and executing a DISPLAY command set, see the Message Advisor for IMS User Guide.
Assigning conditional steps

You can exercise additional control over a Message Advisor request that contains multiple command sets by using one or more COND pseudo command sets to insert conditional steps. Conditional steps let you perform the following tasks:

- request that Message Advisor execute a command set only if it successfully executes a previous command set

—OR—

- request that Message Advisor execute a command set only if it does not successfully execute a previous command set

**Note**

A COND step can (as an option) include the LABEL, IFCOND, MAXRC, and BRANCH keywords.

When you insert a COND step, a return code of 8 or higher in any command set no longer results in the remaining command sets being flushed with a BMC43111W message. Instead, Message Advisor executes all command sets (unless the COND step successfully executes a BRANCH keyword).

For more information about building and executing a COND pseudo command, see “COND pseudo command” on page 399, and see the Message Advisor for IMS User Guide.

**DISPLAY command set syntax**

The following figures show the syntax for each DISPLAY command set.
**Note**

The following items apply to shared queues only:

- **SCOPE keyword**

- For the `DISPLAY TYPE=DEST_QUEUES` command set, `SORT_BY=%OVERFLOW` and `SORT_BY=%PRIMARY`

- For the `DISPLAY TYPE=RECORD` command set, `ENTRYID` keyword

- `DISPLAY TYPE=ANALYZER` command set

---

**Figure 27: DISPLAY TYPE=DESTINATION command set syntax**

```
   Display — IMSId=cccc — ,TYPE=DESTination

   | ,LIMIT=nnnnnnnn
   | ,RATE=nnnnnnnn
   | ,SCOpe= ALLQ
   | ( LOCALQ/LCLQ
   | OVERFLOWQ/OVFQ
   | PRIMARYQ/PRIQ
   | SHAREDQ/SHRQ
   END
```

**Figure 28: DISPLAY TYPE=DEST_QUEUES command set syntax**

```
   Display — IMSId=cccc — ,TYPE=DEST_Queues/DQ

   | ,LIMIT=nnnnnnnn
   | ,LIST_Queues/LQ= MATCH ALL NOTEMPTY
   | ,RATE=nnnnnnnn
   | ,SCOpe= ALLQ
   | ( LOCALQ/LCLQ
   | OVERFLOWQ/OVFQ
   | PRIMARYQ/PRIQ
   | SHAREDQ/SHRQ
   ,SORT_by= %Lmsg
   | %SHmsg
   | %OVERflow
   | %PRImary
   | #MSGS/MESSages
   | #SEGS/SEGments
   | QCT
   END
```
Figure 29: DISPLAY TYPE=MESSAGE command set syntax

```
DISPLAY IMSid=cccc, TYPE=MESSAGE

.. FORMat=  No  Yes
   .. LIMIT=n.nn
   .. MAXMSGS=n.nn
   .. RATE=n.nn

.. SCOpe=  ALLOQ
   .. LOCALQ/LCLQ
   .. OVERFLOWQ/OVFQ
   .. PRIMARYQ/PRIQ
   .. SHAREDQ/SHRQ

END
```

Figure 30: DISPLAY TYPE=RECORD command set syntax

```
DISPLAY IMSid=cccc, TYPE=REcord

.. FORMat=  No  Yes
   .. DRe=m
   .. ENTRYID=xxxxx

.. SCOpe=  ALLOQ
   .. LOCALQ/LCLQ
   .. OVERFLOWQ/OVFQ
   .. PRIMARYQ/PRIQ
   .. SHAREDQ/SHRQ

END
```

Figure 31: DISPLAY TYPE=STATISTICS command set syntax

```
DISPLAY IMSid=cccc, TYPE=STATistics

END
```

Figure 32: DISPLAY TYPE=ANALYZER command set syntax

```
DISPLAY IMSid=cccc, ANALyzier

END
```
Sample DISPLAY command sets

Message Advisor provides sample DISPLAY command sets that can serve as templates when building your own DISPLAY command sets. The MAQSAMP library contains the sample command sets.

Table 193 on page 198 identifies the library members that contain the sample DISPLAY command sets. MAQSAMP also contains an index in member QMR@INDX that lists all sample command sets provided with Message Advisor.

Table 193: Sample DISPLAY command sets

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMR@DSP1</td>
<td>displays message queue statistics</td>
</tr>
<tr>
<td>QMR@DSP2</td>
<td>displays destination queue information</td>
</tr>
<tr>
<td>QMR@DSP3</td>
<td>displays destination information</td>
</tr>
<tr>
<td>QMR@DSP4</td>
<td>displays record information</td>
</tr>
</tbody>
</table>

Note
Many of the samples include field-level prompts (IMSID=?). When you execute a command set through the Message Advisor Interactive System Productivity Facility (ISPF) interface, you are prompted to provide a valid value for any field that contains a prompt.

If you execute a command set outside of the Message Advisor ISPF interface, you are not prompted to replace ?. You must replace all ? with a valid value before submitting the request.

The following sample command sets are similar, but not identical to, the samples provided in MAQSAMP.

Figure 33 on page 198 shows a command set requesting a detailed report of destinations with messages. The command set requests a report for all destinations starting with $R61P.

Figure 33: Command set: Display detailed destination report

```
DISPLAY IMSID=R61P,TYPE=DESTINATION
SELECT DESTINATION=$R61P*
END
```

In Figure 34 on page 199, the first command set requests display of any destination type that has messages and for which the destination name ends in '2.' The second command set requests display of message types MSNAME (i.e., remote CNT), APPC, and OTMA and transactions of any type (TRAN1, TSERL4, or TSUSP3) as
long as there are at least ten messages for the type. MAQSAMP(QMR@DSP2) contains a sample command set similar to Figure 34 on page 199.

**Figure 34: Command set: Display summary destination queues report**

```
DISPLAY IMSID=PR81,TYPE=DEST_QUEUES
  SELECT DESTINATION=*/2
END

DISPLAY IMSID=PR81,TYPE=DEST_QUEUES
  SELECT DESTYPE=(MSNAME,TPNAME,OTMA,SMB),
  QPOSITION=(10,LAST)
END
```

Figure 35 on page 199 shows a command set requesting a message record prefix and text in hexadecimal format. For this sample, the device relative record number (DRRN) was determined from a previously-executed Message Advisor DISPLAY command. MAQSAMP(QMR@DSP4) contains a sample command set similar to Figure 35 on page 199.

**Figure 35: Command set: Display a message record**

```
DISPLAY IMSID=R61P,TYPE=RECORD,DRRN=080000EC
END
```

Figure 36 on page 199 shows a command set requesting a message record prefix in formatted mode. For this sample, the DRRN was determined from a previously-executed Message Advisor DISPLAY command. MAQSAMP(QMR@DSP4) contains a sample command set similar to Figure 36 on page 199.

**Figure 36: Command set: Display a formatted message record**

```
DISPLAY IMSID=R61P,TYPE=RECORD,DRRN=080000EC,
  FORMAT=YES
END
```

Figure 37 on page 199 shows a command set requesting message queue statistics. MAQSAMP(QMR@DSP1) contains a sample command set similar to Figure 37 on page 199.

**Figure 37: Command set: Display message queue statistics**

```
DISPLAY IMSID=R61P,TYPE=STATISTICS
END
```

For more information about building and executing a DISPLAY command set and for information about using the DISPLAY command to create reports, see the *Message Advisor for IMS User Guide.*
DISPLAY TYPE=DESTINATION command set description

The DISPLAY TYPE=DESTINATION command set consists of the following items:

- DISPLAY primary command with associated keywords and parameters
- subcommands with associated keywords and parameters
- END command

Note Keyword and parameter descriptions for the primary command are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

Keyword and parameter descriptions for the subcommands are located in the following chapters:

- “INTERVAL subcommand” on page 293
- “SELECT and REJECT subcommands” on page 313

Note IMS must be active before you can specify the DISPLAY command.

IMSid keyword

IMSid is a required keyword for the DISPLAY TYPE=DESTINATION command.

Table 194 on page 200 lists the IMSid keyword parameter.

Table 194: IMSid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccc</td>
<td>the IMS system to which this DISPLAY TYPE=DESTINATION command set applies</td>
</tr>
<tr>
<td></td>
<td>Note: cccc is a value that you specify.</td>
</tr>
</tbody>
</table>
LIMit keyword

LIMit is an optional keyword for the DISplay TYPE=DESTination command.

Use the LIMit keyword to limit the number of destinations that Message Advisor processes.

Table 195 on page 201 lists the LIMit keyword parameter.

Table 195: LIMit keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnnnnn</td>
<td>the number of destinations to process</td>
</tr>
<tr>
<td></td>
<td>Note: nnnnnnnnn is a value that you specify.</td>
</tr>
</tbody>
</table>

RATE keyword

RATE is an optional keyword for the DISplay TYPE=DESTination command. Use the RATE keyword to control the number of destinations that Message Advisor processes per minute.

Note

The RATE keyword value for the DISplay TYPE=DESTination command temporarily overrides the RATE keyword value for the CUSTOMIZE command.

Table 196 on page 201 lists the RATE keyword parameter.

Table 196: RATE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnnnnn</td>
<td>the number of destinations to process per minute</td>
</tr>
<tr>
<td></td>
<td>Note: nnnnnnnnn is a value that you specify.</td>
</tr>
</tbody>
</table>

SCOpe keyword

SCOpe is an optional keyword for the DISplay TYPE=DESTination command and limits display processing to destinations that match the specified scope. You can specify a list of values for the SCOpe keyword.
The SCOpe keyword applies to shared queues only.

Table 197 on page 202 lists the SCOpe keyword parameters. Figure 38 on page 203 illustrates the queues that are processed for each parameter.

**Table 197: SCOpe keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLQ</td>
<td>includes both local and shared queues in display processing</td>
</tr>
<tr>
<td>LOCALQ</td>
<td>limits display processing to local queues</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can use LCLQ as an alias for LOCALQ.</td>
</tr>
<tr>
<td>OVERFLOWQ</td>
<td>limits display processing to the overflow shared queues</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can use OVFQ as an alias for OVERFLOWQ.</td>
</tr>
<tr>
<td>PRIMARYQ</td>
<td>limits display processing to the primary shared queues</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can use PRIQ as an alias for PRIMARYQ.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHAREDQ</td>
<td>limits display processing to shared queues (both primary and overflow)</td>
</tr>
</tbody>
</table>

**Note:** You can use SHRQ as an alias for SHAREDQ.

SHAREDQ is mutually exclusive with PRIMARYQ and OVERFLOWQ.

---

**Figure 38: SCOpe keyword parameters and associated queues processed**

SCOPE=

- ALLQ
  - SHAREDQ
  - SHRQ
  - PRIMARYQ
  - PRIQ
  - OVERFLOWQ
  - OVFQ
  - LOCALQ
  - LCLQ

---

**TYPE keyword**

TYPE is a required keyword for the **DISplay TYPE=DESTination** command. TYPE indicates the type of information that Message Advisor displays.

**Table 198 on page 204** lists the TYPE keyword parameter.
### Table 198: TYPE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| DESTination | indicates that Message Advisor is to provide a report of destinations with messages  
**Note:** To display information for a specific destination, specify the destination name in a SELECT subcommand. To exclude a specific destination, specify the destination name in a REJECT subcommand. |

## DISPLAY TYPE=DEST_QUEUES command set description

The `DISPLAY TYPE=DEST_QUEUES` command set consists of the following items:

- DISPLAY primary command with associated keywords and parameters
- subcommands with associated keywords and parameters
- END command

**Note**
Keyword and parameter descriptions for the primary command are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

Keyword and parameter descriptions for the subcommands are located in the following chapters:

- “INTERVAL subcommand” on page 293
- “SELECT and REJECT subcommands” on page 313

**Note**
IMS must be active before you can specify the DISplay command.
**IMSid keyword**

IMSid is a required keyword for the `DISPLAY TYPE=DEST_Queues` command.

Table 199 on page 205 lists the IMSid keyword parameter.

**Table 199: IMSid keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cccc</code></td>
<td>the IMS system to which this DISPLAY TYPE=DEST_Queues command set applies&lt;br&gt;Note: <code>cccc</code> is a value that you specify.</td>
</tr>
</tbody>
</table>

**LIMit keyword**

LIMit is an optional keyword for the `DISPLAY TYPE=DEST_Queues` command.

Use the LIMit keyword to limit the number of destinations that Message Advisor processes.

Table 200 on page 205 lists the LIMit keyword parameter.

**Table 200: LIMit keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>nnnnnnnnn</code></td>
<td>the number of destinations to process&lt;br&gt;Note: <code>nnnnnnnnn</code> is a value that you specify.</td>
</tr>
</tbody>
</table>

**LIST_Queues LQ keyword**

LIST_Queues is an optional keyword for the `DISPLAY TYPE=DEST_Queues` command. Use the LIST_Queues keyword to determine the destinations that Message Advisor displays when you include SELECT, REJECT, and INTERVAL subcommands with the `DISPLAY TYPE=DEST_Queues` command set.

**Note**

You can use LQ as an alias for LIST_Queues.

Table 201 on page 206 lists the LIST_Queues keyword parameters.
Table 201: LIST_Queue keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATCH</td>
<td>lists only destinations with queued messages that match all SELECT, REJECT, and INTERVAL subcommand criteria</td>
</tr>
<tr>
<td>ALL</td>
<td>lists all destinations that match the DESTINATION and DESTYPE criteria on SELECT and REJECT subcommands, even if the destinations do not have queued messages</td>
</tr>
<tr>
<td>NOTEMPTY</td>
<td>lists all destinations that have queued messages AND match the DESTINATION and DESTYPE criteria on SELECT and REJECT subcommands, even if none of the queued messages match the other SELECT, REJECT, and INTERVAL subcommand criteria</td>
</tr>
</tbody>
</table>

RATE keyword

RATE is an optional keyword for the DISPLAY TYPE=DEST_QUEUES command. Use the RATE keyword to control the number of destinations that Message Advisor processes per minute.

**Note**
The RATE keyword value for the DISPLAY TYPE=DEST_QUEUES command temporarily overrides the RATE keyword value for the CUSTOMIZE command.

Table 202 on page 206 lists the RATE keyword parameter.

Table 202: RATE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnnnnn</td>
<td>the number of destinations to process per minute</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>nnnnnnnnn is a value that you specify.</td>
</tr>
</tbody>
</table>

SCOpe keyword

SCOpe is an optional keyword for the DISPLAY TYPE=DEST_QUEUES command and limits display processing to destinations that match the specified scope. You can specify a list of values for the SCOpe keyword.

**Note**
The SCOpe keyword applies to shared queues only.
Table 203 on page 207 lists the SCOpe keyword parameters. Figure 38 on page 203 illustrates the queues that are processed for each parameter.

### Table 203: SCOpe keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLQ</td>
<td>includes both local and shared queues in display processing</td>
</tr>
</tbody>
</table>
| LOCALQ    | limits display processing to local queues  
*Note:* You can use LCLQ as an alias for LOCALQ. |
| OVERFLOWQ | limits display processing to the overflow shared queues  
*Note:* You can use OVFQ as an alias for OVERFLOWQ. |
| PRIMARYQ  | limits display processing to the primary shared queues  
*Note:* You can use PRIQ as an alias for PRIMARYQ. |
| SHAREDQ   | limits display processing to shared queues (both primary and overflow)  
*Note:* You can use SHRQ as an alias for SHAREDQ.  
SHAREDQ is mutually exclusive with PRIMARYQ and OVERFLOWQ. |

### Figure 39: SCOpe keyword parameters and associated queues processed

```scope=
ALLQ
  | SHAREDFQ
  | SHRQ
  | PRIMARYQ
  | PRIQ
  | OVERFLOWQ
  | OVFQ
```

**SCOPE=**
SORT_by keyword

SORT_by is an optional keyword for the DISPLAY TYPE=DEST_QUEUES command. Use the SORT_by keyword to determine the criteria by which Message Advisor sorts the displayed information.

Note: If you specify the SORT_by keyword, you must also specify the LIMit keyword.

Table 204 on page 208 lists the SORT_by keyword parameters.

Table 204: SORT_by keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%LGmsg</td>
<td>for non-shared queues, sorts the display output by the percentage of the long message queue data set (LGMSG) used by each destination, in decreasing order</td>
</tr>
<tr>
<td>%SHmsg</td>
<td>sorts the display output by the percentage of the short message queue data set (SHMSG) used by each destination, in decreasing order</td>
</tr>
<tr>
<td>%PRIMary</td>
<td>for shared queues, sorts the display output by the percentage of the long message queue data set (LGMSG) used by each destination, in decreasing order</td>
</tr>
<tr>
<td>%OVERflow</td>
<td>for shared queues, sorts the display output by the percentage of the short message queue data set (SHMSG) used by each destination, in decreasing order</td>
</tr>
<tr>
<td>#MSGS/MESsages</td>
<td>sorts the display output by the number of messages assigned to each destination queue, in decreasing order</td>
</tr>
<tr>
<td></td>
<td>Note: You can use MESsages as an alias for #MSGS.</td>
</tr>
<tr>
<td>#SEGS/SEGments</td>
<td>sorts the display output by the number of segments used by each destination queue, in decreasing order</td>
</tr>
<tr>
<td></td>
<td>Note: You can use SEGments as an alias for #SEGS.</td>
</tr>
</tbody>
</table>
**Parameter** | **Description**
--- | ---
QCT | sorts the display output in a manner similar to the way display output is sorted when you specify #MSGS

When you specify QCT, Message Advisor calculates the number of messages based on information it gathers from IMS counters, instead of information it gathers by accessing message queues. Response time for this parameter should be faster than for the other sort parameters. Because Message Advisor does not access message queues, it cannot compute some values in the display output and leaves them blank.

QCT uses IMS internal destination counters, but changes the way IMS expresses the values. After IMS counts 32,768 (32K) messages, it decreases the message count by increments of one until it reaches zero (64K). After Message Advisor counts 32,768 messages, it increases the message count by increments of one until it reaches 65,535. Like IMS, however, Message Advisor starts over at zero for the 65,536th message.

**Note:** With the other sort parameters, Message Advisor reads IMS message queue data sets to determine the exact number of IMS messages. If the number of messages at each destination is high, response time might take longer than expected.

If you specify QCT, you cannot include the INTERVAL subcommand with the DISPLAY TYPE=DEST_QUEUES command set. You can include SELECT and REJECT subcommands, but you can only specify keywords that apply to destinations, and not keywords that apply to message contents.

---

**TYPE keyword**

TYPE is a required keyword for the `DISPLAY TYPE=DEST_QUEUES` command. TYPE indicates the type of information that Message Advisor displays.

Table 205 on page 209 lists the TYPE keyword parameter.

**Table 205: TYPE keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEST_Queue/DQ</td>
<td>indicates that Message Advisor is to provide a report of destinations with queued messages</td>
</tr>
<tr>
<td><strong>Note:</strong> You can use DQ as an alias for DEST_Queue.</td>
<td></td>
</tr>
</tbody>
</table>
DISPLAY TYPE=MESSAGE command set description

The DISPLAY TYPE=MESSAGE command set consists of the following items:

- DISPLAY primary command with associated keywords and parameters
- subcommands with associated keywords and parameters
- END command

Note
Keyword and parameter descriptions for the primary command are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

Keyword and parameter descriptions for the subcommands are located in the following chapters:

- “INTERVAL subcommand” on page 293
- “SELECT and REJECT subcommands” on page 313

Note
IMS must be active before you can specify the DISPLAY command.

FORMat keyword

FORMat is an optional keyword for the DISPLAY TYPE=MESSAGE command. Use the FORMat keyword to determine how Message Advisor formats the displayed information.

Table 206 on page 210 lists the FORMat keyword parameters.

Table 206: FORMat keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>displays records, including text, in dump format</td>
</tr>
<tr>
<td>Yes</td>
<td>displays and identifies only control block fields</td>
</tr>
</tbody>
</table>
IMSid keyword

IMSid is a required keyword for the DISPLAY TYPE=MESSAGE command.

Table 207 on page 211 lists the IMSid keyword parameter.

Table 207: IMSid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| ccccc     | the IMS system to which this DISPLAY TYPE=MESSAGE command set applies  
            **Note:** ccccc is a value that you specify. |

LIMit keyword

LIMit is an optional keyword for the DISPLAY TYPE=MESSAGE command. Use the LIMit keyword to limit the number of destinations that Message Advisor processes.

Table 208 on page 211 lists the LIMit keyword parameter.

Table 208: LIMit keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| nnnnnnnnn | the number of destinations to process  
            **Note:** nnnnnnnnn is a value that you specify. |

MAXMSGS keyword

MAXMSGS is an optional keyword for the DISPLAY TYPE=MESSAGE command and indicates the maximum number of messages to display.

Table 209 on page 211 lists the MAXMSGS keyword parameter.

Table 209: MAXMSGS keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| 50        | the maximum number of messages to display  
            **Note:** nnnnnnnnn is a value that you specify. |
| nnnnnnnnnn | the maximum number of messages to display  
            **Note:** nnnnnnnnn is a value that you specify. |
RATE keyword

RATE is an optional keyword for the DISPLAY TYPE=MESSAGE command. Use the RATE keyword to control the number of destinations that Message Advisor processes per minute.

Note

The RATE keyword value for the DISPLAY TYPE=MESSAGE command temporarily overrides the RATE keyword value for the CUSTOMIZE command.

Table 210 on page 212 lists the RATE keyword parameter.

Table 210: RATE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnnnnn</td>
<td>the number of destinations to process per minute</td>
</tr>
<tr>
<td></td>
<td>Note: nnnnnnnnn is a value that you specify.</td>
</tr>
</tbody>
</table>

SCOpe keyword

SCOpe is an optional keyword for the DISPLAY TYPE=MESSAGE command and limits display processing to destinations that match the specified scope. You can specify a list of values for the SCOpe keyword.

Note

The SCOpe keyword applies to shared queues only.

Table 211 on page 212 lists the SCOpe keyword parameters. Figure 38 on page 203 illustrates the queues that are processed for each parameter.

Table 211: SCOpe keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLQ</td>
<td>includes both local and shared queues in display processing</td>
</tr>
<tr>
<td>LOCALQ</td>
<td>limits display processing to local queues</td>
</tr>
<tr>
<td></td>
<td>Note: You can use LCLQ as an alias for LOCALQ.</td>
</tr>
<tr>
<td>OVERFLOWQ</td>
<td>limits display processing to the overflow shared queues</td>
</tr>
<tr>
<td></td>
<td>Note: You can use OVFQ as an alias for OVERFLOWQ.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| PRIMARYQ | limits display processing to the primary shared queues  
  *Note:* You can use PRIQ as an alias for PRIMARYQ. |
| SHAREDQ | limits display processing to shared queues (both primary and overflow)  
  *Note:* You can use SHRQ as an alias for SHAREDQ.  
  SHAREDQ is mutually exclusive with PRIMARYQ and OVERFLOWQ. |

**Figure 40: SCOpe keyword parameters and associated queues processed**

**TYPE keyword**

TYPE is a required keyword for the DISPLAY TYPE=MESSAGE command. TYPE indicates the type of information that Message Advisor displays.

Table 212 on page 214 lists the TYPE keyword parameter.
Table 212: TYPE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| MESSAGE   | indicates that Message Advisor is to display the complete text of one or more messages  
  Note: To display a specific message, use a SELECT subcommand with the DRRN keyword. When Message Advisor selects the first segment of the message, it displays all records for that message. |

**DISPLAY TYPE=RECORD command set description**

The **DISPLAY TYPE=RECORD** command set consists of the following items:

- DISPLAY primary command with associated keywords and parameters
- END command

---

**Note**

Keyword and parameter descriptions for the primary command are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

---

IMS must be active before you can specify the DISplay command.

**DRrn keyword**

DRrn is a required keyword for the **DISPLAY TYPE=RECORD** command for non-shared queues. Use the DRrn keyword to display a single message record for a destination.

---

**Note**

The DRrn keyword applies to non-shared queues only.

---

Table 213 on page 215 lists the DRrn keyword parameter.
Table 213: DRrn keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>drrn</td>
<td>the device relative record number (DRRN) that indicates the location of the desired message record within the data set</td>
</tr>
</tbody>
</table>

Note: Use DISplay TYPE=DESTination to determine the DRRN.

ENTRYID keyword

ENTRYID is a required keyword for the DISplay TYPE=RECord command for shared queues. Use the ENTRYID keyword to display a specific shared queues record.

Note: The ENTRYID keyword applies to shared queues only.
You can use the SCOpe keyword to indicate whether Message Advisor is to search the primary or the overflow structure. If SCOpe includes both the primary and the overflow structure, Message Advisor searches the primary structure first. If Message Advisor finds the entry ID, it displays that message. Message Advisor only searches the overflow structure if it does not find the entry ID on the primary structure.

Table 214 on page 215 lists the ENTRYID keyword parameter.

Table 214: ENTRYID keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxxxxxxxx</td>
<td>the specific shared queues record to be displayed</td>
</tr>
</tbody>
</table>

IMSid keyword

IMSid is a required keyword for the DISplay TYPE=RECord command.

Table 215 on page 215 lists the IMSid keyword parameter.

Table 215: IMSid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccc</td>
<td>the IMS system to which this DISplay TYPE=RECord command set applies</td>
</tr>
</tbody>
</table>

Note: cccc is a value that you specify.
**FORMat keyword**

FORMat is an optional keyword for the DISPLAY TYPE=RECORD command. Use the FORMat keyword to determine how Message Advisor formats the displayed information.

Table 216 on page 216 lists the FORMat keyword parameters.

### Table 216: FORMat keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>displays records, including text, in dump format</td>
</tr>
<tr>
<td>Yes</td>
<td>displays and identifies only control block fields</td>
</tr>
</tbody>
</table>

**SCOpe keyword**

SCOpe is an optional keyword for the DISPLAY TYPE=RECORD command and limits display processing to destinations that match the specified scope. You can specify a list of values for the SCOpe keyword.

**Note**
The SCOpe keyword applies to shared queues only.

Table 217 on page 216 lists the SCOpe keyword parameters. Figure 38 on page 203 illustrates the queues that are processed for each parameter.

### Table 217: SCOpe keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLQ</td>
<td>includes both local and shared queues in display processing</td>
</tr>
</tbody>
</table>
| LOCALQ    | limits display processing to local queues  
**Note:** You can use LCLQ as an alias for LOCALQ. |
| OVERFLOWQ | limits display processing to the overflow shared queues  
**Note:** You can use OVFQ as an alias for OVERFLOWQ. |
| PRIMARYQ  | limits display processing to the primary shared queues  
**Note:** You can use PRIQ as an alias for PRIMARYQ. |
Figure 41: SCOpe keyword parameters and associated queues processed

SCOPE=

ALLQ

SHAREDQ
SHRQ

PRIMARYQ
PRIQ

OVERFLOWQ
OVFQ

LOCALQ
LCLQ

Table 218: TYPE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REcord</td>
<td>indicates that Message Advisor is to display a specific message record</td>
</tr>
</tbody>
</table>
DISPLAY TYPE=STATISTICS command set description

The DISPLAY TYPE=STATISTICS command set consists of the following items:

- DISPLAY primary command with associated keywords and parameters
- END command

Note
Keyword and parameter descriptions for the primary command are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value. IMS must be active before you can specify the DISPLAY command.

IMSid keyword

IMSid is a required keyword for the DISPLAY TYPE=STATISTICS command.

Table 219 on page 218 lists the IMSid keyword parameter.

Table 219: IMSid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| cccc      | the IMS system to which this DISPLAY TYPE=STATISTICS command set applies  
            | **Note:** cccc is a value that you specify.             |

TYPE keyword

TYPE is a required keyword for the DISPLAY TYPE=STATISTICS command. TYPE indicates the type of information that Message Advisor displays.

Table 220 on page 219 lists the TYPE keyword parameter.
### Table 220: TYPE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATistics</td>
<td>indicates that Message Advisor is to provide message queue statistics</td>
</tr>
</tbody>
</table>

### DISPLAY TYPE=ANALYZER command set description

The **DISPLAY TYPE=ANALYZER** command set consists of the following items:

- **Note**
  
  The **DISPLAY TYPE=ANALYZER** command set applies to shared queues only.

- DISPLAY primary command with associated keywords and parameters
- END command

**Note**

Keyword and parameter descriptions for the primary command are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

IMS must be active before you can specify the DISplay command.

### IMSid keyword

IMSid is a required keyword for the **DISPLAY TYPE=ANALYZER** command.

Table 221 on page 219 lists the IMSid keyword parameter.

### Table 221: IMSid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccc</td>
<td>the IMS system to which this <strong>DISPLAY TYPE=ANALYZER</strong> command set applies</td>
</tr>
</tbody>
</table>

**Note:** cccc is a value that you specify.
**TYPE keyword**

TYPE is a required keyword for the `DISPLAY TYPE=ANALYZER` command. TYPE indicates the type of information that Message Advisor displays.

Table 222 on page 220 lists the TYPE keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALyzer</td>
<td>indicates that Message Advisor is to provide data from the shared queues Analyzer Facility</td>
</tr>
</tbody>
</table>
IMSCMD and PLEXCMD commands

This chapter describes the Message Advisor for IMS IMSCMD and PLEXCMD command sets, and also provides syntax diagrams and sample command sets.

For more information about building and executing an IMSCMD command set, see the Message Advisor for IMS User Guide.

Overview

Message Advisor lets you submit IMS commands through the Message Advisor Interactive System Productivity Facility (ISPF) and batch interfaces. Message Advisor passes the command to IMS, and the IMS output is passed back to Message Advisor. You can only submit one IMS command per IMSCMD command set. The Message Advisor ISPF interface provides options that allow you to save and/or print the output returned from IMS.

For more information about submitting IMS commands through Message Advisor, see the Message Advisor for IMS User Guide.

Message Advisor lets you submit IMSPL EX commands through the IMSPL EX Operations Manager interface, via the Message Advisor ISPF command interface or through a batch server running Message Advisor. Message Advisor routes the command through the IMSPL EX subsystem that is specified on the PLEXCMD command, and then collects and returns the output for display. You can only submit one command per PLEXCMD command set.

Note

The IMSPL EX facility was introduced with IMS 8.1. The target IMSPL EX must be defined and operational before Message Advisor can communicate with it.

IMSCMD command set syntax

The following figure shows the IMSCMD command set syntax.
Sample IMSCMD command set

The following figure shows a sample command set for using Message Advisor to execute a /DIS A command to IMS system R61P.

**Figure 43: Command set: Execute an IMS command**

```
IMSCMD IMSID=R61P,CMD="/DIS A"
END
```

For more information about submitting IMS commands through Message Advisor, see the *Message Advisor for IMS User Guide*.

IMSCMD command set description

The IMSCMD command set consists of the following items:

- IMSCMD primary command with associated keywords and parameters
- the IMS command to be submitted
- END command

**Note**

Keyword and parameter descriptions are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

CMD keyword

CMD is a required keyword for the IMSCmd command.

Table 223 on page 223 lists the CMD keyword parameter.
Table 223: CMD keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| /command  | the command that Message Advisor submits to the IMS command processor. The value must be a valid IMS command, no longer than 128 bytes and enclosed in quotation marks. If the command contains single quotation marks, you can:  
  - enclose it in double quotation marks  
  - use any special character as a delimiter, preceded by C |

**IMSid keyword**

IMSid is a required keyword for the IMSCmd command. Table 224 on page 223 lists the IMSid keyword parameter.

Table 224: IMSid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccccc</td>
<td>the IMS system to which this IMSCmd command set applies. <strong>Note:</strong> ccccc is a value that you specify.</td>
</tr>
</tbody>
</table>

**PLEXCMD command set syntax**

The following figure shows the PLEXCMD command set syntax.

*Figure 44: PLEXCMD command set syntax*

```
PLEXCMD  PLEXid=cccc  .CMD="/command"  
"ROUTE=cccc,cccc,...cccc"  ,WAIT=nnnnnnn
END
```
Sample PLEXCMD command set

The following figure shows sample PLEXCMD command sets.

**Figure 45: PLEXCMD command set: Execute an IMSPLEX command**

```
PLEXCMD PLEXID=PROD1,WAIT=30,
   CMD='QRY MEMBER SHOW(ALL) TYPE(IMS)'
END
PLEXCMD CMD='QRY IMSPLEX',PLEX=PROD2
END
PLEXCMD CMD='QUERY TRAN NAME(PART)',PLEXID=PLEX3
END
PLEXCMD CMD='/DIS OLDS',PLEXID=IMSP1,ROUTE='IMS1,IMS2'
END
PLEXCMD CMD='/DIS NODE MYNODE1',PLEX=IMSP1
END
```

**PLEXCMD command set description**

The PLEXCMD command set consists of the following items:

- PLEXCMD primary command with associated keywords and parameters
- the IMS or Operations Manager command to be submitted
- END command

**Note**

Keyword and parameter descriptions are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

**CMD keyword**

CMD is a required keyword for the PLEXCMD command.

Table 225 on page 225 lists the CMD keyword parameter.
Table 225: CMD keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'command'</td>
<td>the value must be a valid IMS or Operations Manager command, no longer than 128 characters and enclosed in quotation marks. If the command contains single quotation marks, you can: ■ enclose it in double quotation marks ■ use any special character as a delimiter, preceded by C</td>
</tr>
</tbody>
</table>

PLEXid keyword

PLEXid is a required keyword for the PLEXCMD command.

Table 226 on page 225 lists the PLEXid keyword parameter.

Table 226: PLEXid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccccc</td>
<td>the five-character IMSPLEX name that is the target of this PLEXCMD command. <strong>Note:</strong> ccccc is a value that you specify.</td>
</tr>
</tbody>
</table>

ROUTE keyword

ROUTE is an optional keyword for the PLEXCMD command.

Table 227 on page 225 lists the ROUTe keyword parameter.

Table 227: ROUTe keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cc...cc</td>
<td>one or more IMSIDs (separated by commas) within the IMSPLEX that are to receive the PLEXCMD. If you specify the ROUTe keyword, the command will only be routed to the specified systems for processing. If you do not specify the keyword, all available systems in the IMSPLEX will process the PLEXCMD. <strong>Note:</strong> cc...cc is a value that you specify.</td>
</tr>
</tbody>
</table>
WAIT keyword

WAIT is an optional keyword for the PLEXCMD command.

Table 228 on page 226 lists the WAIT keyword parameter.

**Table 228: WAIT keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>the number of seconds for which each IMS system in the IMSPLEX is allowed to process the command before the Operations Manager abandons waiting for the command to complete. The return code and reason code will indicate that the command was still processing after the specified time and the complete command response was not available.</td>
</tr>
<tr>
<td>nnnnnnnnn</td>
<td>Note: <em>nnnnnnnn</em> is a value that you specify.</td>
</tr>
</tbody>
</table>
REQUEUE command

This chapter describes the Message Advisor for IMS REQUEUE command set, and also provides a syntax diagram and sample command sets. For more information about building and executing a REQUEUE command set, see the Message Advisor for IMS User Guide.

Overview

The REQUEUE command lets you requeue lost messages, and also requeue messages in a variety of other situations. For example, you can use the REQUEUE command with the UNLOAD and DEQUEUE commands to unload messages queued to a specific logical terminal (LTERM) destination, and then requeue the messages to the same or another LTERM. The ability to include SELECT and REJECT subcommands with REQUEUE command sets provides for greater flexibility when requeueing messages.

Message Advisor includes processing modes that allow you to perform the following tasks:

- requeue messages to active queues
- extract selected messages to a data set for later requeue
- validate (test) that a requeue will be successful and count the messages that will be requeued

The REQUEUE command set consists of the REQUEUE primary command and its associated keywords and parameters, with optional subcommands and their associated keywords and parameters. The REQUEUE subcommands are as follows:

- CHANGE subcommand applies changes to selected messages
  For more information, see “CHANGE subcommand” on page 254.
EXTRACT subcommand identifies the data set to which Message Advisor writes extracted messages
For more information, see “OUTPUT, EXTRACT, and SCRAP subcommands” on page 303.

GROUP subcommand defines message groups for the SELECT report
For more information, see “GROUP subcommand” on page 270.

INPUT subcommand identifies a data set to be used as input to the requeue
For more information, see “INPUT subcommand” on page 272.

INTERVAL subcommand requeues messages according to a specified time range
For more information, see “INTERVAL subcommand” on page 293.

REJECT subcommand specifies criteria to exclude specific destinations and/or messages from being requeued
For more information, see “SELECT and REJECT subcommands” on page 313.

SCRAP subcommand identifies the data set to which Message Advisor writes messages when an error condition or a filter prevents successful requeue
For more information, see “OUTPUT, EXTRACT, and SCRAP subcommands” on page 303.

SELECT subcommand specifies criteria to select specific destinations and/or messages to requeue
For more information, see “SELECT and REJECT subcommands” on page 313.

SPILL subcommand allocates temporary spill data sets
For more information, see “SPILL subcommand” on page 275.

---

Requeueing from one system to another

Message Advisor allows you to requeue messages from one system to another for testing or benchmarking purposes.

If Message Advisor detects nonexistent destinations while requeueing messages that originated on a different production system, it may perform any of several actions, including creating virtual destinations or scrapping messages. If the same destination represents a different application or device on the two systems, requeueing may cause problems for the application or device.
If you wish to requeue from one system to another in an MSC environment where the SYSID specifications on the systems are different, BMC Software recommends that you include the following statement in the REQUEUE command set:

```
CHANGE NEWSYSID=(ASSIGN,ASSIGN)
```

This statement will change the SYSID on all requeued messages to appropriate values for the destination system.

### Input for requeues

You can use any of the following input sources to requeue an IMS message:

- online log data set (OLDS)
- system log data set (SLDS)
- data unloaded from the message queue
- data extracted from the OLDS or SLDS

When you requeue messages, Message Advisor performs the following activities:

- automatically selects the checkpoint to use for the requeue (optional)
- allows you to override the selected checkpoint
- dynamically allocates log data set names obtained from RECON data sets
- allows you to override the log data set names by specifying an input data set name, or by specifying the UNIT and VOLSER of an uncataloged SLDS
- selects or rejects messages based on the following criteria:
  - date and time ranges
  - destinations (with masking allowed)
  - origin (with masking allowed)
  - destination type
  - other message prefix or message content selection criteria
- enables you to requeue a specific DUMPQ, PURGE, or SNAPQ checkpoint
- re-establishes conversations (optional)
- requests that remote system messages be requeued (optional)
enables you to modify messages as they are requeued (optional)

Note

For REQUEUE TYPE=COLD, REQUEUE TYPE=EREFAIL, and REQUEUE TYPE=REPROCESS, SLDS input may be supplied via job control language (JCL). To supply the SLDS via JCL, add a //SLDS dd statement that describes the data set to use as requeue input in the Message Advisor Server JCL.

Support for uncataloged SLDS

When performing a requeue, Message Advisor searches the catalog for the SLDS name specified in the RECON data sets. If the data set name is not found, dynamic allocation fails and Message Advisor attempts to find the data set using the UNIT and VOLSER information specified in the RECON data sets.

You can override the RECON data sets with an INPUT subcommand that specifies the data set name, UNIT, and VOLSER for an uncataloged SLDS.

To permanently use an uncataloged SLDS for executing Message Advisor requeues, verify that the RECON control record specifies the proper device type for the TAPEUNIT and/or DASDUNIT parameters. If not, use the CHANGE RECON command to make the appropriate corrections for future PRISLDS records. You can also use the CHANGE.PRILOG SLDS command to make any necessary corrections to existing PRISLDS records you want to use for requeue.

To temporarily use an uncataloged SLDS for executing Message Advisor requeues because of incorrect information in the RECON data sets, specify the data set name, UNIT, and VOLSER of the SLDS in an INPUT subcommand. You can use one UNIT keyword and one or more VOLSER keywords in the subcommand.

User exit

The REQUEUE command provides a user exit for overriding selection criteria. Several sample exits are provided as examples. See members QMREXIT0, QMREXIT1, QMREXIT3, and QMREXIT9 in MAQSAMP.

INCORE keyword and REQUEUE performance

To achieve optimal performance when requeueing messages, you must specify the appropriate INCORE value in the REQUEUE command set. The default,
INCORE=YES, directs Message Advisor to store log records in extended private memory instead of in spill files—as long as adequate memory is available in the Message Advisor address space. Storing log records in extended private memory avoids lengthy input/output (I/O) processing during the requeue activity, and can reduce the requeue time.

When Message Advisor runs in a memory-constrained environment, INCORE=YES may cause it to run slower than it would run with INCORE=NO. The slowdown can occur when the Multiple Virtual Storage (MVS) system pages heavily, or when Message Advisor executes in an MVS performance group that severely limits access to real memory. The slowdown occurs because Message Advisor’s demand for virtual storage using INCORE=YES causes a higher paging rate than the spill file I/O using INCORE=NO.

To avoid excessive paging, Message Advisor must have free access to real memory resources. One way to guarantee adequate resources is to run Message Advisor in an MVS performance group that specifies an unlimited working set size. Alternatively, executing a REQUEUE command with INCORE=FIXED allows Message Advisor to page-fix most of the needed storage. With INCORE=FIXED, the input checkpoint size determines the amount of real memory that Message Advisor page-fixes. For a large input checkpoint, Message Advisor page-fixes a correspondingly large amount of memory. Running Message Advisor with INCORE=FIXED against a large input checkpoint can significantly impact other system processing.

BMC Software recommends using INCORE=YES in most situations. When INCORE=YES, Message Advisor issues a single warning message to the MVS console if it detects an excessive paging rate. If the current request has a critical priority, you can reset Message Advisor to run in an MVS performance group with an unlimited working set size. If you know that a REQUEUE request has critical priority and your site can tolerate the impact of page-fixing, submit the request with INCORE=FIXED.

For more information about building and executing a REQUEUE command set, see the Message Advisor for IMS User Guide.

Assigning conditional steps

You can exercise additional control over a Message Advisor request that contains multiple command sets by using one or more COND pseudo command sets to insert conditional steps. Conditional steps let you perform the following tasks:

- request that Message Advisor execute a command set only if it successfully executes a previous command set

—OR—

- request that Message Advisor execute a command set only if it does not successfully execute a previous command set
**Note**
A COND step can (as an option) include the LABEL, IFCOND, MAXRC, and BRANCH keywords.

When you insert a COND step, a return code of 8 or higher in any command set no longer results in the remaining command sets being flushed with a BMC43111W message. Instead, Message Advisor executes all command sets (unless the COND step successfully executes a BRANCH keyword).

For more information about building and executing a COND pseudo command, see “COND pseudo command” on page 399, and see the *Message Advisor for IMS User Guide*.

**REQUEUE command set syntax**

The following figures show the REQUEUE command set syntax.
Figure 46: REQUEUE command set syntax (part 1 of 2)
Sample REQUEUE command sets

Message Advisor provides sample REQUEUE command sets that can serve as templates when building your own REQUEUE command sets.
The MAQSAMP library contains the sample command sets.

Table 229 on page 235 identifies the library members that contain the sample REQUEUE command sets. MAQSAMP also contains an index in member QMR@INDX that lists all sample command sets provided with Message Advisor.

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMR@REQ1</td>
<td>requeues all messages — COLD start</td>
</tr>
<tr>
<td>QMR@REQ2</td>
<td>requeues all transaction messages — COLD start</td>
</tr>
<tr>
<td>QMR@REQ3</td>
<td>requeues messages to an LTERM — COLD start</td>
</tr>
<tr>
<td>QMR@REQ4</td>
<td>requeues messages — /ERE failure</td>
</tr>
<tr>
<td>QMR@REQ5</td>
<td>requeues messages with changes — reprocess</td>
</tr>
</tbody>
</table>

Note
Many of the samples include field-level prompts (IMSID=?). When you execute a command set through the Message Advisor Interactive System Productivity Facility (ISPF) interface, you are prompted to provide a valid value for any field that contains a prompt.
If you execute a command set outside of the Message Advisor ISPF interface, you are not prompted to replace ?. You must replace all ? with a valid value before submitting the request.

The following sample command sets are similar to the samples provided in MAQSAMP.

Figure 48 on page 236 shows a command set for requeueing all messages following a normal IMS shutdown and cold start.

For the following explanation of the command set shown in Figure 48 on page 236, assume that Message Advisor recorded the latest checkpoint in the Message Advisor Checkpoint Tracking data set. The checkpoint tracking information and the RECON data set information allow Message Advisor to automatically locate the SLDS to use as input. Based on the command set criteria, Message Advisor automatically selects the checkpoint and input file (SLDS).

In the command set, TYPE=COLD indicates that the requeue follows a normal cold start. Message Advisor does not requeue conversational messages (CONVERSATIONS=NONE). Message Advisor writes any message it cannot requeue to the scrap file (SCRAP=YES), and uses the SLDS instead of the OLDS (OLDS=NO). The messages remain in storage while Message Advisor processes them (INCORE=YES).
The command set requests a summary report (REPORT=SUMMARY), and requests that Message Advisor insert messages to IMS as fast as possible (RATE=0).

**Figure 48: Command set: Requeue all messages**

```
REQUEUE IMSID=R61P,MODE=REQUEUE,TYPE=COLD,CONVERSATIONS=NONE,
SCRAP=YES,OLDS=NO,INCORE=YES,REPORT=SUMMARY,RATE=0
END
```

**Note**

Except for the IMSID and REPORT keywords, Figure 48 on page 236 shows default keyword values. The keywords are included for clarity only and are not required.

**Figure 49 on page 236** shows a command set for requeueing all transaction messages following a normal IMS shutdown and cold start. The command set is similar to the sample command set provided in MAQSAMP(QMR@REQ2).

**Figure 49: Command set: Requeue transaction messages**

```
REQUEUE IMSID=R61P,CONVERSATIONS=ALL
SELECT MSGTYPE=TRANSACTIONS
END
```

**Figure 50 on page 236** shows a command set for requeueing messages destined to an LTERM. The command set is similar to the sample command set provided in MAQSAMP(QMR@REQ3). The requeue follows a normal IMS shutdown and cold start. The command set includes an INTERVAL subcommand that specifies the time stamp range for which Message Advisor is to select messages, and a SELECT subcommand that specifies the LTERM for which Message Advisor is to select messages.

The SELECT subcommand limits the requeue to messages destined for LTRM2900. The INTERVAL subcommand limits the requeue to messages queued between 8 A.M. and 5 P.M. on Julian date 95172. The command set also requests a summary report and a detailed destination report (REPORT=(SUMMARY,DESTINATION)).

**Figure 50: Command set: Requeue messages to a specific LTERM**

```
REQUEUE IMSID=R61P,REPORT=(SUMMARY,DESTINATION)
INTERVAL START=95172/0800000,STOP=95172/1700000
SELECT DESTINATION=LTRM2900,QUEUE=ALL,DESTYPE=CNT
END
```

**Figure 51 on page 237** shows a command set for requeueing all messages following an abnormal IMS shutdown, /ERE failure, and cold start. The command set is similar to the sample command set provided in MAQSAMP(QMR@REQ4).

In **Figure 51 on page 237**, TYPE=EREFAIL indicates that the requeue follows a cold start that was performed because of an unsuccessful emergency restart.
command set requests a summary report and a detailed destination report (REPORT=(SUMM,DEST)).

Figure 51: Command set: Requeue all messages following /ERE failure

REQUEUE IMSID=R61P,TYPE=EREFAIL,REPORT=(SUMM,DEST)
END

Figure 52 on page 237 shows a command set for requeueing all messages using a specific checkpoint. The requeue follows a cold start.

For the command set shown in Figure 52 on page 237, Message Advisor uses the checkpoint specified (CHKPT=2001330/080901) instead of determining the checkpoint. Because the command set does not include an INPUT subcommand, Message Advisor determines the SLDS to use for the requeue.

Figure 52: Command set: Requeue all messages using a specific checkpoint

REQUEUE IMSID=R61P,MODE=REQUEUE,TYPE=COLD,CONVERSATIONS=NONE,
CHKPT=2001330/080901,SCRAP=YES,OLDS=NO,INCORE=YES,
REPORT=ALL,RATE=0
END

Figure 53 on page 237 shows a command set for requeueing all messages using a specific input file. The requeue follows a cold start.

For the command set shown in Figure 53 on page 237, Message Advisor searches the data set specified in the INPUT subcommand to locate the requeue checkpoint. If you want Message Advisor to search multiple SLDS, you can include multiple INPUT subcommands with the REQUEUE command set.

Figure 53: Command set: Requeue all messages using a specific input file

REQUEUE IMSID=R61P,MODE=REQUEUE,TYPE=COLD,CONVERSATIONS=NONE,SCRAP=YES,
OLDS=NO,INCORE=YES,REPORT=ALL,RATE=0
INPUT DSN=BMC.R61P.SLDS.D95330.T0807021
END

For more information about building and executing a REQUEUE command set, see the Message Advisor for IMS User Guide.

REQUEUE command set description

The REQUEUE command set consists of the following items:

- REQUEUE primary command with associated keywords and parameters
- subcommands with associated keywords and parameters
- END command
Keyword and parameter descriptions for the subcommands are available at the following locations:

- “CHANGE subcommand” on page 254
- “OUTPUT, EXTRACT, and SCRAP subcommands” on page 303
- “GROUP subcommand” on page 270
- “INPUT subcommand” on page 272
- “INTERVAL subcommand” on page 293
- “SELECT and REJECT subcommands” on page 313
- “SPILL subcommand” on page 275

**CHKPT keyword**

CHKPT is an optional keyword for the REQueue command and indicates the checkpoint from which Message Advisor begins the requeue. If you do not specify the CHKPT keyword, Message Advisor determines the checkpoint from the Checkpoint Tracking data set. If you specify COLD for the TYPE keyword value, Message Advisor determines the valid checkpoint from the Checkpoint Tracking data set, and then uses the RECON data set to locate the SLDS that contains the checkpoint.

**Note**
The CHKPT keyword is not allowed with TYPE=FILE.

Table 230 on page 239 lists the CHKPT keyword parameter.
### Table 230: CHKPT keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td>the time stamp of the checkpoint to use for the requeue. The time stamp must be exact to the second. Tenths of a second are not required and are ignored if specified. <strong>Note:</strong> <code>timestamp</code> is a value that you specify. See “INTERVAL subcommand” on page 293, for a discussion of time stamp syntax.</td>
</tr>
</tbody>
</table>

### COMpress keyword

COMpress is an optional keyword for the REQueue command and indicates whether to compress records.

Table 231 on page 239 lists the COMpress keyword parameters.

### Table 231: COMpress keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>compresses records so that more work data can be handled in main storage</td>
</tr>
<tr>
<td>No</td>
<td>does not compress records</td>
</tr>
</tbody>
</table>

### CONVersations keyword

CONVersations is an optional keyword for the REQueue command and indicates the type of messages that Message Advisor requeues.

The following considerations and restrictions apply when requeuing conversational messages:

- Only one transaction can be on the queue at a time for a given conversation ID for a given user or LTERM.

- It is not possible to "queue up" multiple instances of a conversational transaction to be executed on behalf of a user or an LTERM.

- You can execute a `REQueue TYPE=COLD, MDode=EXtract` from the DUMPQ to an extract file, and then execute a `REQueue TYPE=FILE` to requeue active/scheduled/ held conversations.
For REQueue TYPE=EREfail, Message Advisor will establish conversations in the same state as they existed at the Abend point after the selected checkpoint, unless CONV=None is specified.

For REQueue TYPE=FILE, Message Advisor will establish conversations in the same state as they existed when the conversations were Unloaded or Extracted unless CONV=None is specified. Because IMS allows only one PENDING or SCHEDULED conversation for a given LTERM/USER at one time, Message Advisor may be unable to Requeue and establish some conversations without manual intervention to /EXIT or /HOLD any existing conversational work for a given LTERM/USER.

Table 232 on page 240 lists the CONVersations keyword parameters.

Table 232: CONVersations keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Message Advisor requeues only nonconversational messages</td>
</tr>
<tr>
<td>ACT</td>
<td>Message Advisor requeues active and scheduled conversational and nonconversational messages</td>
</tr>
<tr>
<td>All</td>
<td>Message Advisor requeues all conversational and nonconversational messages</td>
</tr>
<tr>
<td>HELD</td>
<td>Message Advisor requeues all held conversational and nonconversational messages</td>
</tr>
<tr>
<td></td>
<td>Held conversation destinations must be a combination of the user ID and conversation number. For example, the keywords DESTYPE=HELD,DEST=FRED*,CONVID=(0001,000F, 23,CONV0003,CONV01DF) would select conversations with numbers of 0001, 000F, 0023, 0003, or 01DF for any user ID starting with FRED. See the CONVID keyword description in SELECT and REJECT subcommands on page 313 for more information.</td>
</tr>
<tr>
<td>ONLY</td>
<td>Message Advisor requeues only conversational messages</td>
</tr>
<tr>
<td>ONLY_ACT</td>
<td>Message Advisor requeues only active and scheduled conversational messages</td>
</tr>
<tr>
<td>ONLY_HELD</td>
<td>Message Advisor requeues only held conversational messages</td>
</tr>
</tbody>
</table>

**DISCard keyword**

DISCard is an optional keyword for the REQueue command and indicates whether Message Advisor requeues the messages that IMS does not retain following a warm start.

The messages that IMS does not retain include nonrecoverable messages and discardable Advanced Program-to-Program Communication (APPC) and certain open transaction manager access (OTMA) messages.
Table 233 on page 241 lists the DISCard keyword parameters.

### Table 233: DISCard keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Message Advisor discards the messages that IMS does not retain following a warm start</td>
</tr>
<tr>
<td>No</td>
<td>Message Advisor requeues the messages that IMS does not retain following a warm start</td>
</tr>
</tbody>
</table>

### ERROR keyword

ERROR is an optional keyword for the REQueue command and indicates which scrap codes Message Advisor considers to be error conditions.

Consider the following when specifying the ERROR keyword:

- You can specify NOOne or a list of values.
- You can use either scrap codes or the names assigned to the scrap codes. For example, if scrap codes 1, 3, and 5 are to be considered error conditions, you can specify ERROR=(1,3,5) or ERROR=(NOSOURCE,INSERT,LENGTH). The following table lists the valid scrap codes and names:

<table>
<thead>
<tr>
<th>Scrap Code</th>
<th>Name</th>
<th>Scrap Code</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 ^a</td>
<td>NOSOURCE</td>
<td>11 ^a</td>
<td>CONVCCB</td>
</tr>
<tr>
<td>02 ^a</td>
<td>NODEST</td>
<td>12</td>
<td>RESTRICT</td>
</tr>
<tr>
<td>03 ^a</td>
<td>INSERT</td>
<td>13</td>
<td>NOAPPC</td>
</tr>
<tr>
<td>04 ^a</td>
<td>ENQFAIL</td>
<td>14</td>
<td>ZERO TIME</td>
</tr>
<tr>
<td>05 ^a</td>
<td>LENGTH</td>
<td>15</td>
<td>RECORD</td>
</tr>
<tr>
<td>06 ^a</td>
<td>PREFIX</td>
<td>16</td>
<td>BADQUEUE</td>
</tr>
<tr>
<td>07 ^a</td>
<td>QSTOP</td>
<td>17</td>
<td>NODATA</td>
</tr>
<tr>
<td>08</td>
<td>USER</td>
<td>18</td>
<td>NOSYsid</td>
</tr>
<tr>
<td>09</td>
<td>CONVIN</td>
<td>19</td>
<td>NOOTMA</td>
</tr>
<tr>
<td>0A</td>
<td>CONVOUT</td>
<td>1A ^a</td>
<td>BADSYS</td>
</tr>
<tr>
<td>0B ^a</td>
<td>CHANGE</td>
<td>1B</td>
<td>ERECAN</td>
</tr>
<tr>
<td>0C ^a</td>
<td>CANCEL</td>
<td>1C</td>
<td>ERENOENQ</td>
</tr>
<tr>
<td>0D</td>
<td>NORECOVER</td>
<td>1D</td>
<td>EREDEQ</td>
</tr>
</tbody>
</table>
If the ERROR keyword is not used, these scrap codes are the default error conditions.

- You can negate scrap codes by preceding them with ‘¬’. For example, to consider all scrap codes except 1 and 2 to be errors, you can specify ERROR=(¬ 1,¬2).

**Note**

If Message Advisor scraps messages because of an ERROR scrap code, it sets the return code from the REQueue command to 4.

Table 234 on page 242 lists the ERROR keyword parameters.

### Table 234: ERROR keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>The scrap codes that identify messages that cannot be requeued because of an error are considered to be error conditions. ERROR=ERROR is equivalent to ERROR=(NOSOURCE,NODEST,INSERT,ENQFAIL,LENGTH,PREFIX,QSTOP,CANCEL,CONVCCB,NODEID,NOAPPCC,RECORD,BADQUEUE,NODATA,NOSYSID,NOOTMA,BADSYS). You can include other scrap codes or names with the default ERROR scrap codes. For example, to scrump messages with zero time stamps in addition to the default ERROR codes, you can specify ERROR=(ERROR,ZEROTIME). You can also subtract from the default ERROR scrap codes by inserting ¬ in front of the applicable code or name. For example, to exclude messages that are too long from the default scrap codes, you can specify ERROR=(ERROR,¬LENGTH).</td>
</tr>
<tr>
<td>nn</td>
<td>Message Advisor considers the scrap codes you specify to be error conditions. You can use scrap code names in place of the scrap codes.</td>
</tr>
<tr>
<td>NOne</td>
<td>Message Advisor does not consider any scrap codes to be error conditions</td>
</tr>
</tbody>
</table>

**EXIT keyword**

EXIT is an optional keyword for the REQueue command and identifies a user exit that allows for additional selection and rejection of messages for requeue.

Table 235 on page 243 lists the EXIT keyword parameter.
Table 235: EXIT keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccccccccc</td>
<td>the name of a user exit module that allows further selection and rejection of messages for requeue</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The module must reside in an accessible library.</td>
</tr>
</tbody>
</table>

**IMSid keyword**

IMSid is a required keyword for the REQueue command.

Table 236 on page 243 lists the IMSid keyword parameter.

Table 236: IMSid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccc</td>
<td>the IMS system to which this REQueue command set applies</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> cccc is a value that you specify.</td>
</tr>
</tbody>
</table>

**INCore keyword**

INCore is an optional keyword for the REQueue command and indicates whether Message Advisor stores records in main storage or in spill files when it requeues messages.

See “INCORE keyword and REQUEUE performance” on page 230 for additional information.

Table 237 on page 243 lists the INCore keyword parameters.

Table 237: INCore keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed</td>
<td>Message Advisor page-fixes main storage for the requeue</td>
</tr>
<tr>
<td>No</td>
<td>Message Advisor stores log records in spill files instead of in extended private memory</td>
</tr>
</tbody>
</table>
**INTERNAL_PREFIX keyword**

INTERNAL_PREFIX is an optional keyword for the REQueue command and indicates whether Message Advisor retains internal prefixes that may have been added to messages via an IMS exit.

Table 238 on page 244 lists the INTERNAL_PREFIX keyword parameters.

**Table 238: INTERNAL_PREFIX keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEEP</td>
<td>Message Advisor retains internal prefixes when it requeues messages</td>
</tr>
<tr>
<td>DISCARD</td>
<td>Message Advisor does not retain internal prefixes when it requeues messages</td>
</tr>
</tbody>
</table>

**MAXDATA keyword**

MAXDATA is an optional keyword for the REQueue command and indicates the number of bytes of message data (per message) to include in the ERRORDATA, SCRAPDATA, and MSGDATA reports.

Note: Message Advisor always includes the entire message prefix. The MAXDATA keyword limits the amount of data that follows the prefix.

Table 239 on page 244 lists the MAXDATA keyword parameter.

**Table 239: MAXDATA keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>the number of bytes of message data (per message) to include in the ERRORDATA, SCRAPDATA, and MSGDATA reports</td>
</tr>
<tr>
<td>nnnnnnnnn</td>
<td>Note: nnnnnnnnn is a value that you specify.</td>
</tr>
</tbody>
</table>
**MAXMSGS keyword**

MAXMSGS is an optional keyword for the REQueue command and indicates the maximum number of messages to include in the ERRORDATA, SCRAPDATA, and MSGDATA reports.

Table 240 on page 245 lists the MAXMSGS keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 nnnnnnnn</td>
<td>the maximum number of messages to include in the ERRORDATA, SCRAPDATA, and MSGDATA reports</td>
</tr>
<tr>
<td>Note: nnnnnnnn is a value that you specify.</td>
<td></td>
</tr>
</tbody>
</table>

**MOde keyword**

MOde is an optional keyword for the REQueue command and indicates what Message Advisor is to do with the messages it selects for requeue.

You can specify a list of values for the MOde keyword.

Table 241 on page 245 lists the MOde keyword parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXtract</td>
<td>Message Advisor copies messages to an extract file, which you can use later for a TYPE=FILE requeue</td>
</tr>
<tr>
<td>Note: When you specify EXtract without any of the other MOde keyword parameters, Message Advisor extracts messages as they exist on the input log or data set and applies any changes resulting from CHANGE subcommands. Message Advisor does not perform validation, so it may not be possible to requeue all extracted messages.</td>
<td></td>
</tr>
<tr>
<td>REQueue</td>
<td>Message Advisor requeues messages to the IMS system</td>
</tr>
<tr>
<td>Note: IMS must be active before you can specify REQueue.</td>
<td></td>
</tr>
</tbody>
</table>
**OLDS keyword**

OLDS is an optional keyword for the REQueue command and indicates the conditions under which Message Advisor uses the OLDS as input for requeues.

Table 242 on page 246 lists the OLDS keyword parameters.

**Table 242: OLDS keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Message Advisor uses the OLDS only if an archived SLDS is not available</td>
</tr>
<tr>
<td>NEver</td>
<td>Message Advisor never uses the OLDS as input for requeues, even if no archived SLDS is available</td>
</tr>
<tr>
<td>Yes</td>
<td>Message Advisor uses the OLDS as input for requeues, regardless of whether an archive to SLDS has completed</td>
</tr>
</tbody>
</table>

*Note: Yes requires TYPE=COLD.*

**RATE keyword**

RATE is an optional keyword for the REQueue command and determines the number of messages that Message Advisor processes per minute.

*Note*

The RATE keyword value for the REQueue command temporarily overrides the RATE keyword value for the CUSTOMIZE command.

Table 243 on page 247 lists the RATE keyword parameter.
Table 243: RATE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| nnnnnnnnn | the number of messages that Message Advisor processes per minute  
**Note:** One message per second is the slowest rate that can be achieved with RATE values 1-60. |

**REPORT/PRint keyword**

REPORT is an optional keyword for the REQueue command and indicates the type of information Message Advisor provides regarding the requeue activity.

You can specify ALL, None, or a list of values other than ALL or None for the REPORT keyword.

You can use PRint as an alias for REPORT.

Table 244 on page 247 lists the REPORT keyword parameters.

Table 244: REPORT keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| ALL           | Message Advisor includes the following reports: Summary, Destination, Origin, Conversations, and Error  
Message Advisor does not include any other reports. You must specify any other reports individually. |
| CONVersations | Message Advisor reports on requeued conversational messages                  |
| DESTination   | Message Advisor reports on the number of messages requeued for each destination |
| ENVIRONMENT   | Message Advisor reports on environmental statistics to aid diagnosis of issues with a REQUEUE request |
| ERROR         | Message Advisor reports on destinations and messages for which it could not requeue messages because of an error condition |
| ERRORDATA     | Message Advisor displays messages that resulted in an error  
Use the MAXMSGS and MAXDATA keywords to limit the number of messages and the amount of data that Message Advisor displays. |
| MESSAGE/MSG   | Message Advisor reports on messages it successfully requeued or extracted  
Message Advisor applies any CHANGE subcommands, and then requeues or extracts the messages before including the message data in the report.  
**Note:** You can use MSG as an alias for MESSAGE. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MESSAGEDATA/ MSGDATA</td>
<td>Message Advisor displays messages that it successfully requeued or extracted Use the MAXMSGS and MAXDATA keywords to limit the number of messages and the amount of data that Message Advisor displays.</td>
</tr>
<tr>
<td>Note:</td>
<td>You can use MSGDATA as an alias for MESSAGEDATA.</td>
</tr>
<tr>
<td>None</td>
<td>Message Advisor does not provide report output</td>
</tr>
<tr>
<td>ORIGIN</td>
<td>Message Advisor provides a report that contains information on each origin for which it requeued messages</td>
</tr>
<tr>
<td>SCRAP</td>
<td>Message Advisor reports on messages it wrote to the scrap file</td>
</tr>
<tr>
<td>Note:</td>
<td>The SCRAP report duplicates the ERROR report unless you use the SCRAP or ERROR keywords to select different scrap codes.</td>
</tr>
<tr>
<td>SCRAPDATA</td>
<td>Message Advisor displays messages that it wrote to the scrap file Use the MAXMSGS and MAXDATA keywords to limit the number of messages and the amount of data that Message Advisor displays.</td>
</tr>
<tr>
<td>Note:</td>
<td>The SCRAPDATA report duplicates the ERRORDATA report unless you use the SCRAP or ERROR keywords to select different scrap codes.</td>
</tr>
<tr>
<td>SUMMary</td>
<td>Message Advisor provides a statistical summary of requeue processing</td>
</tr>
</tbody>
</table>

**REQ_PRompt/CHKPT_PRompt keyword**

REQ_PRompt is an optional keyword for the REQueue command and indicates whether Message Advisor issues a write-to-operator with reply (WTOR) prompt when it executes a `REQueue TYPE=COLD`. The WTOR indicates the checkpoint that Message Advisor will use for the requeue.

You can use CHKPT_PRompt as an alias for REQ_PRompt.

**Note**
The REQ_PRompt keyword value for the REQueue command overrides the REQ_PRompt keyword value for the SERVER_Options subcommand on the CUStomize command set.

Table 245 on page 249 lists the REQ_PRompt keyword parameters.
Table 245: REQ_PROMPT keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>When Message Advisor executes a REQueue TYPE=COLD, it issues a WTOR prompt that indicates the checkpoint to be used for the requeue. <strong>Note:</strong> BMC Software recommends that you specify Yes if you use the AUTO_REQ, AUTO_REQ_COLDCOM, or AUTO_REQ_COLDSYS options for the IMS_Options subcommand on the CUSTOMIZE command set.</td>
</tr>
<tr>
<td>No</td>
<td>Message Advisor does not issue a WTOR prompt when it executes a REQueue TYPE=COLD</td>
</tr>
</tbody>
</table>

**SCRap/SCP keyword**

SCRap is an optional keyword for the REQueue command and indicates which of the messages that cannot be requeued are written to the scrap file. Consider the following when specifying the SCRap keyword:

- You can specify ALL, None, or a list of scrap codes or values (other than ALL or None).

- You can use either scrap codes or the names assigned to the scrap codes. For example, to scrap messages for scrap codes 1, 3, and 5, you can specify `SCRap=(1,3,5)` or `SCRap=(NOSOURCE,INSERT,LENGTH)`. See the ERROR keyword description for a list of scrap codes and names.

- You can negate scrap codes by preceding them with ‘¬’. For example, to scrap all codes except 1 and 2, you can specify `SCRap=(¬1,¬2)`. You can use SCP as an alias for SCRap.

Table 246 on page 250 lists the SCRap keyword parameters.
Table 246: SCRap keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>Message Advisor scraps only those messages that cannot be requeued because of an error. SCRap=ERROR is equivalent to SCRap=(NOSOURCE,NODEST,INSERT,ENQFAIL,LENGTH,PREFIX,QSTOP,CHANGE,CANCEL,CONVCCB,NOAPPC,RECORD,BADQUEUE,NODATA,NOSYSID,NOOTMA,BADSYS) You can include other scrap codes or names with the default ERROR scrap codes. For example, to scrap messages with zero time stamps in addition to the default ERROR codes, you can specify SCRap=(ERROR,ZEROTIME). You can also subtract from the default ERROR scrap codes by inserting ¬ in front of the applicable code or name. For example, to exclude messages that are too long from the default scrap codes, you can specify SCRap=(ERROR,¬LENGTH).</td>
</tr>
<tr>
<td>nn</td>
<td>Message Advisor scraps messages for the scrap codes you specify. You can use scrap code names in place of the scrap codes.</td>
</tr>
<tr>
<td>No/None</td>
<td>Message Advisor does not scrap messages for any scrap codes. Note: You can use None as an alias for No.</td>
</tr>
<tr>
<td>NOTERROR</td>
<td>Message Advisor scraps messages for all scrap codes other than those included in SCRap=ERROR</td>
</tr>
<tr>
<td>Yes/ALL</td>
<td>Message Advisor scraps messages for all scrap codes. Note: You can use ALL as an alias for Yes.</td>
</tr>
</tbody>
</table>

**TYPE keyword**

TYPE is an optional keyword for the REQueue command and indicates the type of requeue Message Advisor performs.

Table 247 on page 251 lists the TYPE keyword parameters.
# Table 247: TYPE keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLD</td>
<td>Message Advisor uses the DUMPQ/PURGE checkpoint log records from the last IMS shutdown to perform the requeue. You can use the CHKPT keyword to change the checkpoint. Note: TYPE=COLD requires a DUMPQ or PURGE shutdown checkpoint. If you want to use a SNAPQ checkpoint, you must specify TYPE=EREfail. If you specify REQ_PRompt=Yes, WTOR message BMC43298I displays on the MVS console for checkpoint validation. For more information, see the <em>Message Advisor for IMS User Guide</em>. For TYPE=COLD, SLDS input may be supplied via job control language (JCL). To supply the SLDS via JCL, add a //SLDS dd statement that describes the data set to use as requeue input in the Message Advisor Server JCL.</td>
</tr>
<tr>
<td>EREfail</td>
<td>Message Advisor selects the requeue checkpoint from the Checkpoint Tracking dataset, and then requeues messages beginning with the selected checkpoint. Message Advisor requeues all unprocessed messages to the first abend after the selected checkpoint. Message Advisor uses the RECON data set to locate the SLDS containing the selected checkpoint and any subsequent message data. You can use the INPUT subcommand to provide input if the RECON data set is unavailable. You must close and archive all OLDS before running a REQueue TYPE=EREfail. You can use the CHKPT keyword to change the checkpoint. Note: WTOR message BMC43298I always displays on the MVS console for checkpoint validation. For more information, see the <em>Message Advisor for IMS User Guide</em>. For TYPE=EREfail, SLDS input may be supplied via job control language (JCL). To supply the SLDS via JCL, add a //SLDS dd statement that describes the data set to use as requeue input in the Message Advisor Server JCL.</td>
</tr>
<tr>
<td>FILE</td>
<td>Message Advisor obtains data from the data set you specify on the INPUT subcommand. You must use an extract file or an unload file as the input file.</td>
</tr>
</tbody>
</table>
REQUEUE command set description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| REPRocess    | Message Advisor begins the requeue with the selected checkpoint and ends the requeue at the end of the input.  
Message Advisor uses the RECON data set to locate the SLDS containing the specified checkpoint, along with all other SLDS in the same PRISLDS record. You can use the INPUT subcommand to provide input if the RECON data set is unavailable, or if you do not want to use all of the SLDS that Message Advisor would implicitly select.  
**Note:** WTOR message BMC43298I always displays on the MVS console for checkpoint validation. For more information, see the *Message Advisor for IMS User Guide*.  
If you specify TYPE=REPRocess and include an INTERVAL subcommand with the REQueue command set but do not also specify the CHKPT keyword, the following processing changes occur:  
- WTORs are not sent to the console to confirm checkpoint information.  
- Message Advisor selects log files solely on the basis of the INTERVAL parameters.  
- Message Advisor does not capture messages on the queues as of the START_TIME specified on the INTERVAL subcommand; Message Advisor only captures messages added to the queues during the interval time period (01 and 03 records).  
- The Statistics report has fewer output lines since many of the log record types are ignored.  
BMC Software recommends that you specify MOde=EXtract along with TYPE=REPRocess. You can then issue a REQueue TYPE=FILE with optional SELECT and/or CHANGE subcommands in order to place the output back on the same IMS or on a test IMS system.  
REQueue TYPE=REPRocess is intended as an application testing tool and is not recommended for use in a production environment.  
**Note:** For TYPE=REPRocess, SLDS input may be supplied via job control language (JCL). To supply the SLDS via JCL, add a //SLDS dd statement that describes the data set to use as requeue input in the Message Advisor Server JCL.|

**USER_PREFIX keyword**

USER_PREFIX is an optional keyword for the REQueue command and indicates whether Message Advisor retains user prefixes that may have been added to messages via an IMS exit.  
Table 248 on page 253 lists the USER_PREFIX keyword parameters.
Table 248: USER_PREFIX keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEEP</td>
<td>Message Advisor retains user prefixes when it requeues messages</td>
</tr>
<tr>
<td>DISCARD</td>
<td>Message Advisor does not retain user prefixes when it requeues messages</td>
</tr>
</tbody>
</table>

**VALIDATE_SRC keyword**

VALIDATE_SRC is an optional keyword for the REQueue command and indicates whether Message Advisor validates the source LTERM for each message it requeues.

*Note*

The VALIDATE_SRC keyword is only valid for systems with ETO, ETO/ETA, or VTF defined.

Table 249 on page 253 lists the VALIDATE_SRC keyword parameters.

Table 249: VALIDATE_SRC keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Message Advisor validates the source LTERM for each input transaction message it requeues. If the source LTERM fails validation, Message Advisor does not requeue it and scraps it with scrap code ‘NOSOURCE’.</td>
</tr>
<tr>
<td>No</td>
<td>Message Advisor does not validate the source LTERM for each message it requeues</td>
</tr>
</tbody>
</table>

**ZERO_time keyword**

ZERO_time is an optional keyword for the REQueue command and indicates what action Message Advisor takes for messages with zero time stamps.

Table 250 on page 254 lists the ZERO_time keyword parameters.
Table 250: ZERO_time keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQueue</td>
<td>Message Advisor requeues messages with a zero time stamp without checking them against INTERVAL subcommands</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> REQueue is the default if you do <em>not</em> include INTERVAL subcommands with the REQUEUE command set.</td>
</tr>
<tr>
<td>SCRap/SCP</td>
<td>Message Advisor scraps all messages with a zero time stamp</td>
</tr>
<tr>
<td></td>
<td>You can use SCP as an alias for SCRap.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> SCRap is the default if you include INTERVAL subcommands with the REQueue command set.</td>
</tr>
<tr>
<td>USE_LASTmsg</td>
<td>Message Advisor requeues a zero time stamp message only if it requeued the previous message</td>
</tr>
<tr>
<td></td>
<td>Message Advisor uses the time stamp from the previous message to check INTERVAL subcommands.</td>
</tr>
<tr>
<td></td>
<td>For checkpointed messages, the previous message is the message queued just before the message in question on a destination queue. For logged messages, the previous message is the previous message in the log for any destination. If no previous message exists, Message Advisor scraps the message in question.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If you do not include an INTERVAL subcommand with the REQueue command set, you cannot specify USE_LASTmsg.</td>
</tr>
</tbody>
</table>

**CHANGE subcommand**

You can include the CHANGE subcommand with the REQueue command set to apply changes to messages. The CHANGE subcommand is not required for the REQueue command set.

All of the SELECT and REJECT subcommand keywords are allowed on the CHANGE subcommand. Use the SELECT keywords to select the messages to change, and then use the CHANGE keywords to specify what to change for the selected messages. See “SELECT and REJECT subcommands” on page 313, for SELECT keyword and parameter descriptions. Table 251 on page 255 lists the keywords that are common to the SELECT, REJECT, CHANGE, and GROUP subcommands.
### Table 251: Keywords common to SELECT/REJECT/CHANGE/GROUP subcommands

<table>
<thead>
<tr>
<th>Keyword Group</th>
<th>Description</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>destination-related</td>
<td>select which destinations to process</td>
<td>AFFINITY, CNT, CONVID, DBD, DEADQ, DESTINATION, DESTYPE, DYNAMIC, HELD, KEY, LTERM, LUNAME, MSNAME, PSB, QTYPE, RSMB, SMB, STATIC, STATIC, TMEMBER, TPIPE, TPNAMES, TRANSACTION, VSPCNT</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Most of the destination-related keywords (except AFFINITY) are mutually exclusive. In general, you should only specify one destination-related keyword; however, the following combinations are allowed:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ DESTINATION and DESTYPE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ DESTINATION and QTYPE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ TPNAME and LUNAME</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ TMEMBER and TPIPE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ HELD and CONVID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The AFFINITY keyword is message-related for the REQUEUE command.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The QTYPE keyword is destination-related and message-related.</td>
<td></td>
</tr>
<tr>
<td>queue-related</td>
<td>select which queues to process for the selected destinations</td>
<td>QUEUE, REGION, REGION_ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keyword Group</td>
<td>Description</td>
<td>Keywords</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>message-related keywords</td>
<td>select which messages to process for the selected queues</td>
<td>COMMITMODE c, DRRN a, ENTRYID a, MFSNAME, MSGTYPE, NEWCOMMITMODE c, NEWER_THAN, NEWORGQNAME, NEWORGQMANAGER, NEWORGTPPIPE, NEWORGTMEMBER, NODENAME, OIMS, OLDER_THAN, ORIGIN, OTOKEN, PIMS, PTOKEN, QPOSITION a, RCNT, START, STOP, SYSID, UOWID, USERID</td>
</tr>
<tr>
<td>data-related keywords</td>
<td>select which messages to process for the selected queues</td>
<td>DATA, DATASIZE, OFFSET, SEGMENT</td>
</tr>
<tr>
<td>other keywords</td>
<td>special purpose</td>
<td>LABEL, SCRAPCODE c</td>
</tr>
</tbody>
</table>

- **a** Keyword not allowed with the REQUEUE command
- **b** Keyword allowed only on the DISPLAY command
- **c** Keyword allowed only on the REQUEUE command

The following considerations apply to the CHANGE subcommand:
Message Advisor processes all SELECT and REJECT subcommands before it applies CHANGE subcommands. The original values are used to select or reject messages, not the newly-changed values.

The SELECT keywords specified on SELECT and REJECT subcommands that you include with the REQueue command set determine which messages to requeue.

The SELECT keywords specified on CHANGE subcommands that you include with the REQueue command set determine which messages to change.

If you do not include any SELECT keywords in the CHANGE subcommand, Message Advisor changes all messages selected as a result of SELECT, REJECT, and INTERVAL subcommand processing.

If a message does not match the selection keywords on any CHANGE subcommand but does match the selection keywords on a SELECT subcommand, Message Advisor requeues the message without changing it.

If you specify the NEWDEST and/or NEWORIGIN keywords on a CHANGE subcommand, the only SELECT keywords allowed are DESTINATION and/or ORIGIN.

If you specify the NEWTPNAME and/or NEWLUNAME keywords on a CHANGE subcommand, the only SELECT keywords allowed are DESTINATION or TPNAME, and/or LUNAME and/or ORIGIN. You can also specify the DESTYPE keyword, but APPC/TPNAME is the only value allowed.

If you specify the NEWTMEMBER and/or NEWTPIPE keywords on a CHANGE subcommand, the only SELECT keywords allowed are DESTINATION or TMEMBER, and/or TPIPE. You can also specify the DESTYPE keyword, but OTMA is the only value allowed.

**CHANGE subcommand syntax**

The following figure shows the CHANGE subcommand syntax.
Figure 54: CHANGE subcommand syntax

CHANGE subcommand description

The CHANGE subcommand consists of the following items:

- CHANGE primary command
- SELECT keywords and parameters
- CHANGE keywords and parameters

Keyword and parameter descriptions for the SELECT subcommand appear in “SELECT and REJECT subcommands” on page 313.
Keyword and parameter descriptions for the CHANGE subcommand are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

**COMMITMODE keyword**

Use the COMMITMODE keyword to identify the CM0 or CM1 messages to be selected for CHANGE.

For example, during REQUEUE, to change CM0 messages to CM1 messages, use the following CHANGE command:

```plaintext
CHANGE COMMITMODE=CM0,NEWCOMMITMODE=CM1
```

Table 252 on page 259 lists the COMMITMODE keyword parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM0</td>
<td>Commit Mode 0 (commit-then-send)</td>
</tr>
<tr>
<td>CM1</td>
<td>Commit Mode 1 (send-then-commit)</td>
</tr>
</tbody>
</table>

**LABEL keyword**

Use the LABEL keyword to provide a name (label) for the CHANGE subcommand statement. Message Advisor uses the LABEL keyword value to identify the statement in reports.

Table 253 on page 259 lists the LABEL keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| CHG00001 cccccccc up to 8 characters | the name of this CHANGE subcommand statement  
**Note:** The default label is CHG00001 for the first statement. The number increments by one for each subsequent statement. cccccccc is a value that you specify.
NEWAFFINITY keyword

Use the NEWAFFINITY keyword to set a new seven-character affinity. To remove the affinity, specify \texttt{NEWAFFINITY= } , \texttt{NEWAFFINITY=’ ’}, or \texttt{NEWAFFINITY=&NONE}.

\textbf{Note}
The NEWAFFINITY keyword applies to shared queues only.

Table 254 on page 260 lists the NEWAFFINITY keyword parameters.

Table 254: NEWAFFINITY keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{ccccccc}</td>
<td>the new seven-character affinity</td>
</tr>
<tr>
<td>\texttt{Note}: \texttt{ccccccc} is a value that you specify.</td>
<td></td>
</tr>
<tr>
<td>&amp;IMS</td>
<td>sets the affinity to the current IMS system</td>
</tr>
<tr>
<td>\texttt{Note}: You cannot set NEWAFFINITY to &amp;IMS on a REQueue EXTRACT command.</td>
<td></td>
</tr>
<tr>
<td>&amp;NONE</td>
<td>removes the affinity</td>
</tr>
</tbody>
</table>

NEWCOMMITMODE keyword

Use the NEWCOMMITMODE keyword to CHANGE the Commit Mode of the messages identified by the COMMITMODE keyword during REQUEUE. For example, during REQUEUE, to change CM0 messages to CM1 messages use the following CHANGE command:

\texttt{CHANGE COMMITMODE=CM0, NEWCOMMITMODE=CM1}

Table 255 on page 260 lists the NEWCOMMITMODE keyword parameters.

Table 255: NEWCOMMITMODE keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM0</td>
<td>Commit Mode 0 (commit-then-send)</td>
</tr>
<tr>
<td>CM1</td>
<td>Commit Mode 1 (send-then-commit)</td>
</tr>
</tbody>
</table>

NEWDEST keyword

Use the NEWDEST keyword to change the destination name for the selected messages.
Consider the following when specifying the NEWDEST keyword:

- You can change LTERM names to transaction names, and you can change transaction names to LTERM names.

- You can change APPC TPNAMEs and OTMA member names, but you cannot change the destination type. BMC Software recommends using the NEWTPNAME or NEWTMEMBER keywords to apply changes to APPC TPNAMEs and OTMA member names.

- You cannot use the DESTYPE keyword or any keyword that implies DESTYPE with the NEWDEST keyword. For example, you cannot specify CNT=ABC,NEWDEST=XYZ.

- Destination and ORIGIN are the only destination-related keywords allowed with the NEWDEST keyword.

Table 256 on page 261 lists the NEWDEST keyword parameter.

### Table 256: NEWDEST keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cc....cc</td>
<td>the new destination name</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> cc....cc is a value that you specify.</td>
</tr>
</tbody>
</table>

### NEWLUname keyword

Use the NEWLUname keyword to change the APPC LUNAME for the selected messages. The value that you specify indicates the APPC LUNAME to be set in the APPC prefix.

**Note**

NEWLUname requires DESTYPE=TPNAME so that Message Advisor selects only messages with an LUNAME for change.

The only destination-related keywords allowed with the NEWLUname keyword are NEWTPNAME, LUname, TPName, and ORIGIN.

Table 257 on page 261 lists the NEWLUname keyword parameter.

### Table 257: NEWLUname keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccccccccc.ccccccccc</td>
<td>the new APPC LUNAME to be set in the APPC prefix</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> ccccccccc.ccccccccc is a value that you specify.</td>
</tr>
</tbody>
</table>
**NEWMFSNAME keyword**

Use the NEWMFSNAME keyword to change the message format service (MFS) name for the selected messages. The value that you specify indicates the MFS name to be set in the message prefix.

Table 258 on page 262 lists the NEWMFSNAME keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| cccccccc  | the new MFS name to be set in the message prefix  
  **Note:** cccccccc is a value that you specify.  
  If the message to be changed did not contain an MFS name, Message Advisor scraps the message with a CHANGE scrap code. |

**NEWNODENAME keyword**

Use the NEWNODENAME keyword to change the Virtual Telecommunications Access Method (VTAM) node name for the selected messages. The value that you specify indicates the VTAM node name to be set in the system prefix.

Table 259 on page 262 on page 261 lists the NEWNODENAME keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| cccccccc  | the new VTAM node name to be set in the system prefix  
  **Note:** cccccccc is a value that you specify.  
  If the message to be changed did not contain a VTAM node name, Message Advisor scraps the message with a CHANGE scrap code. |

**NEWOIMS keyword**

Use the NEWOIMS keyword to change the originating IMS for the selected messages. The value that you specify indicates the originating IMS to be set in the message prefix.

Table 260 on page 263 lists the NEWOIMS keyword parameter.
NEWORIGIn keyword

Use the NEWORIGIn keyword to change the origin for the selected messages. The value that you specify indicates the origin to be set in the message prefix.

**Note**
You cannot change APPC origins. You cannot change OTMA origins with the NEWORIGIn keyword. Instead, use the NEWORGTMEMBER and NEWORGTPPIPE keywords. For OTMA MQSeries messages, you can also use the NEWORGQMANAGER and NEWORGQNAME keywords to change the Reply-to queue.

The only destination-related keywords allowed with the NEWORIGIn keyword are DESTination, ORIGIn, and NEWDEST.

Table 261 on page 263 lists the NEWORIGIn keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cccccccc</code></td>
<td>the new origin to be set in the message prefix</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td><code>cccccccc</code> is a value that you specify.</td>
</tr>
<tr>
<td></td>
<td>If the message to be changed did not contain a standard origin, Message</td>
</tr>
<tr>
<td></td>
<td>Advisor scraps the message with a CHANGE scrap code.</td>
</tr>
</tbody>
</table>

NEWORGQNAME keyword

Use the NEWORGQNAME keyword to change the origin MQSERIES Reply-to QNAME for the selected messages.

Table 262 on page 264 lists the NEWORGQNAME keyword parameter.
Table 262: NEWORQQNAME keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>newOriginQname</td>
<td>the new origin MQSERIES Reply-to QNAME</td>
</tr>
<tr>
<td></td>
<td>Note: newOriginQname is a value that you specify.</td>
</tr>
</tbody>
</table>

**NEWORQQMANAGER keyword**

Use the NEWORQQMANAGER keyword to change the origin MQSERIES Reply-to QMANAGER name for the selected messages.

Table 263 on page 264 lists the NEWORQQMANAGER keyword parameter.

Table 263: NEWORQQMANAGER keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>newOriginQmanager</td>
<td>the new origin MQSERIES Reply-to QMANAGER name</td>
</tr>
<tr>
<td></td>
<td>Note: newOriginQmanager is a value that you specify.</td>
</tr>
</tbody>
</table>

**NEWORGTPIPE keyword**

Use the NEWORGTPIPE keyword to change the origin TPIPE name for the selected messages.

Table 264 on page 264 lists the NEWORGTPIPE keyword parameter.

Table 264: NEWORGTPIPE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>newOriginTpipe</td>
<td>the new origin TPIPE name</td>
</tr>
<tr>
<td></td>
<td>Note: newOriginTpipe is a value that you specify.</td>
</tr>
</tbody>
</table>

**NEWORGTMEMBER keyword**

Use the NEWORGTMEMBER keyword to change the origin TMEMBER name for the selected messages.

Table 265 on page 265 lists the NEWORGTMEMBER keyword parameter.
Table 265: NEWORGTMEMBER keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>newOriginTmember</td>
<td>the new origin TMEMBER name</td>
</tr>
<tr>
<td></td>
<td>Note: newOriginTmember is a value that you specify.</td>
</tr>
</tbody>
</table>

NEWPIMS keyword

Use the NEWPIMS keyword to change the processing IMS for the selected messages. The value that you specify indicates the processing IMS to be set in the message prefix.

**Note**

In a shared queues environment, when an input message is placed on the IMS queue, the current IMS ID always overrides the NEWPIMS keyword value.

Table 266 on page 265 lists the NEWPIMS keyword parameter.

Table 266: NEWPIMS keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccccc</td>
<td>the new processing IMS to be set in the message prefix</td>
</tr>
<tr>
<td></td>
<td>Note: cccccccc is a value that you specify.</td>
</tr>
<tr>
<td></td>
<td>If the message to be changed did not contain a processing IMS, Message Advisor scraps the message with a CHANGE scrap code.</td>
</tr>
</tbody>
</table>

NEWQueue keyword

Use the NEWQueue keyword to change the queue for the selected messages. The value that you specify indicates the queue on which Message Advisor places the messages.

Table 267 on page 265 lists the NEWQueue keyword parameters.

Table 267: NEWQueue keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>the new queue on which Message Advisor places the selected messages</td>
</tr>
<tr>
<td>2</td>
<td>Note: If the message to be changed is destined for a scheduler message block (SMB), only queue 1 is valid. If you specify any other queue, Message Advisor scraps the message with a CHANGE scrap code.</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
**NEWSAFGROUP keyword**

Use the NEWSAFGROUP keyword to change the SAF Group ID for the selected messages. The value that you specify indicates the SAF Group ID to be set in the message prefix.

Table 268 on page 266 lists the NEWSAFGROUP keyword parameter.

Table 268: NEWSAFGROUP keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccc</td>
<td>the new SAF Group ID to be set in the message prefix</td>
</tr>
</tbody>
</table>

*Note:* `ccccccc` is a value that you specify. If the message to be changed did not contain a SAF Group ID, Message Advisor scraps the message with a CHANGE scrap code.

**Note**

Using the NEWUSERID or NEWSAFGROUP keywords during a REQUEUE operation will now result in a regeneration of the security token found in the APPC and OTMA prefix segments.

**NEWSYSID keyword**

Use the NEWSYSID keyword to change the system ID for the selected messages. Consider the following when specifying the NEWSYSID keyword:

- The keyword value must be two numbers between 1 and 2036, enclosed in parentheses and separated by a comma.

- The first number of the keyword value indicates the remote system ID, and the second number indicates the local system ID.

- You can replace either number of the keyword value with an asterisk (*) to indicate that the corresponding system ID remains the same.

- You can replace either number of the keyword value with the parameters LOCAL/L or ASSIGN/A. LOCAL indicates that Message Advisor changes the system ID to the first locally-defined system ID on the current IMS system. ASSIGN indicates that Message Advisor uses the system ID that is currently appropriate for the message.

- If Message Advisor finds either of the LOCAL or ASSIGN parameters on a CHANGE subcommand when it is extracting (rather than requeuing), it scraps the message with a CHANGE scrap code. Message Advisor must scrap the
message because it requires an available IMS system in order to perform the assignment. Use MODE=(EXtract,VALIDate) to requeue in this situation.

- If you specify a number that does not correspond to a valid system ID on the IMS system, Message Advisor scraps the message with a CHANGE scrap code.

Table 269 on page 267 lists the NEWSYSID keyword parameter.

Table 269: NEWSYSID keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| (nn,nn)   | the new system ID for the selected messages  
Note: nn is a value that you specify.  
You can replace either nn value with * to indicate that the corresponding system ID remains the same.  
You can replace either nn value with LOCAL/L or ASSIGN/A. LOCAL indicates that Message Advisor changes the system ID to the first locally-defined system ID on the current IMS system. ASSIGN indicates that Message Advisor uses the system ID that is currently appropriate for the message. |

NEWTIMESTAMP keyword

Use the NEWTIMESTAMP keyword to change the time stamp for the selected messages to the current time.

Table 270 on page 267 lists the NEWTIMESTAMP keyword parameter.

Table 270: NEWTIMESTAMP keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURRENT</td>
<td>Message Advisor changes the time stamp for the selected messages to the current time</td>
</tr>
</tbody>
</table>

NEWTMEMBER/NEWMEMBER keyword

Use the NEWTMEMBER keyword to change the OTMA member name for the selected messages.

Note

NEWTMEMBER requires DESTYPE=OTMA so that Message Advisor only selects messages with a member name for change.  
The only destination-related keywords allowed with the NEWTMEMBER keyword are NEWTPIPE, TMEMBER, TPIPE, and ORIGin.
You can use NEWMEMBER as an alias for NEWTMEMBER.

Table 271 on page 268 lists the NEWTMEMBER keyword parameter.

Table 271: NEWTMEMBER keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccccccccccccccccc</td>
<td>the new OTMA member name for the selected messages</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>ccccccccccccccccc is a value that you specify.</td>
</tr>
</tbody>
</table>

**NEWTPPIPE/NEWPIPE keyword**

Use the NEWTPPIPE keyword to change the OTMA TPIPE name for the selected messages. The value that you specify indicates the new OTMA TPIPE name to be set in the OTMA prefix.

**Note**

NEWTPPIPE requires DESTYPE=OTMA so that Message Advisor only selects messages with a TPIPE for change.

The only destination-related keywords allowed with the NEWTPPIPE keyword are NEWTMEMBER, TMEMBER, TPIPE, and ORIGin.

You can use NEWPIPE as an alias for NEWTPPIPE.

Table 272 on page 268 lists the NEWTPPIPE keyword parameter.

Table 272: NEWTPPIPE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccccc</td>
<td>the new OTMA TPIPE name to be set in the OTMA prefix</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>ccccccccc is a value that you specify.</td>
</tr>
</tbody>
</table>

**NEWTPNAME keyword**

Use the NEWTPNAME keyword to change the TPNAME for the selected messages.

**Note**

NEWTPNAME requires DESTYPE=TPNAME so that Message Advisor only selects messages with a TPNAME for change. DESTYPE=TPNAME and TPNAME are the only keywords that you can specify with NEWTPNAME.

The only destination-related keywords allowed with the NEWTPNAME keyword are NEWLUname, TPName, LUname, and ORIGin.
Table 273 on page 269 lists the NEWTPNAME keyword parameter.

Table 273: NEWTPNAME keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cc....cc</td>
<td>the new TPNAME for the selected messages</td>
</tr>
</tbody>
</table>

Note: cc....cc is a value that you specify.

**NEWUSERID keyword**

Use the NEWUSERID keyword to change the RACF user ID for the selected messages. The value that you specify indicates the RACF user ID to be set in the message prefix.

Table 274 on page 269 lists the NEWUSERID keyword parameter.

Table 274: NEWUSERID keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccccc</td>
<td>the new RACF user ID to be set in the message prefix</td>
</tr>
</tbody>
</table>

Note: cccccccc is a value that you specify.

If the message to be changed did not contain a RACF user ID, Message Advisor scraps the message with a CHANGE scrap code.

---

**Note**

Using the NEWUSERID or NEWSAFGROUP keywords during a REQUEUE operation will now result in a regeneration of the security token found in the APPC and OTMA prefix segments.

**REPLACE keyword**

Use the REPLACE keyword along with the DATA keyword to replace all occurrences of the DATA keyword value in the message text with the REPLACE keyword value. Consider the following when specifying the REPLACE keyword:

- You can specify a maximum of 256 characters.
- You do not have to enclose the string in quotes as long as no special characters are present and the string fits on one line; otherwise, you must enclose the string in quotes.

**Note**

If you do not enclose the string in quotes, the string is translated to uppercase.
You can specify a hexadecimal string.

You can specify a null string.

If the length of the text segment changes, Message Advisor automatically adjusts the segment header word. The adjustment could cause Message Advisor to scrap the message with an invalid length.

You can use the SEGMENT and OFFSET keywords to limit which occurrences of the DATA keyword are replaced.

Table 275 on page 270 lists the REPLACE keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cc..cc</td>
<td>the data string that will replace the string specified for the DATA keyword</td>
</tr>
<tr>
<td></td>
<td>Note: cc..cc is a value that you specify.</td>
</tr>
</tbody>
</table>

GROUP subcommand

You can include the GROUP subcommand with the REQueue command set in order to define message groups for the SELECT report. See the Message Advisor for IMS User Guide for a description of the SELECT report. The GROUP subcommand is not required for the REQueue command set.

All of the SELECT and REJECT subcommand keywords are allowed on the GROUP subcommand. The SELECT keywords select which messages to include in the group. See “SELECT and REJECT subcommands” on page 313, for SELECT keyword and parameter descriptions. See the table in “CHANGE subcommand” on page 254 for a list of the keywords that are common to the SELECT, REJECT, CHANGE, and GROUP subcommands.

Because Message Advisor processes all SELECT and REJECT subcommands before applying GROUP subcommands, only the selected messages are counted for a GROUP subcommand.

GROUP subcommand syntax

The following figure shows the GROUP subcommand syntax.
Figure 55: GROUP subcommand syntax

GROUP subcommand description

The GROUP subcommand consists of the following items:

- GROUP primary command
- LABEL keyword
- SELECT keywords and parameters

Keyword and parameter descriptions for the SELECT subcommand appear in “SELECT and REJECT subcommands” on page 313.

Note

Keyword and parameter descriptions for the GROUP subcommand are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

LABEL keyword

Use the LABEL keyword to provide a name (label) for the GROUP subcommand statement. Message Advisor uses the LABEL keyword value to identify the statement in reports.

Table 276 on page 271 lists the LABEL keyword parameter.

Table 276: LABEL keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRP00001</td>
<td>the name of this GROUP subcommand statement</td>
</tr>
<tr>
<td>ccccccccc up to 8 characters</td>
<td>Note: The default label is GRP00001 for the first statement. The number increments by one for each subsequent statement. ccccccccc is a value that you specify.</td>
</tr>
</tbody>
</table>
INPUT subcommand

You can include the INPUT subcommand with the REQueue command set in order to identify a data set to be used as input to the requeue action. The INPUT subcommand is required for REQueue TYPE=FILE and optional for the other REQUEUE types.

INPUT subcommand syntax

The following figure shows the INPUT subcommand syntax.

Figure 56: INPUT subcommand syntax

```
INPUT
  ,DDName/File=cccccccc
  ,DSName/Dataseq=dsname
  ,UNIT=cccccccc
  ,VOLser/VOLUME=(cccccc,...)
```

INPUT subcommand description

The INPUT subcommand consists of the following items:

- INPUT primary command
- keywords and parameters

Note

Keyword and parameter descriptions for the INPUT subcommand are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

DDName/File keyword

Use the DDName keyword to specify a pre-allocated data set to use as input to the requeue.

Note

The DDName keyword is only allowed with a batch server and is mutually exclusive with all other INput subcommand keywords.

You can use File as an alias for DDName.

Table 277 on page 273 lists the DDName keyword parameter.
Table 277: DDName keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccc</td>
<td>the name of the pre-allocated data set to use as input to the requeue</td>
</tr>
</tbody>
</table>

Note: cccccc is a value that you specify.

DSName/DAtaset keyword

Use the DSName keyword to identify a data set to use as input to the requeue.

Consider the following when specifying the DSName keyword:

- You can specify a generation data group (GDG) with (0), (+n), or (-n).
- The relative number is in relation to the current job, such that the same relative number always refers to the same data set within a job.
- The data set must exist.
- If you use (+n), a previous jobstep or command set must create the data set.
- You can include symbolic keywords when specifying a value for the DSName keyword. A symbolic keyword begins with a percent sign (%) and optionally ends with a percent sign (%). All supported symbolic keywords are allowed, but only %USER and %IMSID are useful since the data set must exist and the other symbolic keywords make predefining the data set difficult.
  A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the remaining characters are concatenated to the symbolic value.
- You can include system symbols when specifying a value for the DSName keyword. A system symbol begins with an ampersand (&) and optionally ends with a period (.). The system will substitute a value for valid symbols. You will receive an allocation error if you use an undefined system symbol or if the substitution results in an invalid data set name. See the JCL Reference Manual for more information on system symbols.

See “Command syntax considerations” on page 37 for additional items to consider when specifying the DSName keyword value.

You can use DAtaset as an alias for DSName.

Table 278 on page 274 lists the DSName keyword parameter.
Table 278: DSName keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dsname</td>
<td>the name of the data set to use as input to the requeue</td>
</tr>
<tr>
<td>Note:</td>
<td>dsname is a value that you specify.</td>
</tr>
</tbody>
</table>

**UNIT keyword**

Use the UNIT keyword to indicate the generic unit name to apply when allocating the data set.

Table 279 on page 274 lists the UNIT keyword parameter.

Table 279: UNIT keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccc</td>
<td>the generic unit name to apply when allocating the data set</td>
</tr>
<tr>
<td>Note:</td>
<td>ccccccc is a value that you specify.</td>
</tr>
</tbody>
</table>

**VOLser/VOLUMe keyword**

Use the VOLser keyword to indicate the volume serial number for an uncataloged data set.

*Note*

If you specify the VOLser keyword, you must also specify the UNIT keyword.

You can use VOLUMe as an alias for VOLser.

Table 280 on page 274 lists the VOLser keyword parameter.

Table 280: VOLser keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(cccccc,...)</td>
<td>a list of volume serial numbers for an uncataloged data set</td>
</tr>
<tr>
<td>Note:</td>
<td>cccccc is a value that you specify.</td>
</tr>
</tbody>
</table>
**SPILL subcommand**

You can include the SPILL subcommand with the REQueue command set in order to override spill file specifications created during the CUSTOMIZE process.

(For more information about the CUSTOMIZE process, see “CUSTOMIZE command” on page 147.)

The SPILL subcommand causes Message Advisor to allocate temporary spill data sets. The SPILL subcommand is not required for the REQueue command set. If the REQueue command set does not include a SPILL subcommand, Message Advisor uses the spill data set specified in the customization options.

If needed, Message Advisor will allocate two VSAM and two non-VSAM spill data sets. Message Advisor always allocates non-VSAM data sets as temporary data sets. VSAM data sets will be defined when needed and deleted at the end of the requeue. If the VSAM data sets already exist, Message Advisor will use the pre-existing data sets.

**SPILL subcommand syntax**

The following figure shows the SPILL subcommand syntax.

*Figure 57: SPILL subcommand syntax*

```
SPILL             CYLS_PRIM=nnnnnnnn         PREFIX=dsname

   CYLS_SEC=nnnnnnnn  
   .DATACLASS=cccccccc  
   .MGMTCLASS=cccccccc  
   .STORCLASS=cccccccc  
   .UNIT=cccccccc  
   .VOLUMENV=(cccccc,...)  
```

**SPILL subcommand description**

The SPILL subcommand consists of the following items:

- SPILL primary command
- keywords and parameters
Keyword and parameter descriptions for the SPILL subcommand are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

**Note**

CYLS_PRIM keyword

CYLS_PRIM keyword is a required keyword for the SPILL subcommand. Use the CYLS_PRIM keyword to indicate the number of primary cylinders to allocate for new data sets.

Table 281 on page 276 lists the CYLS_PRIM keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnnnnn</td>
<td>the number of primary cylinders to allocate for new data sets</td>
</tr>
</tbody>
</table>

Note: nnnnnnnnn is a value that you specify.

CYLS_SEC keyword

CYLS_SEC keyword is an optional keyword for the SPILL subcommand. Use the CYLS_SEC keyword to indicate the number of secondary cylinders to allocate for new data sets.

Table 282 on page 276 lists the CYLS_SEC keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnnnnn</td>
<td>the number of secondary cylinders to allocate for new data sets</td>
</tr>
</tbody>
</table>

Note: nnnnnnnnn is a value that you specify.

DATACLAS keyword

DATACLAS keyword is an optional keyword for the SPILL subcommand. Use the DATACLAS keyword to indicate the SMS data class for new VSAM data sets. Message Advisor ignores the DATACLAS keyword for existing data sets.

Table 283 on page 277 lists the DATACLAS keyword parameter.
Table 283: DATACLAS keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccc</td>
<td>the SMS data class for new VSAM data sets</td>
</tr>
<tr>
<td>Note:</td>
<td>cccccc is a value that you specify.</td>
</tr>
</tbody>
</table>

MGMTCLAS keyword

MGMTCLAS is an optional keyword for the SPILL subcommand. Use the MGMTCLAS keyword to indicate the SMS management class for new VSAM data sets. Message Advisor ignores the MGMTCLAS keyword for existing data sets.

Table 284 on page 277 lists the MGMTCLAS keyword parameter.

Table 284: MGMTCLAS keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccc</td>
<td>the SMS management class for new VSAM data sets</td>
</tr>
<tr>
<td>Note:</td>
<td>cccccc is a value that you specify.</td>
</tr>
</tbody>
</table>

PREFIX keyword

PREFIX is a required keyword for the SPILL subcommand. Use the PREFIX keyword to indicate the prefix for VSAM data sets to be used as spill files.

You can use symbolic keywords in the prefix. For more information about symbolic keywords, see “CUSTOMIZE command” on page 147.

Message Advisor appends ‘.SPILL1’ and ‘.SPILL4’ to the prefix you specify in order to create cluster names for the two required VSAM spill files. If the clusters are already defined, Message Advisor uses the predefined clusters. If the clusters do not exist, Message Advisor creates and then deletes them.

Note

The prefix does not apply to non-VSAM spill files. For the SPILL subcommand, Message Advisor allocates non-VSAM data sets as temporary data sets.

Table 285 on page 278 lists the PREFIX keyword parameter.
Table 285: PREFIX keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| dsname    | the prefix for VSAM data sets to be used as spill files  
            | Note: dsname is a value that you specify. |

**STORCLAS keyword**

STORCLAS is an optional keyword for the SPILL subcommand. Use the STORCLAS keyword to indicate the SMS storage class for new VSAM data sets. Message Advisor ignores the STORCLAS keyword for existing data sets.

Table 286 on page 278 lists the STORCLAS keyword parameter.

Table 286: STORCLAS keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| cccccc    | the SMS storage class for new VSAM data sets  
            | Note: cccccc is a value that you specify. |

**UNIT keyword**

UNIT is an optional keyword for the SPILL subcommand. Use the UNIT keyword to indicate the generic unit name to apply when allocating temporary non-VSAM spill files.

Table 287 on page 278 lists the UNIT keyword parameter.

Table 287: UNIT keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| cccccc    | the generic unit name to apply when allocating temporary non-VSAM spill files  
            | Note: cccccc is a value that you specify. |

**VOLser/VOLume keyword**

Use the VOLser keyword to indicate the volume serial number for new VSAM data sets. The VOLser keyword is not necessary for catalogued data sets.

You can use VOLume as an alias for VOLser.
Table 288 on page 279 lists the VOLser keyword parameter.

### Table 288: VOLser keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(cccccc,...)</td>
<td>the volume serial number for new VSAM data sets</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> ccccc is a value that you specify.</td>
</tr>
</tbody>
</table>
UNLOAD command

This chapter describes the Message Advisor for IMS UNLOAD command set, and also provides a syntax diagram and sample command sets. For more information about building and executing an UNLOAD command set, see the Message Advisor for IMS User Guide.

Overview

The UNLOAD command lets you unload messages from active IMS message queues to a data set, based on criteria that you specify. Message Advisor does not dequeue the messages that it unloads; instead, Message Advisor leaves the unloaded messages on the IMS message queues. The DEQUEUE command’s MODE=UNLOAD_DEQUEUE feature lets you unload and immediately dequeue messages.

You can use the UNLOAD command to perform the following tasks:

- unload messages that meet your selection criteria
- select or reject messages based on the following criteria:
  - date and time ranges
  - destinations (with masking allowed)
  - destination type
  - origins (with masking allowed)
  - other message prefix and message data selection criteria
  - conversational or non-conversation messages

Message Advisor allows the following masking characters (m) for most selection keywords:

- ? replaces exactly one character in a multicharacter string.
* replaces any number of characters.

Masking characters can be embedded in the string.

After you unload or extract messages, you can requeue them to a test system. This capability is especially useful when conducting quality assurance and performance testing.

You can also use Message Advisor to unload and dequeue messages that are queued to an inoperative Multiple Systems Coupling (MSC) link partner. When the MSC partner returns to operative status, you can use the unloaded messages data set to requeue the messages.

The UNLOAD command set consists of the UNLOAD primary command and its associated keywords and parameters, with subcommands and their associated keywords and parameters. The UNLOAD subcommands are as follows:

- INTERVAL subcommand unloads messages according to a specified time range
  For more information, see “INTERVAL subcommand” on page 293.

- OUTPUT subcommand identifies the file to which Message Advisor unloads messages
  For more information, see “OUTPUT, EXTRACT, and SCRAP subcommands” on page 303.

- SELECT subcommand specifies criteria to select specific destinations and/or messages to be unloaded
  For more information, see “SELECT and REJECT subcommands” on page 313.

- REJECT subcommand specifies criteria to exclude specific destinations and/or messages from being unloaded
  For more information, see “SELECT and REJECT subcommands” on page 313.

For more information about building and executing an UNLOAD command set, see the *Message Advisor for IMS User Guide*.

---

**Assigning conditional steps**

You can exercise additional control over a Message Advisor request that contains multiple command sets by using one or more COND pseudo command sets to insert conditional steps. Conditional steps let you perform the following tasks:

- request that Message Advisor execute a command set only if it successfully executes a previous command set
—OR—

- request that Message Advisor execute a command set only if it does not successfully execute a previous command set

---

**Note**

A COND step can (as an option) include the LABEL, IFCOND, MAXRC, and BRANCH keywords.

When you insert a COND step, a return code of 8 or higher in any command set no longer results in the remaining command sets being flushed with a BMC43111W message. Instead, Message Advisor executes all command sets (unless the COND step successfully executes a BRANCH keyword).

For more information about building and executing a COND pseudo command, see “COND pseudo command” on page 399, and see the *Message Advisor for IMS User Guide*.

---

**UNLOAD command set syntax**

The following figure shows the UNLOAD command set syntax.
Note
The SCOPE keyword applies to shared queues only.

Figure 58: UNLOAD command set syntax

Sample UNLOAD command sets

Message Advisor provides sample UNLOAD command sets that can serve as templates when building your own UNLOAD command sets. The MAQSAMP library contains the sample command sets.

Table 289 on page 285 identifies the library members that contain the sample UNLOAD command sets. MAQSAMP also contains an index in member QMR@INDX that lists all sample command sets provided with Message Advisor.
Table 289: Sample UNLOAD command sets

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMR@UNL1</td>
<td>unloads all messages from a single destination</td>
</tr>
<tr>
<td>QMR@UNL2</td>
<td>unloads messages from multiple destinations</td>
</tr>
</tbody>
</table>

**Note**

Many of the samples include field-level prompts (IMSID=?). When you execute a command set through the Message Advisor Interactive System Productivity Facility (ISPF) interface, you are prompted to provide a valid value for any field that contains a prompt.

If you execute a command set outside of the Message Advisor ISPF interface, you are not prompted to replace ?. Therefore, you must replace all ? with a valid value before submitting the request.

The following sample command sets are similar, but not identical to, the samples provided in MAQSAMP.

**Figure 59 on page 285** shows a command set for unloading all messages in all queues from a specific destination. MAQSAMP(QMR@UNL1) contains a command set similar to **Figure 59 on page 285**.

The command set shown in **Figure 59 on page 285** unloads all messages from destination $R61PP01. Because the command set does not include an OUTPUT statement, Message Advisor unloads the messages to the default unload data set.

**Figure 59: Command set: Unload from one destination**

```
UNLOAD IMSID=R61P
  SELECT DESTINATION=$R61PP01
END
```

**Figure 60 on page 285** shows a command set for unloading all messages from multiple destinations using the FORCE keyword. MAQSAMP(QMR@UNL2) contains a sample command set similar to **Figure 60 on page 285**.

The command set shown in **Figure 60 on page 285** unloads all messages from MSNAME destinations (DESTYPE=MSN) and from destinations beginning with $R61P (DESTINATION=$R61P*). Because the command set does not include an OUTPUT statement, Message Advisor unloads the messages to the default unload data set.

**Figure 60: Command set: Unload from multiple destinations**

```
UNLOAD IMSID=R61P,FORCE=YES
  SELECT DESTINATION=$R61P*
  SELECT DESTYPE=MSN
END
```
For more information about building and executing an UNLOAD command set, see the Message Advisor for IMS User Guide.

UNLOAD command set description

The UNLOAD command set consists of the following items:

- UNLOAD primary command with associated keywords and parameters
- subcommands with associated keywords and parameters
- END command

Note
Keyword and parameter descriptions for the primary command are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

Keyword and parameter descriptions for the subcommands are located in the following chapters:

- “INTERVAL subcommand” on page 293
- “OUTPUT, EXTRACT, and SCRAP subcommands” on page 303
- “SELECT and REJECT subcommands” on page 313

Note
Before you can specify the UNLoad command, IMS must be active.

CONVersations keyword

CONVersations is an optional keyword for the UNLoad command. The CONVersations keyword provides the default value for the CONVID keyword for the SELECT and REJECT subcommands and can be overridden for specific selections.

Consider the following when specifying the CONVersations keyword:
Message Advisor unloads conversational messages only when a similar DEQueue command would cause the messages to be dequeued. Messages are always unloaded for dynamic terminals when the associated user is not active. For static terminals and active dynamic terminals, the conversational message will be unloaded only if the FORCE keyword would allow the user to be stopped. If stopping is not allowed, an error status code is issued for the conversation.

Conversational messages will often be found on the backup queue for LTERMs. The default value for the Queue keyword for SELECT and REJECT subcommands for the UNLoad command does not include the backup queue or the suspend queue. If conversational messages on these queues are to be included in processing, you must specify Q=ALL, Q=B, or Q=S in addition to CONV=ALL or CONV=ONLY.

IMS will exit a conversation when the associated node or user is stopped and restarted (even when the IMS /EXIT command is not issued). In some cases, held conversations might also be exited. Because the FORCE keyword indicates whether Message Advisor is allowed to stop a destination, special considerations occur when an active conversational message exists but is not selected or is rejected.

FORCE=RESETALL and FORCE=STOPAll will stop destinations without checking to see if conversations exist (even for the UNLoad command). This action might cause active conversations to be exited.

FORCE=YES will allow all messages to be unloaded.

FORCE=NO will cause an error status code to be issued if an active conversation exists and the destination is not stopped.

FORCE=RESET, FORCE=RESETACT, and FORCE=STOP will process messages based on the DESTYPE keyword. Although the UNLoad command will not stop the destination, it will process messages as if the destination had been stopped (similar to DEQueue processing). When a DESTYPE is used that allows multiple types to be processed (such as DESTYPE=ALL), all selected messages will be unloaded. If a specific DESTYPE value is used, an error status code will be issued.

For example, if both an active and held conversation exist for a static LTERM and a command set is created to unload the held conversation, DESTYPE=HELD will cause an error status code to be issued because stopping and restarting the user would also exit the active conversation. Any other DESTYPE value allows unload processing.

Another example involves a dynamic LTERM that has an active conversational message queued to a transaction and has other non-conversational messages queued to the LTERM. Using DESTYPE=LTERM to unload the non-conversational messages will cause an error status code to be issued. Using DESTYPE=ALL allows unload processing.

Table 290 on page 288 lists the CONVersations keyword parameters.
Table 290: CONVersations keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Message Advisor does not unload conversations. Non-conversational messages are unloaded. This value is the default value.</td>
</tr>
<tr>
<td>All/Yes</td>
<td>Message Advisor unloads both conversational and non-conversational messages for the selected destinations.</td>
</tr>
<tr>
<td>ONLY</td>
<td>Message Advisor unloads conversations for the selected destinations, but does not unload non-conversational messages.</td>
</tr>
</tbody>
</table>

FORCE keyword

FORCE is an optional keyword for the UNLoad command.

Consider the following when specifying the FORCE keyword:

- Message Advisor attempts to select the same messages that would be selected by a DEQueue command with the same FORCE keyword. The FORCE actions that occur for DEQueue do not always occur for UNLoad.

- FORCE applies to LTERMs and conversational SMBs only. Message Advisor always processes non-conversational SMBs, regardless of the FORCE setting.

- For conversational SMBs, FORCE indicates how Message Advisor handles users in conversation with the SMB. FORCE=No and FORCE=Reset are the same and indicate that Message Advisor does not unload messages for active users. Message Advisor only unloads conversational messages for inactive users. Any other FORCE value indicates that Message Advisor unloads messages whether the user is active or not.

Table 291 on page 288 lists the FORCE keyword parameters. This table describes how the FORCE keyword parameters apply to LTERMs. For more information on the effect of the FORCE command on LTERMs that are in conversation, see “CONVersations keyword” on page 286.

Table 291: FORCE keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset</td>
<td>Message Advisor only unloads messages for inactive LTERMS. This value is the default value. For additional considerations for LTERMs in conversation, see “CONVersations keyword” on page 286.</td>
</tr>
<tr>
<td>No</td>
<td>Message Advisor only unloads messages for LTERMs that are already stopped.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RESETACTive</td>
<td>Message Advisor unloads messages for both active and inactive LTERMs. For additional considerations for LTERMs in conversation, see “CONVersations keyword” on page 286.</td>
</tr>
<tr>
<td>RESETALL</td>
<td>Message Advisor stops and restarts all LTERMs that match SELECT and REJECT subcommands, even if:</td>
</tr>
<tr>
<td></td>
<td>■ the LTERMs have no messages to unload</td>
</tr>
<tr>
<td></td>
<td>■ the LTERM is active</td>
</tr>
<tr>
<td></td>
<td>■ the LTERM is in conversation</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> IMS cancels active conversations when an LTERM is restarted.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Message Advisor processes FORCE=RESETALL even if it does not unload messages. You can use FORCE=RESETALL with the UNLoad command or with DEQueue MODE=VALidate to stop LTERMs with no messages queued. FORCE=RESETALL also lists all destinations that match (even those without messages).</td>
</tr>
<tr>
<td>Stop</td>
<td>Message Advisor unloads messages for both active and inactive LTERMs. For additional considerations for LTERMs in conversation, see “CONVersations keyword” on page 286.</td>
</tr>
<tr>
<td>STOPAll</td>
<td>Message Advisor stops all LTERMs that match SELECT and REJECT subcommands, even if:</td>
</tr>
<tr>
<td></td>
<td>■ the LTERMs have no messages to unload</td>
</tr>
<tr>
<td></td>
<td>■ the LTERM is active</td>
</tr>
<tr>
<td></td>
<td>■ the LTERM is in conversation</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> IMS cancels active conversations when an LTERM is restarted.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Message Advisor processes FORCE=STOPAll even if it does not unload messages. You can use FORCE=STOPAll with the UNLoad command and with DEQueue MODE=VALidate to stop LTERMs with no messages queued. FORCE=STOPAll lists all destinations that match (even those without messages).</td>
</tr>
<tr>
<td>Yes</td>
<td>Message Advisor unloads messages for all LTERMs</td>
</tr>
</tbody>
</table>

**IMSid keyword**

IMSid is a required keyword for the UNLoad command.

*Table 292 on page 290 lists the IMSid keyword parameter.*
Table 292: IMSid keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| cccc      | the IMS system to which this UNLoad command set applies  
            **Note:** cccc is a value that you specify. |

RATE keyword

RATE is an optional keyword for the UNLoad command.

Table 293 on page 290 lists the RATE keyword parameter.

Table 293: RATE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| nnnnnnnnn | the number of selected destinations processed per minute  
            **Note:** nnnnnnnnn is a value that you specify. |

REPORT/PRINT keyword

REPORT is an optional keyword for the UNLoad command.

You can use PRINT as an alias for REPORT.

Table 294 on page 290 lists the REPORT keyword parameters.

Table 294: REPORT keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>prints a summary and destination report that contains information regarding each message DRRN that Message Advisor selected and unloaded</td>
</tr>
<tr>
<td>DESTination</td>
<td>prints a destination report that contains a line for each queue from which messages were selected. Separate lines are always printed for each conversational message selected.</td>
</tr>
<tr>
<td>DRRN</td>
<td>prints a destination report that contains a line for each message DRRN that Message Advisor selected</td>
</tr>
<tr>
<td>None</td>
<td>Message Advisor does not print report output</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>SUMMary</td>
<td>prints a summary of messages unloaded, total destinations in error, and total destinations selected</td>
</tr>
</tbody>
</table>

**SCOpe keyword**

SCOpe is an optional keyword for the UNLoad command and limits unload processing to destinations that match the specified scope. You can specify a list of values for the SCOpe keyword.

*Note*

The SCOpe keyword applies to shared queues only.

Table 295 on page 291 lists the SCOpe keyword parameters. Figure 61 on page 292 illustrates the queues that are processed for each parameter.

**Table 295: SCOpe keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLQ</td>
<td>includes both local and shared queues in unload processing</td>
</tr>
</tbody>
</table>
| LOCALQ    | limits unload processing to local queues  
  *Note:* You can use LCLQ as an alias for LOCALQ. |
| OVERFLOWQ | limits unload processing to the overflow shared queues  
  *Note:* You can use OVFQ as an alias for OVERFLOWQ. |
| PRIMARYQ  | limits unload processing to the primary shared queues  
  *Note:* You can use PRIQ as an alias for PRIMARYQ. |
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHAREDQ</td>
<td>limits unload processing to shared queues (both primary and overflow)</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can use SHRQ as an alias for SHAREDQ.</td>
</tr>
<tr>
<td></td>
<td>SHAREDQ is mutually exclusive with PRIMARYQ and OVERFLOWQ.</td>
</tr>
</tbody>
</table>

**Figure 61: SCOPE keyword parameters and associated queues processed**

```plaintext
SCOPE=

ALLQ

SHAREDQ
SHRQ

PRIMARYQ
PRIQ

OVERFLOWQ
OVFQ

LOCALQ
LCLQ
```
INTERVAL subcommand

This chapter describes the Message Advisor for IMS INTERVAL subcommand. The INTERVAL subcommand allows you to select messages for processing according to date and time ranges. You can include the INTERVAL subcommand with the following command sets:

- DEQUEUE
- DISPLAY
- REQUEUE
- UNLOAD

Overview

You can include the INTERVAL subcommand with the DEQUEUE, DISPLAY, REQUEUE, and UNLOAD command sets to process messages according to date and time ranges. The subcommand is not required for any of the command sets.

*Note*

For the DEQUEUE command in a shared queues environment, the INTERVAL subcommand is only valid with `DEQUEUE MODE=UNLOAD_DEQUEUE`.

The INTERVAL subcommand’s primary components are the START/STOP keyword pair and the NEWER_THAN/OLDER_THAN keyword pair. The START/STOP keyword pair allows you to select messages according to a fixed time stamp window, and the NEWER_THAN/OLDER_THAN keyword pair allows you to select messages according to a time stamp window that moves with the current system date. Consider the following when specifying the START/STOP and NEWER_THAN/OLDER_THAN keyword pairs:

- The START keyword is mutually exclusive with the NEWER_THAN keyword, but can be used with the OLDER_THAN keyword.
The STOP keyword is mutually exclusive with the OLDER_THAN keyword, but can be used with the NEWER_THAN keyword.

You can include more than one INTERVAL subcommand to specify multiple intervals.

Note
All of the INTERVAL subcommand keywords are also valid on SELECT and REJECT subcommands. However, the SELECT and REJECT subcommands will not be used to select SLDS for the REQUEUE command. You must use INTERVAL subcommands to select which SLDS to process. For more information on the SELECT and REJECT subcommands, see “SELECT and REJECT subcommands” on page 313.

START and STOP Keywords

Use the START and STOP keyword pair to specify a beginning and ending time stamp by which to select messages for processing. Message Advisor selects all messages with a time stamp that is the same as or occurs after the START keyword value. In other words, the time stamps must be greater than or equal to ($\geq$) the START keyword value, and less than or equal to ($\leq$) the STOP keyword value:

START $\geq$ time stamp $\leq$ STOP

When the START keyword value equals the STOP keyword value, Message Advisor selects only those messages with exactly that time stamp.

You can also use the START and STOP keywords independently. Use the START keyword alone to select all messages with a time stamp that occurs after a specific time, and use the STOP keyword alone to select all messages with a time stamp that occurs before a specific time. If you specify both keywords on the same INTERVAL subcommand, the START time must be the same as or occur before the STOP time.
Example

Consider a data set with records for a specific date, 01288:

- If `START=01288/0907480` and `STOP=01288/0907480`, Message Advisor returns all messages with that exact time stamp.

- If `START=01288/0907480` and `STOP=01289/0907480` (the next day), Message Advisor returns all messages from 9:07:48 on the first day until 9:07:48 on the next day.

- If `START=01287/0907480` and `STOP` is not used, Message Advisor returns all messages after 9:07:48 on that day, including messages with that exact time stamp and including all messages from subsequent days.

Time stamp format

Message Advisor time stamps conform to the Data Base Recovery Control (DBRC) method. The time stamp syntax is `.yyyy.ddd.hh.mm.ss.t.<offset>`. Consider the following when specifying time stamps:

- You can remove each `.`, or you can replace each `.`, with a delimiter. You can use any character except `+`, `−`, a quote, alphanumeric characters, or national characters as the delimiter.

- If a time stamp includes blanks or commas as delimiters, you must enclose the entire time stamp in quotes.

- You can specify `yyyy` as a four-digit year or as a two-digit year. A two-digit year is assumed to exist in a sliding window of -70 through +29 from the current year.

- `ddd` is the Julian day of the year. You must specify `ddd` as three digits, using leading zeroes if necessary.

- `hh`, `mm`, `ss`, and `t` are hours, minutes, seconds, and tenths of seconds, respectively. You must specify hours, minutes, and seconds as two digits each, using leading zeroes if necessary.

- Hours must be expressed in 24-hour time, as a value between 00 and 24.

- All time stamp parts following `ddd` are optional and can be truncated. Message Advisor treats any truncated parts as if you specified zeroes for those parts. For the `STOP` keyword, Message Advisor treats any truncated parts as if you specified 23:59:59. For example, if you specify `START=2020.123, STOP=2020.123`, all messages for that date are included.
Note

Some time stamp specifications can be ambiguous. For example, Message Advisor could interpret ‘202012315’ as ‘2020.123/15:00:00’ or ‘20.201/23:15:00’. Use delimiters to avoid ambiguity. Message Advisor generates a syntax error for ambiguous time stamps.

Time zone offset

The time zone offset is optional. The offset consists of a plus or minus sign followed by a number of hours (one or two digits), optionally followed by a colon, optionally followed by a number of minutes. Consider the following when specifying time zone offsets:

- If you include a colon after the number of hours, you must specify the minutes. The only valid minute values are 00, 15, 30, and 45.
- When added to Universal Time Coordinated (UTC), the offset must correspond to the time specified (Message Advisor subtracts the offset from the time specified to obtain the UTC). For example, an offset of +0 indicates that the specified time is GMT, while an offset of -8 indicates that the specified time is Pacific Standard Time.

Time zone label

If a time zone label table exists in the RECON data set, you can use a time zone label instead of a numeric offset. A time zone label consists of one to five alphanumeric characters that do not start with a number.

When you use a time zone label, Message Advisor must access the RECON data set to obtain the numeric offset. For this reason, using a numeric offset is faster.

Message Advisor always considers a time stamp without an offset or label to be current local time.

WARNING

Message Advisor ignores any default TIMEZIN specification in the RECON data set. Be careful when specifying times immediately following a time change. You must use an offset if the time stamp occurs before the time change.

NEWER_THAN and OLDER_THAN keywords

You can use the NEWER_THAN keyword in place of the START keyword to select all messages that occur after a specific day. For example, NEWER_THAN=0 selects only
messages with today’s time stamp, NEWER_THAN=1 includes messages with yesterday’s time stamp, etc. In other words, when you specify the NEWER_THAN keyword, Message Advisor selects messages that meet the following condition:

\[
\text{time stamp date } \geq \text{ current date } - \text{NEWER_THAN}
\]

You can use the OLDER_THAN keyword in place of the STOP keyword to select all messages that occur before a specific day. For example, OLDER_THAN=0 selects only messages with time stamps before today, OLDER_THAN=1 excludes messages with today’s time stamp and messages with yesterday’s time stamp, etc. In other words, when you specify the OLDER_THAN keyword, Message Advisor selects messages that meet the following condition:

\[
\text{time stamp date } < \text{ current date } - \text{OLDER_THAN}
\]

If you specify both the NEWER_THAN and OLDER_THAN keywords on the same INTERVAL subcommand, the NEWER_THAN keyword value must be greater than the OLDER_THAN keyword value. When you specify both keywords, Message Advisor selects messages that meet the following condition:

\[
\text{current date } - \text{NEWER_THAN } \leq \text{time stamp date } < \text{current date } - \text{OLDER_THAN}
\]

Figure 62 on page 298 uses a number line to show how the NEWER_THAN and OLDER_THAN keywords work.


**Note**

NEWER_THAN=0 is valid and selects messages with today’s date. OLDER_THAN=0 excludes messages with today’s date. You cannot specify NEWER_THAN=0 and OLDER_THAN=0 on the same INTERVAL subcommand.

---

**INTERVAL subcommand syntax**

The following figure shows the INTERVAL subcommand syntax.

---

**Figure 63: INTERVAL subcommand syntax**

```
INTERVAL
  LABEL=cccccccc
  .NEWER_than=rnm
  .OLDER_than=rnn
  .STArt=yyyy:ddd.hh:mm:ss.l<offset>
  .STOP=yyyy:ddd.hh:mm:ss.l<offset>
```
INTERVAL subcommand description

The INTERVAL subcommand consists of the following items:

- INTERVAL primary command
- LABEL keyword
- START/STOP and NEWER_THAN/OLDER_THAN keyword pairs

Note
Keyword and parameter descriptions are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

LABEL keyword

Use the LABEL keyword to provide a name (label) for the INTerval subcommand statement.

Message Advisor uses the LABEL keyword value to identify the statement in reports.

Table 296 on page 299 lists the LABEL keyword parameter.

Table 296: LABEL keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT00001</td>
<td>the name of this INTerval subcommand statement</td>
</tr>
<tr>
<td>cccccccc up to 8 characters</td>
<td>Note: The default label is INT00001 for the first statement. The number increments by one for each subsequent statement. ccccccc is a value that you specify.</td>
</tr>
</tbody>
</table>

NEWER_than keyword

Use the NEWER_than keyword to select all messages that occur after a specific day. The NEWER_than keyword excludes the STArt keyword.
Table 297 on page 300 lists the NEWER_than keyword parameter.

**Table 297: NEWER_than keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| $nnn$ range 0-365 | selects messages that occur within $nnn$ days of the current system date  

*Note: $nnn$ is a value that you specify.*

**OLDER_than keyword**

Use the OLDER_than keyword to select all messages that occur before a specific day. The OLDER_than keyword excludes the STOP keyword.

Table 298 on page 300 lists the OLDER_than keyword parameter.

**Table 298: OLDER_than keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| $nnn$ range 0-365 | selects messages that are more than $nnn$ days older than the current system date  

*Note: $nnn$ is a value that you specify.*

**STArt keyword**

Use the STArt keyword to select all messages with a time stamp that occurs after a specific time. Use the STArt keyword in combination with the STOP keyword to specify a beginning and ending time stamp by which to select messages for processing. The STArt keyword excludes the NEWER_than keyword.

Table 299 on page 300 lists the STArt keyword parameter.

**Table 299: STArt keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| $ yyyy.ddd.hh.mm.ss.t<offset>$ | selects messages with a time stamp that occurs after a specific time  

*Note: $ yyyy.ddd.hh.mm.ss.t<offset>$ is a value that you specify. See “Time stamp format” on page 295 for additional information.*
STOP keyword

Use the STOP keyword to select all messages with a time stamp that occurs before a specific time. Use the STArt keyword in combination with the STOP keyword to specify a beginning and ending time stamp by which to select messages for processing. The STOP keyword excludes the OLDER_than keyword.

Table 300 on page 301 lists the STOP keyword parameter.

Table 300: STOP keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.yyyy.ddd.hh.mm.ss.t&lt;offset&gt;</td>
<td>selects messages with a time stamp that occurs before a specific time</td>
</tr>
<tr>
<td><strong>Note:</strong> .yyyy.ddd.hh.mm.ss.t&lt;offset&gt; is a value that you specify. See “Time stamp format” on page 295.</td>
<td></td>
</tr>
</tbody>
</table>
OUTPUT, EXTRACT, and SCRAP subcommands

This chapter describes the Message Advisor for IMS OUTPUT, EXTRACT, and SCRAP subcommands. The OUTPUT subcommand applies to the DEQUEUE and UNLOAD command sets. The EXTRACT and SCRAP subcommands apply to the REQUEUE command set only.

Overview

The OUTPUT subcommand lets you define the data set to which Message Advisor unloads messages. You can include the OUTPUT subcommand with the DEQUEUE and UNLOAD command sets. The subcommand is not required for either of the command sets.

The EXTRACT and SCRAP subcommands apply to the REQUEUE command set only. Neither subcommand is required. The EXTRACT subcommand allows you to define the data set to which Message Advisor extracts messages. The SCRAP subcommand allows you to define the data set in which Message Advisor stores the messages it cannot requeue because of an error condition.

Syntax, keywords, and parameters are identical for the OUTPUT, EXTRACT, and SCRAP subcommands.

OUTPUT/EXTRACT/SCRAP subcommand syntax

The following figure shows the syntax for the OUTPUT, EXTRACT, and SCRAP subcommands.
The OUTPUT, EXTRACT, and SCRAP subcommands consist of the following items:

- Primary command

- Keywords and parameters

**Note**

Keyword and parameter descriptions are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.
APPEND keyword

Use the APPEND keyword to indicate whether Message Advisor replaces existing data in the data set.

If you specify DISP=OLD, APPEND defaults to No. If you specify DISP=MOD, APPEND defaults to Yes. If you specify DISP=NEW, you cannot also specify the APPEND keyword. The other valid DISP and APPEND keyword combinations are as follows:

- DISP=OLD, APPEND=Yes
  This keyword combination extends the existing data set. The keywords fail if no data set exists.

- DISP=MOD, APPEND=No
  This keyword combination overwrites the existing data set, or creates a new data set if one does not exist.

  **Note**
  DISP=MOD, APPEND=No is not allowed with GDG data sets.

Table 301 on page 305 lists the APPEND keyword parameters.

Table 301: APPEND keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Message Advisor replaces existing data in the data set</td>
</tr>
<tr>
<td>Yes</td>
<td>Message Advisor appends new data to existing data in the data set</td>
</tr>
</tbody>
</table>

BLKSIze keyword

The BLKSIze keyword indicates the block size for new data sets, or for existing data sets for which the block size is not defined. Message Advisor ignores the BLKSIze keyword for existing data sets with valid block sizes.

Table 302 on page 306 lists the BLKSIze keyword parameter.
Table 302: BLKSzize keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23476</td>
<td>the block size that Message Advisor uses for new data sets, or for existing data sets for which the block size is not defined</td>
</tr>
<tr>
<td>nnnn</td>
<td>range 84-32760</td>
</tr>
<tr>
<td>Note: nnnn is a value that you specify. The value must be at least as large as the largest possible IMS message, plus 16. Check the LGMSG data set record length to determine the largest possible message.</td>
<td></td>
</tr>
</tbody>
</table>

CYLS_PRIM keyword

The CYLS_PRIM keyword indicates the number of primary cylinders that Message Advisor allocates for new data sets. The CYLS_PRIM keyword is required for new data sets, and is not allowed if you specify DISP=OLD.

Table 303 on page 306 lists the CYLS_PRIM keyword parameter.

Table 303: CYLS_PRIM keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnnnnn</td>
<td>the number of primary cylinders that Message Advisor allocates for new data sets</td>
</tr>
<tr>
<td>Note: nnnnnnnnn is a value that you specify.</td>
<td></td>
</tr>
</tbody>
</table>

CYLS_SEC keyword

The CYLS_SEC keyword indicates the number of secondary cylinders that Message Advisor allocates for new data sets. The CYLS_SEC keyword is not allowed if you specify DISP=OLD, and is optional in other cases.

Table 304 on page 306 lists the CYLS_SEC keyword parameter.

Table 304: CYLS_SEC keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnnnnn</td>
<td>the number of secondary cylinders that Message Advisor allocates for new data sets</td>
</tr>
<tr>
<td>Note: nnnnnnnnn is a value that you specify.</td>
<td></td>
</tr>
</tbody>
</table>
**DATACLAS keyword**

The DATACLAS keyword indicates the SMS data class for new data sets. Message Advisor ignores the DATACLAS keyword for existing data sets. The DATACLAS keyword is not allowed if you specify DISP=OLD.

Table 305 on page 307 lists the DATACLAS keyword parameter.

**Table 305: DATACLAS keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccc</td>
<td>the SMS data class for new data sets</td>
</tr>
<tr>
<td></td>
<td>Note: cccccc is a value that you specify.</td>
</tr>
</tbody>
</table>

**DDName keyword**

The DDName keyword indicates that Message Advisor is to use a pre-allocated data set. You cannot use the DDName keyword with any other keywords, and it is only allowed with a batch server.

You can use File as an alias for DDName.

Table 306 on page 307 lists the DDName keyword parameter.

**Table 306: DDName keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cccccc</td>
<td>indicates that Message Advisor is to use a pre-allocated data set</td>
</tr>
<tr>
<td></td>
<td>Note: cccccc is a value that you specify.</td>
</tr>
</tbody>
</table>

**DISP keyword**

The DISP keyword indicates the disposition of the output file.

Table 307 on page 308 lists the DISP keyword parameters.
Table 307: DISP keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLD</td>
<td>indicates that the data set already exists, and Message Advisor will not create a new data set</td>
</tr>
<tr>
<td>MOD</td>
<td>indicates that the data set may already exist, but Message Advisor will create the data set if it does not already exist</td>
</tr>
<tr>
<td>NEW</td>
<td>indicates that the data set does not already exist, and Message Advisor will create it</td>
</tr>
</tbody>
</table>

**DSName keyword**

The DSName keyword indicates the name that identifies the data set.

You can use DAtaset as an alias for DSName.

The following considerations apply when specifying a value for the DSName keyword:

- You can include symbolic keywords when specifying a value for the DSName keyword. A symbolic keyword begins with a percent sign (%) and optionally ends with a percent sign (%). Message Advisor substitutes the appropriate value when it allocates the data set. See Table 308 on page 309 for descriptions of the symbolic keywords allowed with the DSName keyword.

  A non-alphanumeric character must follow a symbolic keyword. If a percent sign (%) which is not part of another symbolic keyword follows the keyword, the trailing % is removed and the remaining characters are concatenated to the symbolic value.

- You can include system symbols when specifying a value for the DSName keyword. A system symbol begins with an ampersand (&) and optionally ends with a period (.). The system will substitute a value for valid symbols. You will receive an allocation error if you use an undefined system symbol or if the substitution results in an invalid data set name. See the *JCL Reference Manual* for more information on system symbols.

- You can specify a generation data group (GDG) for the DSName keyword value. The following considerations apply when specifying a GDG:

  — You can specify a GDG by appending (0), (+n), or (-n) to the data set name.

  — The relative number is in relation to the current job, such that the same relative number always refers to the same data set within a job.
— You must pre-define the GDG before you can specify it for the DSName keyword.

— You can include symbolic keywords when specifying a GDG. However, since allocation errors will result if you do not pre-define the GDG and some of the symbolic keywords make pre-defining the GDG difficult, %IMSID and %USER are the only useful symbolic keywords.

— You must provide a model DSCB by either allocating an uncataloged data set on the catalog volume, or by specifying a MODEL_dsname keyword value that points to an existing catalogued data set.

— You cannot specify the combination DISP=MOD, APPEND=No when specifying a GDG.

You can specify a member name on the DSName keyword for existing data sets; however, you cannot specify a member with DISP=NEW.

See “Command syntax considerations” on page 37 for additional items to consider when specifying the DSName keyword value.

Table 308 on page 309 lists the symbolic keywords that you can use in the DSName keyword parameters.

**Table 308: DSName symbolic keywords**

<table>
<thead>
<tr>
<th>Symbolic Keyword</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%DATE</td>
<td>yyddd</td>
<td>the day of the year</td>
</tr>
<tr>
<td>%DATE4</td>
<td>yyyyddd</td>
<td>the day of the year</td>
</tr>
<tr>
<td>%IMSID</td>
<td>cccc</td>
<td>the IMS ID of the active system</td>
</tr>
<tr>
<td>%TIME</td>
<td>hhmmss</td>
<td>the current time of day</td>
</tr>
<tr>
<td>%TIMET</td>
<td>hhmmssst</td>
<td>the current time of day</td>
</tr>
<tr>
<td>%USER</td>
<td>cccccccc</td>
<td>the user ID associated with the batch job, or the TSO user</td>
</tr>
</tbody>
</table>

**MGMTCLAS keyword**

The MGMTCLAS keyword indicates the SMS management class for new data sets. Message Advisor ignores the MGMTCLAS keyword for existing data sets. The MGMTCLAS keyword is not allowed if you specify DISP=OLD.

Table 309 on page 310 lists the MGMTCLAS keyword parameter.
Table 309: MGMTCLAS keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| cccccccc  | the SMS management class for new data sets  
            | Note: cccccccc is a value that you specify. |

**MODEL_dsname keyword**

The MODEL_dsname keyword indicates the name of an existing catalogued data set to be used for model DSCB information. The following considerations apply to the MODEL_dsname keyword:

- The MODEL_dsname keyword may be necessary if you specify a GDG for the DSName keyword.
- Message Advisor ignores the MODEL_dsname keyword for existing data sets.
- The MODEL_dsname keyword is not allowed if you specify `DISP=OLD`.
- You can include the symbolic keywords `%IMSID` and `%USER` when specifying a value for the MODEL_dsname keyword.

Table 310 on page 310 lists the MODEL_dsname keyword parameter.

Table 310: MODEL_dsname keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| dsname    | the name of an existing catalogued data set to be used for model DSCB information  
            | Note: dsname is a value that you specify. |

**STORCLAS keyword**

The STORCLAS keyword indicates the SMS storage class for new data sets. Message Advisor ignores the STORCLAS keyword for existing data sets. The STORCLAS keyword is not allowed if you specify `DISP=OLD`.

The following table lists the STORCLAS keyword parameter.
Table 311: STORCLAS keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| cccccccc  | the SMS storage class for new data sets  
Note: cccccccc is a value that you specify. |

UNIT keyword

The UNIT keyword indicates the generic unit name that Message Advisor uses when allocating the data set.

Table 312 on page 311 lists the UNIT keyword parameter.

Table 312: UNIT keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| cccccccc  | the generic unit name that Message Advisor uses when allocating the data set  
Note: cccccccc is a value that you specify. |

VOLser/VOLume keyword

The VOLser keyword indicates the volume serial number that Message Advisor uses when allocating new data sets, or the volume serial number for an existing data set that is not catalogued. The VOLser keyword is optional; however, if you specify the VOLser keyword, you must also specify the UNIT keyword.

You can use VOLume as an alias for VOLser.

Table 313 on page 311 lists the VOLser keyword parameter.

Table 313: VOLser keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| (ccccccc,...) | the volume serial number that Message Advisor uses when allocating new data sets, or the volume serial number for an existing data set that is not catalogued  
You can specify multiple volume serial numbers for a multi-volume data set.  
Note: (ccccccc,...) is a value that you specify. |
SELECT and REJECT subcommands

This chapter describes the Message Advisor for IMS SELECT and REJECT subcommands. You can include the SELECT and REJECT subcommands with the following command sets:

- DEQUEUE
- DISPLAY
- REQUEUE
- UNLOAD

Overview

Message Advisor’s SELECT and REJECT subcommands let you process messages on the IMS message queues according to criteria that you specify. The SELECT subcommand lets you select specific messages for processing. The REJECT subcommand lets you exclude specific messages from processing. You can include both subcommands with the following command sets:

- DEQUEUE
- DISPLAY
- REQUEUE
- UNLOAD

All of the SELECT and REJECT subcommand keywords are also valid on the following subcommands:

- CHANGE (see “CHANGE subcommand” on page 254)
- GROUP (see “GROUP subcommand” on page 270)
The table in “SELECT/REJECT/CHANGE/GROUP subcommand keyword groups” on page 318 lists the keywords that are common to the SELECT, REJECT, CHANGE, and GROUP subcommands.

The SELECT and REJECT subcommands are not required for the DISPLAY, REQUEUE, and UNLOAD command sets. One of the SELECT, REJECT, or INTERVAL subcommands is required for the DEQUEUE command set.

Syntax, keywords, and parameters are identical for the SELECT and REJECT subcommands, except for minor differences as noted in the individual keyword descriptions.

Message Advisor allows the following masking characters (m) for most selection keywords:

- ? replaces exactly one character in a multicharacter string.
- * replaces any number of characters.
- Masking characters can be embedded in the string.

**SELECT/REJECT subcommand syntax**

The following figures show the syntax for the SELECT and REJECT subcommands.

---

**Note**

The following values apply to shared queues only:

- DESTYPE=UNDEFINED
- ENTRYID keyword
- KEY keyword
- QTYPE keyword
Figure 65: SELECT/REJECT subcommand syntax (part 1 of 3)
Figure 66: SELECT/REJECT subcommand syntax (part 2 of 3)
Figure 67: SELECT/REJECT subcommand syntax (part 3 of 3)
SELECT/REJECT/CHANGE/GROUP subcommand keyword groups

The following table lists the keywords that are common to the SELECT, REJECT, CHANGE, and GROUP subcommands.

Table 314: Keywords common to SELECT/REJECT/CHANGE/GROUP subcommands

<table>
<thead>
<tr>
<th>Keyword Group</th>
<th>Description</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>destination-related</td>
<td>select which destinations to process</td>
<td>AFFINITY, CONV ID, DBD, DEADQ, DESTINATION, DESTYPE, DYNAMIC, HELD, KEY, LTERM, LUNAME, MSNAME, PSB, QTYPE, RSMB, SMB, STATIC, TMEMBER, TPIPE, TPNME, TRANSACTION, VSPCNT</td>
</tr>
<tr>
<td>keywords</td>
<td>Note: Most of the destination-related keywords (except AFFINITY) are mutually exclusive. In general, you should only specify one destination-related keyword; however, the following combinations are allowed:</td>
<td></td>
</tr>
<tr>
<td>destination-related</td>
<td>■ DESTINATION and DESTYPE</td>
<td></td>
</tr>
<tr>
<td>keywords</td>
<td>■ DESTINATION and QTYPE</td>
<td></td>
</tr>
<tr>
<td>queue-related keywords</td>
<td>select which queues to process for the selected destinations</td>
<td>QUEUE, REGION, REGION_ID</td>
</tr>
<tr>
<td>select which queues to process for the selected destinations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>keywords</td>
<td>The AFFINITY keyword is message-related for the REQUEUE command. The QTYPE keyword is destination-related and message-related.</td>
<td></td>
</tr>
</tbody>
</table>

Note: Most of the destination-related keywords (except AFFINITY) are mutually exclusive. In general, you should only specify one destination-related keyword; however, the following combinations are allowed:

- DESTINATION and DESTYPE
- DESTINATION and QTYPE
- TPNAME and LUNAME
- TMEMBER and TPIPE
- HELD and CONVID
- REGION and REGION_ID

The AFFINITY keyword is message-related for the REQUEUE command. The QTYPE keyword is destination-related and message-related.
<table>
<thead>
<tr>
<th>Keyword Group</th>
<th>Description</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>message-related</td>
<td>select which messages to process for the selected queues</td>
<td>DRRN a, ENTRYID a, MFSNAME, MSGTYPE, NEWER_THAN, NODENAME, OIMS, OLDER_THAN, ORIGIN, ORGQNAME, ORGQMANAGER, ORGTPPIPE, ORGTMEMBER, OTMAQ a, OTOKEN, PIMS, PTOKEN, QPOSITION a, RCNT, START, STOP, SYSID, UOWID, USERID</td>
</tr>
<tr>
<td>data-related keywords</td>
<td>select which messages to process for the selected queues</td>
<td>DATA, DATASIZE, OFFSET, SEGMENT</td>
</tr>
<tr>
<td>other keywords</td>
<td>special purpose</td>
<td>LABEL, SCRAPCODE c</td>
</tr>
</tbody>
</table>

Some of the SELECT and REJECT keywords and values are not allowed with certain commands. Table 315 on page 320 identifies the command and keyword combinations that are not allowed.
Table 315: SELECT/REJECT invalid command and keyword combinations

<table>
<thead>
<tr>
<th>Command</th>
<th>Keywords not allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEQUEUE/UNLOAD</td>
<td>REGION, REGION_ID, SCRAPCODE</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>SCRAPCODE</td>
</tr>
<tr>
<td>REQUEUE</td>
<td>CONVID, DESTYPE=HELD, DRRN, ENTRYID, HELD, KEY, OTMAQ, QPOSITION, QUEUE=SUSPEND, BACKUP, or HELD, REGION, REGION_ID</td>
</tr>
</tbody>
</table>

Consider the following when specifying SELECT and REJECT subcommand keywords:

- If you only use destination- and queue-related keywords, processing will be fastest since there is no need to examine individual messages.
- If you use message prefix-related keywords, each message prefix must be examined, so processing will be slower.
- If you use message data-related keywords, processing will be slowest, since all records for each message must be examined.

**SELECT/REJECT subcommand description**

The SELECT and REJECT subcommands consist of the following items:

- Primary command
- Keywords and parameters
Keyword and parameter descriptions are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

**AFFINITY keyword**

Use the AFFINITY keyword to select or reject messages with a specific affinity. The AFFINITY keyword allows a seven-character affinity mask. Consider the following when specifying the AFFINITY keyword:

---

*Note*
The AFFINITY keyword applies to shared queues only.

---

- You can specify a list of affinities for the AFFINITY keyword value. The list can include the special values that are described in Table 316 on page 322. For example, AFFINITY=(&NONE, IMSA) returns only destinations with no affinity or with affinity to IMSA.

- If you do not specify the AFFINITY keyword, Message Advisor returns all destinations which otherwise match, with or without an affinity.

- You can specify `AFFINITY=&NONE` or `AFFINITY=' '` to return only destinations with no affinity.

- If you specify a mask for the AFFINITY keyword value, Message Advisor only returns destinations with an affinity that matches the mask. Message Advisor does NOT return destinations without an affinity (i.e., Message Advisor treats `AFFINITY=*` the same as it treats `AFFINITY=?*`).

- Local queue destinations (on non-shared queues systems or on shared queues systems) are considered to have no affinity and will only match a blank or a SELECT subcommand without an affinity.

- AFFINITY is a message-related keyword when used on a SELECT or REJECT subcommand with the REQueue command, but is a destination-related keyword for all other commands.

- If you specify either (but not both) &EXISTS or &ORPHAN, processing occurs as follows:

  — Message Advisor performs a FNDEST CREATE for destinations that it finds on the shared queues with an affinity for the current IMS.
— If Message Advisor cannot find or create the destination, the destination matches &ORPHAN.

— If Message Advisor finds or creates the destination but the destination is not of the correct type for that queue, the destination matches &ORPHAN.

— For queue 7 and queue 9, the destination will also match &ORPHAN if the shared queues token does not match the IMS token.

— Destinations that exist or that Message Advisor can create as the correct type match &EXISTS.

The intent of the above processing is to allow finding messages on the shared queues which can no longer be processed, and are thus "orphaned."

Table 316 on page 322 lists the special parameters that are allowed for the AFFINITY keyword.

### Table 316: AFFINITY keyword special parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| `mmmmmmmm`  | matches destinations with the specified affinity  
  *Note: `mmmmmmmm` is a value that you specify. You can specify a list of values or patterns.* |
| &EXISTS     | matches only destinations with affinity to the current IMS and which CAN be created on the current IMS  
  *Note: &EXISTS is not allowed on SELECT or REJECT subcommands with the REQueue command.* |
| &IMS        | matches only destinations with an affinity for the current IMS  
  *Note: &IMS is not allowed on SELECT or REJECT subcommands with the REQueue EXTRACT command. AFFINITY=(&EXISTS,&ORPHAN) is the same as AFFINITY=&IMS.* |
| &NONE       | matches only destinations with no affinity |
| &ORPHAN     | matches only destinations with affinity to the current IMS and which CANNOT be created on the current IMS  
  *Note: AFFINITY=&ORPHAN will not necessarily find all messages that cannot be processed. Messages without affinity will never match, since there may be another IMS that can process them even if the IMS is not currently active. There may also be LTERMs that can still be created by VT or ETO, which are no longer used. You may need to specify a time interval to locate these messages and LTERMs. &ORPHAN is not allowed on SELECT or REJECT subcommands with the REQueue command.* |
CNT keyword

Use the CNT keyword to select or reject messages that are destined to local logical terminals (LTERMs) that are identified in specific communications name tables (CNTs). You can specify a list of CNTs for the CNT keyword value.

Table 317 on page 323 lists the CNT keyword parameter.

Table 317: CNT keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmm...)</td>
<td>the names that identify the LTERMs for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td>You can specify STATic, DYNAMic, or DEADQ names. You cannot specify MSNAMEs or VSPcnt names.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> mmmmmmmmm is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

COMMITMODE keyword

Use the COMMITMODE keyword to select and reject commit mode 0 and commit mode 1 messages.

Table 318 on page 323 lists the COMMITMODE keyword parameter.

Table 318: COMMITMODE keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM0</td>
<td>Commit Mode 0 (commit-then-send)</td>
</tr>
<tr>
<td>CM1</td>
<td>Commit Mode 1 (send-then-commit)</td>
</tr>
</tbody>
</table>

CONVID keyword

Use the CONVID keyword to select or reject conversational messages. You can specify keywords or a list of conversation numbers for the CONVID keyword value. You can use CONVERSATIONS or CONV as an alias for CONVID.

Table 319 on page 324 lists the CONVID keyword parameter.
Table 319: CONVID keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>selects or rejects both conversational and non-conversational messages</td>
</tr>
<tr>
<td>ANY</td>
<td>selects or rejects both conversational and non-conversational messages</td>
</tr>
<tr>
<td>CONVxxxx</td>
<td>the number that identifies the conversation that you want to select or reject for processing. Note: xxxx is a value that you specify. You can enter xxxx without the preceding CONV, and Message Advisor changes the format to CONVxxxx.</td>
</tr>
<tr>
<td>ONLY</td>
<td>selects or rejects both conversational and held conversational messages. Non-conversational messages are not selected.</td>
</tr>
<tr>
<td>NONE</td>
<td>selects or rejects only non-conversational messages</td>
</tr>
</tbody>
</table>

- You can combine multiple keywords. For example, CONVID=(NONE, ONLY) is equivalent to CONVID=ALL.

- Message Advisor ignores duplicate keywords.

- You can combine keywords with conversation numbers. The conversation numbers will restrict which conversational messages are processed as shown in the following examples:

  — CONVID=(0002) or CONVID=(ONLY, 0002) selects or rejects only conversations with an ID of 0002. Both active and held conversations are selected or rejected. You must specify DESTYPE=HELD to restrict processing to held conversations only.

  — CONVID=(NONE, 0001) selects or rejects all non-conversational messages and all conversations with an ID of 0001.

  — CONVID=(ALL, 0001) selects or rejects all non-conversational messages and all conversations with an ID of 0001.

- For shared queues systems, Message Advisor processes all conversational filtering at the message level. Message Advisor searches all destinations. A message that contains a conversation prefix and SPA is considered conversational. A message that does not contain a conversation prefix and SPA is considered non-conversational.

- For non-shared queues systems, the ONLY and NONE keywords provide additional filtering at the destination level. ONLY processes only conversational SMBs and CNTs which are in conversation. NONE processes non-conversational SMBs, but processes all CNTs because CNTs in conversation may also have non-conversational messages queued.

  A message is considered conversational if it meets all the following requirements:
contains a conversation prefix and SPA
— is queued to a conversational transaction or an LTERM in conversation

A message is considered non-conversational if it meets any of the following requirements:
— is queued to a non-conversational transaction or an LTERM not in conversation (even if the message contains a SPA)
— does not contain a SPA (even if the conversation is still active)

■ is queued to a non-conversational transaction or an LTERM not in conversation (even if the message contains a SPA)
■ does not contain a SPA (even if the conversation is still active)
■ For the DISplay command set, the default CONVID parameter is All for SELECT statements.
■ For the DEQueue and UNLoad command sets, the default CONVID parameter for SELECT statements is the value specified for CONV on the command.
■ For all command sets, the default CONVID parameter is All for REJECT statements.

CONVID=A is not an abbreviation for CONVID=ALL. CONVID=A is interpreted as CONVID=000A and processes only held conversations with an ID of 000A.

■ Most CONVID values will not be allowed with the SORT=QCT keyword on the DISPLAY TYPE=DQ command because the QCT keyword indicates that messages are not to be examined or counted (the displayed counts are taken from CNT or SMB).

**Note**

Held conversation destinations must be a combination of the user ID and conversation number. A sample DISplay command set that includes a SELECT statement using the CONVID keyword is provided below:

```
DISPLAY  TYPE=DEST_Q,IMS=XRB6
SELECT
  HELD=FRED*,
  CONVID=(0001,000F,23,CONV0003,CONV01DF)
END
```

The command set shown above would display conversations with numbers of 0001, 000F, 0023, 0003, or 01DF for any user ID starting with FRED.
You can omit CONVID and/or HELD, and Message Advisor will scan for destinations that match any specified selections.
DATA keyword

Use the DATA keyword to select or reject messages that contain a specific data string. Consider the following when specifying the DATA keyword:

- You can include pattern matching characters in the data string.
- If you do not specify either of the OFFSET or SEGMENT keywords, the string can occur in any data segment of the message, but must occur entirely in one segment.
- Message Advisor does not search the message prefix or segment header words for the data string.
- You can specify a data string that contains up to 256 characters.
- You do not need to enclose the data string in quotes if you do not use special characters and the string fits on one line. If the string contains lowercase characters, blanks, or commas, you must enclose the string in quotes; otherwise, the string is translated to uppercase.
- You can specify a hexadecimal string. For example, X'D3E84B15'.

Table 320 on page 326 lists the DATA keyword parameter.

Table 320: DATA keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cc...cc</td>
<td>the data string that identifies the messages that you want to select or reject for processing</td>
</tr>
<tr>
<td>up to 256 characters</td>
<td></td>
</tr>
<tr>
<td>Note: cc...cc is a value that you specify.</td>
<td></td>
</tr>
</tbody>
</table>

DATASIZE keyword

Use the DATASIZE keyword to select or reject messages of a specific data size, or of a data size that falls within a specific range. The data size is the sum of all data or text segments in all records for the message, including each segment header word but not including any prefix segments.

You can specify a single number or a range for the DATASIZE keyword value. A range is two numbers in parentheses, separated by a comma. For example, DATASIZE=(100,150) selects or rejects messages with a total data size between 100 and 150 bytes, inclusive. DATASIZE=100 selects or rejects messages with a data size of exactly 100 bytes. If you specify a range, you can replace the second number with an
asterisk (*) to indicate no upper limit. For example, \(\text{DATASIZE}=(1000,\ast)\) selects or rejects messages with 1000 or more bytes of data.

Table 321 on page 327 lists the \text{DATASIZE} keyword parameter.

\begin{table}[h]
\centering
\begin{tabular}{|l|l|}
\hline
\textbf{Parameter} & \textbf{Description} \\
\hline
\(\text{n}\text{n}n\text{n},\text{n}\text{n}n/\ast)\) & the data size or data size range of the messages that you want to select or reject for processing  \\
 & Specify a single \text{n}\text{n}n\text{n} value to select or reject messages of that exact data size.  \\
 & If you specify a range, you can replace the second \text{n}\text{n}n\text{n} value with * to indicate no upper limit.  \\
 & \textbf{Note:} \text{n}\text{n}n\text{n} is a value that you specify. \\
\hline
\end{tabular}
\caption{\text{DATASIZE} keyword parameter}
\end{table}

\section*{DBD keyword}

Use the \text{DBD} keyword to select or reject messages that are destined to transactions that are currently associated with program specification blocks (PSBs) that are also associated with specific database descriptions (DBDs). Consider the following when specifying the \text{DBD} keyword:

- You can specify a list of DBDs for the \text{DBD} keyword value, but patterns are not allowed.

- The \text{DBD} keyword assumes \text{DESTYpe}=\text{TRANSACTIONS} and is mutually exclusive with the \text{DESTYpe} keyword, the \text{DESTination} keyword, and any of the individual destination type keywords.

- The \text{DBD} keyword requires that the IMS system be active and running \text{DELTA PLEX}.

- Message Advisor queries the IMS system to obtain a list of transactions that are associated with the specified DBDs, and then uses the transactions for destination filtering. This allows you to filter for messages which \textit{may} have caused a database update.

\textbf{Note}

The transactions associated with a DBD may have changed since the input file was created. The \text{DBD} keyword always provides current associations, and not necessarily the associations that were in effect when the message was logged.

Table 322 on page 328 lists the \text{DBD} keyword parameter.
Table 322: DBD keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(cccccccc,...)</td>
<td>the DBDs for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td>Message Advisor selects or rejects messages destined to transactions that are currently associated with PSBs that are also associated with the specified DBDs.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> cccccccc is a value that you specify. You can specify a list of values. Patterns are not allowed.</td>
</tr>
</tbody>
</table>

DEADQ keyword

Use the DEADQ keyword to select or reject messages that are destined to dead letter queues (with Extended Terminal Option (ETO)). You can specify a list of dead letter queues for the DEADQ keyword value.

Table 323 on page 328 lists the DEADQ keyword parameter.

Table 323: DEADQ keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmmm,...)</td>
<td>the dead letter queues for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> mmmmmmmmm is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

DESTination keyword

Use the DESTination keyword to select or reject messages by destination. You can specify a list of destinations or patterns for the DESTination keyword value. Destinations that match any of the patterns will match the SELECT or REJECT statement.

**Note**

For the REQueue, DEQueue, and UNLoad command sets, system destination names (such as the master or secondary master, or names starting with DFS) will not match a pattern. To requeue, dequeue, or unload to system destinations, you must include the specific destination name in a SELECT statement.
On a CHANGE subcommand, DESTination only applies to transactions and LTERMs.

Table 324 on page 329 lists the DESTination keyword parameter.

Table 324: DESTination keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(mm...mm,...)$</td>
<td>the destinations for which you want to select or reject messages for processing. You can specify any type of destination for the DESTination keyword value. Destination names must be eight characters or less except for TPNames, which can be up to 64 characters, and TMEMBER names, which can be up to 16 characters. <strong>Note:</strong> $mm...mm$ is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

**DESType/DTYPe keyword**

Use the DESType keyword to select or reject messages by destination type. To exclude a destination type (except ALL), add NO in front of the applicable parameter. For example, \texttt{DESType=(NOAPPC,NOOTMA)} includes all destination types except Advanced Program-to-Program Communication (APPC) and open transaction manager access (OTMA) programs.

You can use DTYPe as an alias for DESType.

You can specify some of the DESType keyword parameters as individual keywords to avoid having to specify both DESTination and DESType. For example, \texttt{CNT=ABC} is the same as \texttt{DEST=ABC,DESTY=CNT}. The individual destination type keywords are mutually exclusive with each other and with the DESTination and DESType keywords. Following are the DESType keyword parameters that you can specify as individual keywords:

- CNT
- CNT-E
- CNT-V
- DEADQ
- DYNAMic
- ETO
- LTERM
- MEMBER
- MSName
- RSMB
- SMB
- STATic
- TMEMBER
- TPName
- TRANSACTION
- UNDEFINED (applies to shared queues only)
- VIRTual
- VSPcnt
- VTF

The following additional considerations apply to the DESTYpe keyword parameters:

- HELD indicates held conversational messages. HELD is not allowed with the REQueue command; it is only allowed with the DEQueue, DI Spend, and UNLoad commands.

- SYSTEM indicates a system CNT. System CNTs have one of the following characteristics:
  - name is WTOR
  - name starts with DFS
  - the CNT is marked as system (CNT3SYS or CNT3MTI bit on)
  - the CNT is logged on as system console (CTTDSYSC bit on)

**Note**
For the REQueue, DEQueue, and UNLoad command sets, system destination names (such as the master or secondary master, or names starting with DFS) will not match a pattern. To requeue, dequeue, or unload to system destinations, you must include the specific destination name in a SEl ect statement.
On a shared queues system, DESTYpe (other than ALL) implies an affinity to the current IMS or no affinity. Messages with other affinities will not match, regardless of the AFFINITY keyword. Only destinations which exist on the current IMS will match unless you specify UNDEFINED.

Table 325 on page 331 lists the DESTYpe keyword parameters. Figure 68 on page 332 illustrates the destination types that are processed for each parameter.

### Table 325: DESTYpe keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>selects or rejects messages that are destined to all destination types</td>
</tr>
<tr>
<td>BMP</td>
<td>selects or rejects messages that are destined to batch message processing (BMP) programs</td>
</tr>
<tr>
<td>CNT</td>
<td>selects or rejects messages that are destined to local terminals</td>
</tr>
<tr>
<td>DEADQ</td>
<td>selects or rejects messages that are destined to dead letter queues (with ETO)</td>
</tr>
<tr>
<td>DYNAMic/VIRTual/ETO/VTF/CNT-E/CNT-V</td>
<td>selects or rejects messages that are destined to virtual terminal or ETO LTERMs</td>
</tr>
<tr>
<td>HELD</td>
<td>selects or rejects messages that are associated with held conversations</td>
</tr>
<tr>
<td></td>
<td>Note: HELD is not allowed with the REQueue command; it is only allowed with the DEQueue, DISplay, and UNLoad commands.</td>
</tr>
<tr>
<td>LTERMs</td>
<td>selects or rejects messages that are destined to logical terminals</td>
</tr>
<tr>
<td>MPP</td>
<td>selects or rejects messages that are destined to message processing programs (MPPs)</td>
</tr>
<tr>
<td>MSName</td>
<td>selects or rejects messages that are destined to Multiple Systems Coupling (MSC) destinations</td>
</tr>
<tr>
<td>OTMA</td>
<td>selects or rejects messages that are destined to OTMA programs</td>
</tr>
<tr>
<td>RSMB</td>
<td>selects or rejects messages that are destined to remote transaction scheduler message blocks (RSMBs)</td>
</tr>
<tr>
<td>SMB</td>
<td>selects or rejects messages that are destined to local transaction scheduler message blocks (SMBs)</td>
</tr>
<tr>
<td>STATic</td>
<td>selects or rejects messages that are destined to static terminals</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>selects or rejects messages that are destined to system destinations</td>
</tr>
<tr>
<td>TPName/APPC</td>
<td>selects or rejects messages that are destined to APPC LU 6.2 transaction programs</td>
</tr>
<tr>
<td>TRANsaction</td>
<td>selects or rejects messages that are destined to transactions</td>
</tr>
<tr>
<td>UNDEFINED</td>
<td>on a shared queues system, selects or rejects messages for which the destination does not exist on the current IMS system</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSPcnt</td>
<td>selects or rejects messages that are destined to Intersystem Communication (ISC) destinations</td>
</tr>
</tbody>
</table>

**Figure 68: DESType keyword parameters and associated destination types processed**

**DESType**=

![Diagram of DESType keyword parameters and associated destination types processed]

**DRrn keyword**

Use the DRrn keyword to select or reject a specific device relative record number (DRRN) for processing. Consider the following when specifying the DRrn keyword:

---

**Note**

The DRrn keyword applies to non-shared queues only.
You can only specify the DRrn keyword with the DEQueue, UNLoad, and DISplay command sets.

You must specify a destination when you specify the DRrn keyword.

You can specify the Queue keyword with the DRrn keyword, but Message Advisor validates the queue that is specified in the message instead of the actual queue that contains the message. The queue that is specified in the message and the actual queue that contains the message are usually the same, except for messages that are on the transaction Suspend queue. For messages that are on the transaction Suspend queue, the queue that is specified in the message is queue 1, so the Queue keyword value must include the parameter 1 to delete messages on the Suspend queue.

Only messages on the local queue will match. Messages on shared queues will not match if you specify the DRrn keyword.

Table 326 on page 333 lists the DRrn keyword parameter.

### Table 326: DRrn keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>drrn</strong></td>
<td>the DRRN that you want to select or reject for processing</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td><em>drrn</em> is a value that you specify.</td>
</tr>
</tbody>
</table>

DYNAMic/VIRTual/ETO/VTF/CNT-E/CNT-V keyword

Use the DYNAMic keyword to select or reject messages that are destined to virtual terminal or ETO LTERM.s. You can specify a list of LTERM.s for the DYNAMic keyword value.

You can use VIRTual, ETO, VTF, CNT-E, and CNT-V as aliases for DYNAMic.

Table 327 on page 333 lists the DYNAMic keyword parameter.

### Table 327: DYNAMic keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(mmmmmmmmmm,...)</strong></td>
<td>the virtual terminal or ETO LTERM.s for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td><em>mmmmmmmmmm</em> is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>
ENTRYID keyword

ENTRYID is an optional keyword for the SESelect and REJect subcommands. You can specify a single 12-byte hexadecimal value to indicate the list entry ID for a shared queues message. Consider the following when specifying the ENTRYID keyword:

Note
The ENTRYID keyword applies to shared queues only.

- The ENTRYID is the shared queues equivalent of a DRRN and the ENTRYID keyword is mutually exclusive with the DRRN keyword.

- You can specify the ENTRYID keyword on SESelect and REJect subcommands with the DISplay, DEQueue, and UNLoad commands, but not with the REQueue command.

Table 328 on page 334 lists the ENTRYID keyword parameter.

Table 328: ENTRYID keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxxxxxxxxx</td>
<td>the list entry ID for a shared queues message</td>
</tr>
<tr>
<td></td>
<td>Note: xxxxxxxxxxx is a value that you specify.</td>
</tr>
</tbody>
</table>

KEY keyword

KEY is an optional keyword for the SESelect and REJect subcommands and can be used to specify a single 16-byte shared queue key in hexadecimal.

Note
The KEY keyword applies to shared queues only.

Consider the following when specifying the KEY keyword:

- The KEY keyword is mutually exclusive with all other destination-related keywords.

- Message Advisor for IMS uses the first and last byte of the key to determine the queue type.

- If the first byte has a value of X'01' to X'0B', Message Advisor for IMS uses the value as a client queue type. If the first byte has a value of X'40' or greater,
Message Advisor *for IMS* uses the last byte to determine the private queue type. Other values for the first byte are invalid.

- For private queue types, when the last byte has a value of 'X01' through 'X0B' or 'X81' through 'X8B', Message Advisor *for IMS* uses the last nibble as the private queue type. Otherwise, the key is assumed to be for the control queue.

- You can specify the KEY keyword on SELect and REJect subcommands with the DISplay, DEQueue, and UNLoad commands, but not with the REQueue command.

**HELD keyword**

Use the HELD keyword to select or reject held conversational messages. The value must be the user ID associated with the held conversations. You can also include user IDs in a DESTination parameter list if the DESType keyword value includes the HELD parameter. You can use the HELD keyword and the CONVID keyword together to identify specific conversations.

*Note*

The HELD keyword is not allowed with the REQueue command; it is only allowed with the DISplay, DEQueue, and UNLoad commands.

Table 329 on page 335 lists the HELD keyword parameter.

**Table 329: HELD keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>(mmmmmmmmmm,...)</em></td>
<td>the held conversational users that you want to select or reject for processing</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> <em>mmmmmmmmmm</em> is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

**LABEL keyword**

Use the LABEL keyword to provide a name (label) for the SELect or REJect subcommand statement. Message Advisor uses the LABEL keyword value to identify the statement in reports.

Table 330 on page 336 lists the LABEL keyword parameter.
Table 330: LABEL keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEL00001</td>
<td>the name of this SESelect or REJect subcommand statement</td>
</tr>
<tr>
<td>REJ00001</td>
<td></td>
</tr>
<tr>
<td>cccccccc</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The default label is SEL0001 or REJ0001 for the first statement. The number increments by one for each subsequent statement. cccccccc is a value that you specify.

**LTERM keyword**

Use the LTERM keyword to select or reject messages that are destined to logical terminals. You can specify a list of LTERMs for the LTERM keyword value.

Table 331 on page 336 lists the LTERM keyword parameter.

Table 331: LTERM keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmm,...)</td>
<td>the logical terminals for which you want to select or reject messages for processing</td>
</tr>
</tbody>
</table>

**Note:** mmmmmmmmm is a value that you specify. You can specify a list of values or patterns.

**LUname keyword**

Use the LUname keyword to select or reject messages that are associated with specific logical unit (LU) names. Consider the following when specifying the LUname keyword:

- You can specify a list of LU names or patterns for the LUname keyword value.
- You can use LUname and TPName together to identify a specific APPC destination.
- Message Advisor assumes DESType=TPName when you specify the LUname keyword.
- Only APPC destinations with a matching LU name will match the SESelect or REJect statement.
An LU name can include a network ID (netid.luname). If you specify a network ID, both the LU name and the network ID must match the SELECT or REJECT statement. If you do not specify a network ID, any network ID will match the SELECT or REJECT statement.

Table 332 on page 337 lists the LUname keyword parameter.

**Table 332: LUname keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmmm.mmmmmmmm,...)</td>
<td>the LU names for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> mmmmmmmmmm.mmmmmmmm is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

**MFSNAME keyword**

Use the MFSNAME keyword to select or reject messages by message format service (MFS) name. Consider the following when specifying the MFSNAME keyword:

- You can specify a list of MFS names or patterns for the MFSNAME keyword value.
- Only messages where the MFS name in the system prefix matches one of the MFS names that you specify will match the SELECT or REJECT statement.
- Messages with a blank or null MFS name will not match, even for MFSNAME=*. Therefore, you can specify MFSNAME=* to select all messages with a non-blank MFS name.

Table 333 on page 337 lists the MFSNAME keyword parameter.

**Table 333: MFSNAME keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmmmm,...)</td>
<td>the MFS names by which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> mmmmmmmmm is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>
MSGTYPE keyword

Use the MSGTYPE keyword to select or reject messages by message type. You can specify a list of message types for the MSGTYPE keyword value. To exclude a message type (except ALL), add NO in front of the applicable parameter. For example, MSGTYPE=NOINPUT includes all message types except input messages.

Table 334 on page 338 lists the MSGTYPE keyword parameters. **CR**Figure 58 on page 337 illustrates the message types that are processed for each parameter.

Table 334: MSGTYPE keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>selects or rejects all message types for processing</td>
</tr>
</tbody>
</table>
| BMPSWITCH           | selects or rejects messages sent from a non-transaction driven batch message processing (BMP) program to a transaction (program-to-program switch)  
Note: Because another input message does not drive these messages, you may want to requeue them along with input messages. |
| CMD                 | selects or rejects messages sent from any source to the system              |
| IN-APPC/APPC        | selects or rejects messages sent from an APPC session to a transaction  
You can use APPC as an alias for IN-APPC. |
| IN-CNT/CNT          | selects or rejects messages sent from a terminal to a transaction  
You can use CNT as an alias for IN-CNT. |
| IN-OTMA/OTMA        | selects or rejects messages sent from an OTMA session to a transaction  
You can use OTMA as an alias for IN-OTMA. |
| INPUT               | selects or rejects all input messages (INPUT=IN-CNT + IN-APPC + IN-OTMA) |
| LTERMS              | selects or rejects all messages that are destined to logical terminals or output devices (LTERMS=OUTPUT + LTERMSWITCH + SYSTEM) |
| LTERMSWITCH         | selects or rejects messages sent from an input source (logical terminal) to another logical terminal (message switch) |
| OUTPUT              | selects or rejects all output messages that originated from message regions (OUTPUT=REPLY + REPORT) |
| PGMSWITCH           | selects or rejects messages sent from a transaction-driven program to another transaction (program-to-program switch)  
Note: Input messages or other switches drive these types of messages. You may want to exclude these messages to avoid duplicates when requeueing messages from a log. |
| REPLY/IOPCB         | selects or rejects messages sent from a message region back to the input source (normally IOPCB)  
You can use IOPCB as an alias for REPL. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORT/ALT-PCB</td>
<td>selects or rejects messages sent from a message region to an LTERM other than the input source (ALT-PCB)</td>
</tr>
<tr>
<td></td>
<td>You can use ALT-PCB as an alias for REPORT.</td>
</tr>
<tr>
<td>SWITCH</td>
<td>selects or rejects all switches, including program-to-program switches and message switches from one LTERM to another (SWITCH=PGMSWITCH + BMPSWITCH + LTERMSWITCH)</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>selects or rejects all messages from the system</td>
</tr>
<tr>
<td>TRANSACTIONS</td>
<td>selects or rejects all messages that drive transactions (TRANSACTIONS=INPUT + PGMSWITCH + BMPSWITCH)</td>
</tr>
</tbody>
</table>

**Figure 69: MSGTYPE keyword parameters and associated message types processed**
**MSNAME keyword**

Use the MSNAME keyword to select or reject messages that are destined to Multiple Systems Coupling (MSC) destinations. MSNAME indicates the MSC destination name. You can specify a list of MSC destination names for the MSNAME keyword value.

Table 335 on page 340 lists the MSNAME keyword parameter.

**Table 335: MSNAME keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| (mmmmmmmmmm,...) | the MSC destination names for which you want to select or reject messages for processing  
Note: mmmmmmmmm is a value that you specify. You can specify a list of values or patterns. |

**NEWER_than keyword**

Use the NEWER_than keyword to select all messages that occur after a specific day. The NEWER_than keyword excludes the STAStart keyword.

**Note**

See “NEWER_THAN and OLDER_THAN keywords” on page 296 for more information about using the NEWER_than keyword.

Table 336 on page 340 lists the NEWER_than keyword parameter.

**Table 336: NEWER_than keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| nnn | selects messages that occur within nnn days of the current system date  
**Note:** nnn is a value that you specify. |

**NODENAME keyword**

Use the NODENAME keyword to select or reject messages that are destined to Virtual Telecommunications Access Method (VTAM) node names. Consider the following when specifying the NODENAME keyword:
You can specify a list of VTAM node names or patterns for the NODENAME keyword value.

Message Advisor assumes DESType=_CNT when you specify the NODENAME keyword.

Only messages where the node name in the message prefix matches one of the node names that you specify will match the SELECT or REJECT statement.

Messages with a blank node name will not match the SELECT or REJECT statement, even for NODENAME=*.
Therefore, NODENAME=* selects all VTAM messages.

Table 337 on page 341 lists the NODENAME keyword parameter.

Table 337: NODENAME keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmmmm,...)</td>
<td>the VTAM node names for which you want to select or reject messages for processing</td>
</tr>
</tbody>
</table>

**Note:** mmmmmmmmm is a value that you specify. You can specify a list of values or patterns.

OFFSET keyword

Use the OFFSET keyword along with the DATA keyword to indicate that the string specified for the DATA keyword must occur at an exact location within the message.

You can specify a single number or a range for the OFFSET keyword value. A range is two numbers in parentheses, separated by a comma. If you specify a single number, the string must begin at that offset. If you specify a range, the string can start anywhere in the range and must be wholly contained within the range.

If you do not also specify the SEGMENT keyword, the offset is from the beginning of the message data. Message Advisor considers each data segment to immediately follow the previous data segment and ignores record boundaries. If you do specify the SEGMENT keyword, the offset is from the beginning of the first segment that you indicate. Because Message Advisor counts but does not search the segment header word, the beginning offset must be at least four.

**Note**

When you specify the OFFSET keyword along with the REPLACE keyword on a CHANGE subcommand, Message Advisor only replaces data strings that are within the specified offset.
Table 338 on page 342 lists the OFFSET keyword parameter.

Table 338: OFFSET keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(nnnn,nnnn/*)</td>
<td>the offset or offset range at which the string specified for the DATA keyword must occur within the message.</td>
</tr>
<tr>
<td></td>
<td>If you specify an offset range, you can replace the second nnnn value with * to indicate no upper limit.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: nnnn is a value that you specify.</td>
</tr>
</tbody>
</table>

OIMS keyword

Use the OIMS keyword to select or reject messages by originating IMS ID. You can specify a list of IMS IDs or patterns for the OIMS keyword value.

Table 339 on page 342 lists the OIMS keyword parameter.

Table 339: OIMS keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmm,m,...)</td>
<td>the originating IMS IDs for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: mmmmmmmmm is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

OLDER_than keyword

Use the OLDER_than keyword to select all messages that occur before a specific day. The OLDER_than keyword excludes the STOP keyword.

**Note**

See “NEWER_THAN and OLDER_THAN keywords” on page 296 for more information about using the OLDER_than keyword.

Table 340 on page 343 lists the OLDER_than keyword parameter.
Table 340: OLDER_than keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnn</td>
<td>selects messages that are more than nnn days older than the current system date</td>
</tr>
<tr>
<td>range 0-365</td>
<td>Note: nnn is a value that you specify.</td>
</tr>
</tbody>
</table>

**ORIGin keyword**

Use the ORIGin keyword to select or reject messages by origin. Consider the following when specifying the ORIGin keyword:

- You can specify a list of origins or patterns for the ORIGin keyword value.
- An origin that matches any of the patterns will match the SELect or REject statement.
- The message origin is the name from the message prefix, which is normally the LTERM that created the original input message for the unit of work.
- Messages with a blank or null origin will not match, even for ORIGin=*. Therefore, ORIGin=* selects or rejects all messages with a non-blank origin.
- You can specify the value SYSTEM to indicate that Message Advisor is to select or reject messages with a system origin.
- You cannot specify APPC origins by name.
- You cannot specify OTMA origins with the ORIGin keyword. Instead, use the ORGTMEMBER and ORGTPPIPE keywords. For OTMA MQSeries messages, you can also use the ORGQMANAGER and ORGQNAME keywords to select by the Reply-to queue.

Table 341 on page 344 lists the ORIGin keyword parameter.
Table 341: ORIGin keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmm,...)</td>
<td>the message origins by which you want to select or reject messages for processing ORIGin is normally the LTERM that created the original input message for this unit of work.</td>
</tr>
<tr>
<td>Note: mmmmmmmmm is a value that you specify. You can specify a list of values or patterns.</td>
<td></td>
</tr>
</tbody>
</table>

**ORGQNAME keyword**

Use the ORGQNAME keyword to select or reject messages by the origin MQSERIES Reply-to QNAME. Consider the following when specifying the ORGQNAME keyword:

- You can specify a list of origin QNAMEs or patterns for the ORGQNAME keyword value.
- An origin QNAME that matches any of the patterns will match the SELECT or REJECT statement.
- Messages with a blank or null origin will not match, even for ORGQNAME=*. Therefore, ORGQNAME=* selects or rejects all messages with a non-blank origin QNAME.

Table 342 on page 344 lists the ORGQNAME keyword parameter.

Table 342: ORGQNAME keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>originQname</td>
<td>the origin MQSERIES Reply-to QNAME by which you want to select or reject messages for processing</td>
</tr>
<tr>
<td>Note: originQname is a value that you specify. You can specify a list of values or patterns.</td>
<td></td>
</tr>
</tbody>
</table>

**ORGQMANAGER keyword**

Use the ORGQMANAGER keyword to select or reject messages by the origin MQSERIES Reply-to QMANAGER name. Consider the following when specifying the ORGQMANAGER keyword:
You can specify a list of origin QMANAGER names or patterns for the ORGQMANAGER keyword value.

An origin QMANAGER that matches any of the patterns will match the SELECT or REJECT statement.

Messages with a blank or null origin will not match, even for ORGQMANAGER=*.

Therefore, ORGQMANAGER=* selects or rejects all messages with a non-blank origin QMANAGER name.

Table 343 on page 345 lists the ORGQMANAGER keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>originQmanager</td>
<td>the origin MQSERIES Reply-to QMANAGER name by which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> <em>originQmanager</em> is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

**ORGTIPIPE keyword**

Use the ORGTPIPE keyword to select or reject messages by the origin TPIPE name. Consider the following when specifying the ORGTPIPE keyword:

- You can specify a list of origin TPIPE names or patterns for the ORGTPIPE keyword value.

- An origin TPIPE that matches any of the patterns will match the SELECT or REJECT statement.

- Messages with a blank or null origin will not match, even for ORGTPIPE=*.

Therefore, ORGTPIPE=* selects or rejects all messages with a non-blank origin TPIPE name.

Table 344 on page 346 lists the ORGTPIPE keyword parameter.
Table 344: ORGTPipe keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>originTpipe</td>
<td>the origin TPIPE name by which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> <code>originTpipe</code> is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

**ORGTMEMBER keyword**

Use the ORGTMEMBER keyword to select or reject messages by the origin TMEMBER name. Consider the following when specifying the ORGTMEMBER keyword:

- You can specify a list of origin TMEMBER names or patterns for the ORGTMEMBER keyword value.
- An origin TMEMBER that matches any of the patterns will match the SELect or REJect statement.
- Messages with a blank or null origin will not match, even for `ORGTMEMBER=*`. Therefore, `ORGTMEMBER=*` selects or rejects all messages with a non-blank origin TMEMBER name.

Table 345 on page 346 lists the ORGTMEMBER keyword parameter.

Table 345: ORGTMEMBER keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>originTmember</td>
<td>the origin TMEMBER name by which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> <code>originTmember</code> is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

**OTMAQ keyword**

Use the OTMAQ keyword to restrict messages on OTMA destinations.

**Note**
The OTMAQ keyword is *not* allowed for the REQueue command set.
Table 346 on page 347 lists the OTMAQ keyword parameters.

Table 346: OTMAQ keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>does not restrict messages on OTMA destinations to any particular queue</td>
</tr>
<tr>
<td>REGULAR</td>
<td>restrict messages on OTMA destinations to messages on the REGULAR output queue</td>
</tr>
<tr>
<td>HOLDQ</td>
<td>restricts messages on OTMA destinations to messages using the SEND-ONLY protocol waiting on the HOLDQ to be retrieved by the client</td>
</tr>
</tbody>
</table>

**OTOKEN keyword**

Use the OTOKEN keyword to select or reject messages by originating IMS token. You can specify a list of eight-byte hexadecimal values for the OTOKEN keyword value.

*Note*

For a message to match, the value must exactly match the originating IMS token (timestamp) in the message.

Table 347 on page 347 lists the OTOKEN keyword parameter.

Table 347: OTOKEN keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(xx...xx,...)</td>
<td>the originating IMS token(s) for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> xx...xx is a value that you specify. You can specify a list of values.</td>
</tr>
</tbody>
</table>

**PIMS keyword**

Use the PIMS keyword to select or reject messages by processing IMS ID. You can specify a list of IMS IDs or patterns for the PIMS keyword value.

Table 348 on page 348 lists the PIMS keyword parameter.
Table 348: PIMS keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| (mmmmmmmmmm,...) | the processing IMS IDs for which you want to select or reject messages for processing.  
  **Note:** *mmmmmmmmmm* is a value that you specify. You can specify a list of values or patterns. |

**PSB keyword**

Use the PSB keyword to select or reject messages by PSB. Message Advisor selects or rejects only those messages that are destined to transactions that are currently associated with the specified PSBs. Consider the following when specifying the PSB keyword:

- You can specify a list of PSBs or patterns for the PSB keyword value.
- When you specify the PSB keyword, Message Advisor assumes `DESTYpe=TRANSACTION`.
- The PSB keyword is mutually exclusive with the `DESTYpe` keyword, the `DESTination` keyword, and any of the individual destination type keywords.
- Because Message Advisor queries the IMS system to obtain a list of transactions that are associated with the specified PSBs and uses those transactions for destination filtering, the IMS system must be active before you can specify the PSB keyword.

**Note**  
The transactions that are associated with a PSB may have changed since the input file was created. The PSB keyword always provides the current associations, and not necessarily the associations that were in effect at the time the message was logged.

Table 349 on page 349 lists the PSB keyword parameter.
Table 349: PSB keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmm,...)</td>
<td>the PSBs for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td>Message Advisor selects or rejects only those messages that are destined</td>
</tr>
<tr>
<td></td>
<td>to transactions that are currently associated with the specified PSBs.</td>
</tr>
<tr>
<td>Note: mmmmmmmmm is a value</td>
<td>You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

PTOKEN keyword

Use the PTOKEN keyword to select or reject messages by processing IMS token. You can specify a list of eight-byte hexadecimal values for the PTOKEN keyword value.

Note

For a message to match, the value must exactly match the processing IMS token (timestamp) in the message.

Table 350 on page 349 lists the PTOKEN keyword parameter.

Table 350: PTOKEN keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(xx...xx,...)</td>
<td>the processing IMS token(s) for which you want to select or reject messages</td>
</tr>
<tr>
<td></td>
<td>for processing</td>
</tr>
<tr>
<td>Note: xx...xx is a value</td>
<td>You can specify a list of values.</td>
</tr>
</tbody>
</table>

QPOSITION keyword

Use the QPOSITION keyword to select or reject only messages in a specific position or range of positions on a queue. Consider the following when specifying the QPOSITION keyword:

- If you specify a single value (e.g., QPOSITION=3), only the message in that position on the queue will match.
- If you specify two values (e.g., QPOSITION=(1,10)), all messages in that range of positions will match.
The word FIRST in the first position indicates the first message on the queue. The word LAST or an asterisk indicates the last message on the queue. For example, QPOSITION=FIRST refers to the first message on the queue, QPOSITION=LAST refers to the last message on the queue, and QPOSITION=(3,*)) refers to the third through the last message on the queue.

The QPOSITION keyword is mutually exclusive with the DRRN and ENTRYID keywords.

You can specify the QPOSITION keyword on SELECT and REJECT subcommands with the DISPLAY, DEQUEUE, and UNLOAD commands, but not with the REQUEUE command.

When you use the QPOSITION keyword with the DISPLAY TYPE=DEST_QUEUES command, the counts that are returned include only the messages within the specified range.

**QPOSITION keyword and relative position**

The relative position that is selected when you specify the QPOSITION keyword depends on the type of queue, as follows:

- For non-shared queues, the relative position selected depends on whether you specify the QUEUE keyword on the same SELECT or REJECT subcommand that includes the QPOSITION keyword. If you do not specify the QUEUE keyword or if you specify QUEUE=ALL, the position is relative to all messages for the destination (regardless of the queue). If you specify any other value for the QUEUE keyword, the position is relative to the beginning of each queue. For example, if a destination has two messages on queue 1 and ten messages on queue 3, then 'QUEUE=3,QPOSITION=6' refers to the sixth message on queue 3. 'QUEUE=ALL,QPOSITION=6' refers to the fourth message on queue 3 (the sixth message overall). For transactions, Message Advisor assumes that suspend queue messages follow all other messages.

- For shared client queues the position is relative to a specific key and queue type. If a destination has messages on different queue types (for example, the transaction ready queue and the transaction suspend queue), there will be more than one message in each position. You should specify the QTYPE keyword unless you want to process both queues.

- For shared private queues, the position is relative to each queue type, regardless of key. Since the position of the messages is random, private queue messages should not normally be selected by position. The QPOSITION keyword is not allowed with shared private queues if you also specify the KEY keyword, and QPOSITION=LAST is never allowed for private queues.

Table 351 on page 351 lists the QPOSITION keyword parameters.
Table 351: QPOSITION keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| nn        | selects or rejects messages for a specific position on a queue  
Note: nn is a value that you specify. |
| (nn,nn)   | selects or rejects messages for a range of positions on a queue  
Note: nn,nn is a value that you specify. |
| FIRST     | when specified in the first position, indicates the first message on the queue |
| LAST      | when specified in the second position, indicates the last message on the queue  
Note: You can use * as an alias for LAST. Because queues are not static, it is possible (but not likely) for more than one message to match if you specify QPOSITION=LAST. If a new message is added to the end of a queue while the last message is being processed, both may match as the last message. If the last message is dequeued while the penultimate message is being scanned, there may be no match. |

QTYPE keyword

QTYPE is an optional keyword for the SELect and REJect subcommands and limits processing to specific queues. You can specify a list of queue types with the QTYPE keyword.

Note

The QTYPE keyword applies to shared queues only.

Consider the following when specifying the QTYPE keyword:

- If you only specify client queue types for the QTYPE keyword on a SELect or REJect subcommand, the DEQueue, UNLoad, and DISplay commands will search only the indicated queues for matching messages. For the REQueue command, the MSGRSQTY field in the message prefix determines the queue type for filtering.

- When a private queue keyword (COLDQ or LOCKQ) is used with QTYPE by itself or in combination with client queue types, only messages which are currently on the private queue AND which were originally on one of the listed client queues will match. For example, QTYPE=(COLDQ, SERIAL) will display only serial transactions on the cold queue. You must use two SELect subcommands to select both the COLDQ and the SERIAL client queue.
If you specify the QTYPE keyword on a SELECT or REJECT subcommand with a DISPLAY, DEQUEUE, or UNLOAD command for non-shared queues, the subcommand will match nothing and an error may result.

If you specify the QTYPE keyword on a SELECT or REJECT subcommand with a REQUEUE command, filtering will be based on the queue type in the message (if any). Because the queue type in the message can only be a client queue type, the COLDQ and LOCKQ parameters will not match. Also, messages that were created on a non-shared queue system will have a queue type of zero and will not match.

When specified on SELECT or REJECT subcommands with the REQUEUE command, QTYPE is used only to select messages. You can requeue messages to any queue, regardless of the QTYPE keyword.

Table 352 on page 352 lists the QTYPE keyword parameters. **CR**Figure 59 on page 352 illustrates the queues that are processed for each parameter.

Table 352: QTYPE keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| ALLDEST   | processes all queues that are listed for the ALLTRAN and ALLTERM parameters  
**Note:** Only client queues will match unless you explicitly select private queues. |
| ALLTERM   | limits processing to queues 5, 7, 8, and 9  
**Note:** You can use CNT as an alias for ALLTERM.  
Only client queues will match unless you explicitly select private queues. |
| ALLTRAN   | limits processing to queues 1, 3, and 4  
**Note:** You can use SMB as an alias for ALLTRAN.  
Only client queues will match unless you explicitly select private queues. |
| COLDQ     | private queue  
**Note:** Private queue types are not allowed on SELECT and REJECT subcommands with the REQUEUE command.  
When you specify a private queue type for the QTYPE keyword by itself or in combination with a client queue type, only messages which are currently on the private queue AND which were originally on one of the listed client queues will match. For example, QTYPE=(COLDQ, SERIAL) will only display serial transactions on the cold queue. You must use two SELECT subcommands to select both the COLDQ and the SERIAL client queue. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| ControlQ | private queue  
  **Note:** Private queue types are not allowed on SELect and REJect subcommands with the REQueue command.  
The control queue does not contain IMS messages, but contains CQS control information. You can use the DISplay commands to display these control elements. However, you cannot use other keywords except for the DESTination keyword, which you can use to select the key.  
  **Note:** You can use CNTLQ as an alias for ControlQ.  
| LOCKQ | private queue  
  **Note:** Private queue types are not allowed on SELect and REJect subcommands with the REQueue command.  
When you specify a private queue type for the QTYPE keyword by itself or in combination with a client queue type, only messages which are currently on the private queue AND which were originally on one of the listed client queues will match. For example, QTYPE=(LOCKQ, SERIAL) will only display serial transactions on the lock queue. You must use two SELECT subcommands to select both the LOCKQ and the SERIAL client queue.  
| Q1 | limits processing to queue 1  
  **Note:** You can use 1, TRAN1, TRANRDY, and TRANSACT as aliases for Q1.  
| Q2 | limits processing to queue 2 (a staging queue)  
  **Note:** The staging queue types are not allowed on the REQueue command.  
The staging queues contain continuation segments for IMS messages, but not the beginning of the message. You cannot use destination- or message-related selection keywords when selecting the staging queues.  
| Q3 | limits processing to queue 3  
  **Note:** You can use 3, TSUSP3, and SUSPEND as aliases for Q3.  
| Q4 | limits processing to queue 4  
  **Note:** You can use 4, TSERL4, and SERIAL as aliases for Q4.  
| Q5 | limits processing to queue 5  
  **Note:** You can use 5, LTERM5, and LTRM5 as aliases for Q5.  
| Q6 | limits processing to queue 6 (a staging queue)  
  **Note:** The staging queue types are not allowed on the REQueue command.  
The staging queues contain continuation segments for IMS messages, but not the beginning of the message. You cannot use destination- or message-related selection keywords when selecting the staging queues.  

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Q7        | limits processing to queue 7  
**Note:** You can use 7, APPC7, and TPNAME7 as aliases for Q7. |
| Q8        | limits processing to queue 8  
**Note:** You can use 8, REMOTE8, and RMT8 as aliases for Q8. |
| Q9        | limits processing to queue 9  
**Note:** You can use 9 and OTMA9 as aliases for Q9. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAGEQ</td>
<td>limits processing to queues 2 and 6</td>
</tr>
</tbody>
</table>

Note: The staging queue types are not allowed on the REQueue command. The staging queues contain continuation segments for IMS messages, but not the beginning of the message. You cannot use destination- or message-related selection keywords when selecting the staging queues.

Figure 70: QTYPE keyword parameters and associated queues processed
Queue keyword

Use the Queue keyword to select or reject messages that are destined to specific queues.

Consider the following when specifying the Queue keyword:

- You can specify a list of queues for the Queue keyword value. For example, `Queue=(1,2)` selects or rejects messages for queues 1 and 2.

- To exclude queues from processing, add ‘¬’ in front of the applicable parameters (except for All). For example, `Queue=¬3` selects or rejects messages for all queues except queue 3.

- For the REQueue and DISplay command sets, the default Queue parameter is All.

- For the DEQueue and UNLoad command sets, the default Queue parameters are 1, 2, 3, 4, and HELD. To include the Backup and Suspend queues, you must specify All.

- If you specify the Queue keyword and Message Advisor encounters a message without a shared queue type, Message Advisor processes the Queue keyword normally. When a message contains a shared queue type, Message Advisor ignores the queue in the message and uses the queue type to determine a match as follows:

<table>
<thead>
<tr>
<th>Queue=</th>
<th>matches these queue types:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>QTYPE 1</td>
</tr>
<tr>
<td>2</td>
<td>ALLTRAN (QTYPEs 1, 3, 4)</td>
</tr>
<tr>
<td>3</td>
<td>QTYPE 3</td>
</tr>
<tr>
<td>4</td>
<td>QTYPE 4</td>
</tr>
<tr>
<td>ALL</td>
<td>ALLDEST (QTYPEs 1, 3, 4, 5, 7, 8, 9)</td>
</tr>
<tr>
<td>BACKUP/5</td>
<td>QTYPE 5</td>
</tr>
<tr>
<td>HELD</td>
<td>none</td>
</tr>
<tr>
<td>SUSPEND</td>
<td>QTYPE 3</td>
</tr>
</tbody>
</table>

Table 353 on page 357 lists the Queue keyword parameters.
### Table 353: Queue keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| All       | selects or rejects messages for all queues  
For the REQueue command set, All includes queues 1, 2, 3, and 4.  
For the DEQueue, Display, and UNLoad command sets, All includes queues 1, 2, 3, and 4, as well as the Backup, Suspend, and Held queues. |
| 1         | selects or rejects messages that are destined to the queue that contains replies to response or conversational-type messages |
| 2         | selects or rejects messages that are destined to the queue that contains replies to transactions from terminals in exclusive mode; used with the /SET command |
| 3         | selects or rejects messages that are destined to the queue that contains IMS system messages |
| 4         | selects or rejects messages that are destined to the queue that contains message switches, alternate PCB output, and other traffic |
| Backup    | selects or rejects messages that are destined to the Backup queue  
The Backup queue contains active conversational messages.  
**Note:** Backup is not allowed for the REQueue command set. You can use 5 as an alias for Backup. |
| Held      | selects or rejects held conversational messages  
**Note:** Held is not allowed for the REQueue command set. |
| Suspend   | selects or rejects messages that are destined to the transaction Suspend queue  
**Note:** Suspend is not allowed for the REQueue command set. |

### RCNT keyword

Use the RCNT keyword to select or reject messages for remote CNTs (RCNTs). Consider the following when specifying the RCNT keyword:

- You can specify a list of remote CNTs or patterns for the RCNT keyword value.
- Message Advisor only selects or rejects messages that are queued to MSNAMEs for the specified remote CNTs.
- The RCNT keyword is mutually exclusive with the DESTYpe keyword.
- When you specify the RCNT keyword, Message Advisor assumes `DESType=MSNAME`. 
You can also specify DESTination or MSNAME to search only specific MSNAMEs. Otherwise, Message Advisor searches all MSNAME destinations.

It is not required that the remote CNT currently exist, or currently be associated with a specific MSNAME. Message Advisor checks all messages that are queued to MSNAME destinations for the specified remote CNT destinations in the message prefix.

Table 354 on page 358 lists the RCNT keyword parameter.

### Table 354: RCNT keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmm,...)</td>
<td>the remote CNTs for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>mmmmmmmmmmm is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

### REGION keyword

Use the REGION keyword to select or reject messages by region. Message Advisor selects or rejects only temporary queue messages for the specified regions.

---

**Note**

The REGION keyword is only allowed for the DISplay command set.

---

Consider the following when specifying the REGION keyword:

- You can specify a list of regions for the REGION keyword value.
- Use the job name to identify batch message processing (BMP) regions.
- Use the transaction code to identify message processing program (MPP) regions (to select or reject regions that are currently processing that transaction).
- If you specify REGION=*, Message Advisor only selects temporary queue messages.

Table 355 on page 359 lists the REGION keyword parameter.
Table 355: REGION keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmm,...)</td>
<td>the regions for which you want to select or reject messages for processing Message Advisor selects or rejects only temporary queue messages for the specified regions. Note: mmmmmmmmm is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

**REGION_ID keyword**

Use the REGION_ID keyword to select or reject messages for a specific region ID. Message Advisor selects or rejects only temporary queue messages for the specified region ID.

*Note*

The REGION_ID keyword is only allowed for the DISplay command set.

Consider the following when specifying the REGION_ID keyword:

- Specify REGION_ID=0 to select only destination queues (no temporary queues).
- If you do not specify REGION_ID, Message Advisor displays and processes both temporary and destination queues.

Table 356 on page 359 lists the REGION_ID keyword parameter.

Table 356: REGION_ID keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnnnnn</td>
<td>the region ID for which you want to select or reject messages for processing Message Advisor selects or rejects only temporary queue messages for the specified region ID. Note: nnnnnnnnn is a value that you specify.</td>
</tr>
</tbody>
</table>

**RSMB keyword**

Use the RSMB keyword to select or reject messages that are destined to remote scheduler message blocks (RSMBs).
Table 357 on page 360 lists the RSMB keyword parameter.

**Table 357: RSMB keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmm,...)</td>
<td>the remote SMBs for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> $mmmmmmmm$ is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

**SAFGROUP keyword**

Use the SAFGROUP keyword to select or reject messages that contain a security segment with a specific SAF Group ID. Consider the following when specifying the SAFGROUP keyword:

- You can specify a list of SAF Group IDs or patterns for the SAFGROUP keyword value.

- Only messages that contain a security segment with one of the SAF Group IDs that you specify will match the SElect or REject statement.

- If you specify the SAFGROUP keyword, messages that do not contain a security segment will not match the SELect or REJect statement, even for SAFGROUP=*. Therefore, you can specify SAFGROUP=* to select or reject all messages with a security segment.

Table 358 on page 360 lists the SAFGROUP keyword parameter.

**Table 358: SAFGROUP keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmm,...)</td>
<td>the SAF Group IDs for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> $mmmmmmmm$ is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

**SCRAPcode keyword**

Use the SCRAPcode keyword to specify a list of scrap codes to select messages which were previously scrapped for requeue.
**Note**
The SCRAPcode keyword is only allowed for the REQueue command set.

Consider the following when specifying the SCRAPcode keyword:

- The SCRAPcode keyword is only effective if one of the input files is a SCRAP file from a previous requeue. If none of the input files is a SCRAP file, no messages will match SELECT or REJECT statements that include the SCRAPcode keyword.

- You can use keywords instead of scrap code numbers (for example, `SCRAP=SELECT`). For a list of the supported keywords, see the ERROR keyword description in “REQUEUE command” on page 227. The SCRAP/ERROR report also lists the supported keywords.

- Use ‘¬’ in front of scrap codes to negate those codes. For example, to scrap all codes except 1 and 2, specify `SCRAP=(¬1,¬2)`.

Table 359 on page 361 lists the SCRAPcode keyword parameters.

### Table 359: SCRAPcode keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| (nn) | the scrap codes to be selected for requeue  
*Note:* nn is a value that you specify. |
| ANY | requeues all scrap codes |
| ERROR | requeues only the scrap codes that indicate a default error condition |

`SCRAPcode=ERROR` is equivalent to `SCRAP=(NOSOURCE, NODEST, INSERT, ENQFAIL, LENGTH, PREFIX, QSTOP, CHANGE, CANCEL, ConVCCB, N0APPC, RECORD, BADQUEUE, NODATA, NOSYSID, NOTMA, BADSYS)`

| NOTERROR | requeues only the scrap codes that do not indicate a default error condition |

### SEGMENT keyword

If you *do not* specify the DATA keyword, use the SEGMENT keyword to select or reject only those messages that contain the specified number of segments. If you *do* specify the DATA keyword, use the SEGMENT keyword to indicate that Message Advisor is to search only the specified segments for the DATA keyword string.

The SEGMENT keyword value can be a single number or a range (two numbers in parentheses separated by a comma). You can replace the second number of a range with an asterisk (*) to indicate no upper limit.
Note
When you specify the SEGMENT keyword in combination with the REPLACE keyword on a CHANGE subcommand, Message Advisor only replaces data strings within the specified segments.

Table 360 on page 362 lists the SEGMENT keyword parameter.

Table 360: SEGMENT keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(nnnn,nnnn/*)</td>
<td>if you do not specify the DATA keyword, the number of segments a message must contain to match the SELECT or REJECT statement</td>
</tr>
<tr>
<td></td>
<td>if you do specify the DATA keyword, the specific segments Message Advisor is to search for the DATA keyword string</td>
</tr>
<tr>
<td></td>
<td>You can specify a single number or a range. If you specify a range, you can replace the second nnnn value with * to indicate no upper limit.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> nnnn is a value that you specify.</td>
</tr>
</tbody>
</table>

SMB keyword

Use the SMB keyword to select or reject messages that are destined to local scheduler message blocks (SMBs). You can specify a list of SMBs for the SMB keyword value.

Table 361 on page 362 lists the SMB keyword parameter.

Table 361: SMB keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmm,...)</td>
<td>the SMBs for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> mmmmmmmmm is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

STArt keyword

Use the STArt keyword to select all messages with a time stamp that occurs after a specific time. Use the STArt keyword in combination with the STOP keyword to specify a beginning and ending time stamp by which to select messages for processing. The STArt keyword excludes the NEWER_than keyword.
Note
See “START and STOP Keywords” on page 294 for more information about using the STArt keyword.

Table 362 on page 363 lists the STArt keyword parameter.

Table 362: STArt keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.yyyy.ddd.hh.mm.ss.t&lt;offset&gt;</td>
<td>selects messages with a time stamp that occurs after a specific time</td>
</tr>
<tr>
<td></td>
<td>Note: .yyyy.ddd.hh.mm.ss.t&lt;offset&gt; is a value that you specify. See “Time stamp format” on page 295 for additional information.</td>
</tr>
</tbody>
</table>

STATic keyword

Use the STATic keyword to select or reject messages that are destined to static terminals. You can specify a list of static terminals for the STATic keyword value. Static terminals are those that are defined in the IMS SYSGEN.

Table 363 on page 363 lists the STATic keyword parameter.

Table 363: STATic keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmm,...)</td>
<td>the static terminals for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td>Note: mmmmmmmmm is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

STOP keyword

Use the STOP keyword to select all messages with a time stamp that occurs before a specific time. Use the STArt keyword in combination with the STOP keyword to specify a beginning and ending time stamp by which to select messages for processing. The STOP keyword excludes the OLDER_than keyword.
Note
See “START and STOP Keywords” on page 294 for more information about using the STOP keyword.

Table 364 on page 364 lists the STOP keyword parameter.

Table 364: STOP keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| yyyy.ddd.hh.mm.ss.t<offset> | selects messages with a time stamp that occurs before a specific time  
**Note**: yyyy.ddd.hh.mm.ss.t<offset> is a value that you specify. See “Time stamp format” on page 295 for additional information. |

**SYSID keyword**

Use the SYSID keyword to select or reject messages by system ID. Consider the following when specifying the SYSID keyword:

- You must specify the SYSID keyword value as two numbers between 1 and 2036. You must enclose the numbers in parentheses and use a comma to separate them.

- The first number of the SYSID keyword value indicates the remote system ID, and the second number indicates the local system ID.

- You can replace either number of the SYSID keyword value with an asterisk (*) to indicate that any number is allowed.

- You can replace either number of the SYSID keyword value with the following parameters:
  - LOCAL
  - REMOTE
  - UNDEFINED

- You can replace the second number of the SYSID keyword value with the parameter SAME.

Table 365 on page 365 lists the SYSID keyword parameters.
### Table 365: SYSID keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| \((nnnn,nnnn)\) | Between 1 and 2036 the remote and local system IDs for which you want to select or reject messages for processing.  
  The first number indicates the remote system ID, and the second number indicates the local system ID.  
  **Note:** *nnnn* is a value that you specify.                                                                                      |
| *               | Can replace either of the SYSID *nnnn* values so that any number will match the SELECT or REJECT statement.                                     |
| LOCAL/L         | Can replace either of the SYSID *nnnn* values so that any system ID that is defined as local on the current IMS system will match the SELECT or REJECT statement.  
  You can use L as an alias for LOCAL.                                                                                                  |
| REMOTE/R        | Can replace either of the SYSID *nnnn* values so that any system ID that is defined as remote on the current IMS system will match the SELECT or REJECT statement.  
  You can use R as an alias for REMOTE.                                                                                                 |
| SAME/S          | Can replace the second SYSID *nnnn* value so that the system IDs must be identical in order to match the SELECT or REJECT statement.  
  You can use S as an alias for SAME.                                                                                                   |
| UNDEFINED/U     | Can replace either of the SYSID *nnnn* values so that any undefined system IDs will match the SELECT or REJECT statement.  
  You can use U as an alias for UNDEFINED.                                                                                              |

### TMEMBER/MEMBER keyword

Use the TMEMBER keyword to select or reject messages that are destined to OTMA transactions that are identified by the specified TMEMBER names. Consider the following when specifying the TMEMBER keyword:

- You can use MEMBER as an alias for TMEMBER.
- You can specify a list of TMEMBERs or patterns for the TMEMBER keyword value.
- Use TMEMBER and TPIPE together to identify a specific OTMA destination.
- Message Advisor assumes DESTYpe=OTMA when you specify the TMEMBER keyword.
Only OTMA destinations with a matching TMEMBER name will match the SELECT or REJECT statement.

Table 366 on page 366 lists the TMEMBER keyword parameter.

Table 366: TMEMBER keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(m...m...)</td>
<td>the TMEMBER names that identify the OTMA destinations for which you want to select or reject messages for processing. The TMEMBER name can be up to 16 characters. <strong>Note:</strong> <code>m...m</code> is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

TPIPE/PIPE keyword

Use the TPIPE keyword to select or reject messages that are destined to OTMA transactions that are identified by the specified TPIPE names. Consider the following when specifying the TPIPE keyword:

- You can use PIPE as an alias for TPIPE.
- You can specify a list of TPIPEs or patterns for the TPIPE keyword value.
- Use TPIPE and TMEMBER together to identify a specific OTMA destination.
- Message Advisor assumes `DESType=OTMA` when you specify the TPIPE keyword.
- Only OTMA destinations with a matching TPIPE name will match the SELECT or REJECT statement.

Table 367 on page 366 lists the TPIPE keyword parameter.

Table 367: TPIPE keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(m...m...)</td>
<td>the TPIPE names that identify the OTMA destinations for which you want to select or reject messages for processing. <strong>Note:</strong> <code>m...m</code> is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>
**TPName keyword**

Use the TPName keyword to select or reject messages that are destined to APPC LU 6.2 transaction programs. You can specify a list of TPNAMEs for the TPName keyword value.

Use TPName and LUname together to identify a specific APPC destination.

*Table 368 on page 367 lists the TPName keyword parameter.*

**Table 368: TPName keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mm...mm,...)</td>
<td>the TPNAMEs for which you want to select or reject messages for processing. The TPNAME can be up to 64 characters. Note: mm...mm is a value that you specify.</td>
</tr>
</tbody>
</table>

**TRANSACTION keyword**

Use the TRANSACTION keyword to select or reject messages that are destined to specific transactions. You can specify a list of transactions for the TRANSACTION keyword value.

*Table 369 on page 367 lists the TRANSACTION keyword parameter.*

**Table 369: TRANSACTION keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mmmmmmmm,...)</td>
<td>the transactions for which you want to select or reject messages for processing. The transaction can be local or remote. Note: mmmmmmm is a value that you specify. You can specify a list of values or patterns.</td>
</tr>
</tbody>
</table>

**UOWID keyword**

Use the UOWID keyword to select or reject messages by UOWID. You can specify a list of 32-byte hexadecimal values for the UOWID keyword value. Consider the following when specifying the UOWID keyword:
- For a message to match, the UOWID must exactly match the UOWID of the message.

- The UOWID keyword is mutually exclusive with the OIMS, OTOKEN, PIMS, and PTOKEN keywords (which combine to form a UOWID).

- The UOWID is supposed to be unique in the sysplex, but requires scanning a queue to find a match. If you specify the UOWID keyword, BMC Software recommends that you also specify the DESTination and QTYPE keywords and indicate a single destination on a single queue. Otherwise, Message Advisor for IMS has to scan all queues to find a match.

Note
See “SELECT and REJECT subcommands for shared queues” on page 389 for more information about the QTYPE keyword.

Table 370 on page 368 lists the UOWID keyword parameter.

Table 370: UOWID keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(xx...xx,...)</td>
<td>the UOWID(s) for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td><strong>Note:</strong> xx...xx is a value that you specify.</td>
<td></td>
</tr>
</tbody>
</table>

**USERID keyword**

Use the USERID keyword to select or reject messages that contain a security segment with a specific user ID. Consider the following when specifying the USERID keyword:

- You can specify a list of user IDs or patterns for the USERID keyword value.

- Only messages that contain a security segment with one of the user IDs that you specify will match the SELECT or REJECT statement.

- If you specify the USERID keyword, messages that do not contain a security segment will not match the SELECT or REJECT statement, even for USERID=*. Therefore, you can specify USERID=* to select or reject all messages with a security segment.

Table 371 on page 369 lists the USERID keyword parameter.
Table 371: USERID keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>(mmmmmmmm,...)</code></td>
<td>the user IDs for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td>Note: <code>mmmmmmmm</code> is a value that you specify. You can specify a list of</td>
</tr>
<tr>
<td></td>
<td>values or patterns.</td>
</tr>
</tbody>
</table>

VSPcnt keyword

Use the VSPcnt keyword to select or reject messages that are destined to VTAM subpool LTERMs. You can specify a list of VSPCNTs for the VSPcnt keyword value.

Table 371 on page 369 lists the VSPcnt keyword parameter.

Table 372: VSPcnt keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>(mmmmmmmmmm,...)</code></td>
<td>the VSPCNTs for which you want to select or reject messages for processing</td>
</tr>
<tr>
<td></td>
<td>Note: <code>mmmmmmmmmm</code> is a value that you specify. You can specify a list of</td>
</tr>
<tr>
<td></td>
<td>values or patterns.</td>
</tr>
</tbody>
</table>
Message Advisor shared queues command sets

This chapter describes the Message Advisor for IMS command sets as they are used in a shared queues (SQ) environment, and also provides sample command sets. For more information about building and executing the command sets in an SQ environment, see the Message Advisor for IMS User Guide.

DEQUEUE command for shared queues

Message Advisor’s DEQUEUE command lets you dequeue (delete) messages in an SQ environment.

When used in an SQ environment, the command functions the same as it does for non-shared queues. See “DEQUEUE command” on page 177 for detailed information.

For more information about building and executing an SQ DEQUEUE command set, see the Message Advisor for IMS User Guide.

Sample SQ DEQUEUE command sets

Message Advisor provides sample SQ DEQUEUE command sets that can serve as templates when building your own SQ DEQUEUE command sets. The MAQSAMP library contains the sample command sets.

Table 373 on page 372 identifies the library members that contain the sample SQ DEQUEUE command sets.
Table 373: Sample SQ DEQUEUE command sets

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMC@DEQ1</td>
<td>dequeues all messages from a single destination</td>
</tr>
<tr>
<td>QMC@DEQ2</td>
<td>dequeues selected messages from multiple destinations</td>
</tr>
<tr>
<td>QMC@DEQ3</td>
<td>dequeues a specific message</td>
</tr>
</tbody>
</table>

For more information about building and executing an SQ DEQUEUE command set, see the *Message Advisor for IMS User Guide*.

**SQ DEQUEUE command set description**

Only the DEQUEUE keywords that apply to shared queues are described in this section.

For a complete description of the DEQUEUE command set, see “DEQUEUE command” on page 177.

**LOG keyword**

LOG is an optional keyword for the DEQueue command and indicates whether to log DEQueue activities.

Table 374 on page 372 lists the LOG keyword parameters.

Table 374: LOG keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>CQS is used to dequeue messages with normal logging</td>
</tr>
<tr>
<td>NO</td>
<td>messages are dequeued directly from the coupling facility structure with no logging, which means that a structure recovery will recover the dequeued messages</td>
</tr>
</tbody>
</table>

**SCOpe keyword**

SCOpe is an optional keyword for the DEQueue command and limits dequeue processing to destinations that match the specified scope. You can specify a list of values for the SCOpe keyword.

Table 375 on page 373 lists the SCOpe keyword parameters. Figure 71 on page 373 illustrates the queues that are processed for each parameter.
### Table 375: SCOpe keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLQ</td>
<td>includes both local and shared queues in dequeue processing</td>
</tr>
</tbody>
</table>
| LOCALQ    | limits dequeue processing to local queues  
  **Note:** You can use LCLQ as an alias for LOCALQ. |
| OVERFLOWQ | limits dequeue processing to the overflow shared queues  
  **Note:** You can use OVFQ as an alias for OVERFLOWQ. |
| PRIMARYQ  | limits dequeue processing to the primary shared queues  
  **Note:** You can use PRIQ as an alias for PRIMARYQ. |
| SHAREDQ   | limits dequeue processing to shared queues (both primary and overflow)  
  **Note:** You can use SHRQ as an alias for SHAREDQ.  
  SHAREDQ is mutually exclusive with PRIMARYQ and OVERFLOWQ. |

### Figure 71: SCOpe keyword parameters and associated queues processed

```
SCOPE=

ALLQ

SHAREDQ  SHRQ

LOCALQ  LCLQ

PRIMARYQ  PRIQ

OVERFLOWQ  OVFQ
```
**DISPLAY command for shared queues**

Message Advisor’s DISPLAY command lets you display messages in an SQ environment.

When used in an SQ environment, the command functions the same as it does for non-shared queues. See the section titled Display for detailed information.

For more information about building and executing an SQ DISPLAY command set, see the *Message Advisor for IMS User Guide*.

**Sample SQ DISPLAY command sets**

Message Advisor provides sample SQ DISPLAY command sets that can serve as templates when building your own SQ DISPLAY command sets. The MAQSAMP library contains the sample command sets.

Table 376 on page 374 identifies the library members that contain the sample SQ DISPLAY command sets.

**Table 376: Sample SQ DISPLAY command sets**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMC@DSP1</td>
<td>displays message queue statistics</td>
</tr>
<tr>
<td>QMC@DSP2</td>
<td>displays destination queue information</td>
</tr>
<tr>
<td>QMC@DSP3</td>
<td>displays destination information</td>
</tr>
<tr>
<td>QMC@DSP4</td>
<td>displays record information</td>
</tr>
</tbody>
</table>

*Note*

Many of the samples include field-level prompts (e.g., IMSID=?). When you execute a command set through the Message Advisor Interactive System Productivity Facility (ISPF) interface, you will be prompted to provide a valid value for any field that contains a prompt.

If you execute a command set outside of the Message Advisor ISPF interface, you will not be prompted to replace the ?. Therefore, you must remember to replace all ? with a valid value before submitting the request.

The following sample command sets are similar, but not identical to, the samples provided in MAQSAMP.
Figure 72 on page 375 shows a command set requesting a detailed report of destinations with messages. The command set requests a report for all destinations starting with $R61P.

**Figure 72: Command set: Display detailed destination report**

```
DISPLAY IMSID=R61P,TYPE=DESTINATION
   SELECT DESTINATION=$R61P*
END
```

In Figure 73 on page 375, the first command set requests a display of destination queues for destinations that begin with MSC and end in 2. The second command set requests a display of destination queues for TRAN1, APPC, and OTMA destinations. The third command set requests a display of destination queues for the staging destinations (Q2 and Q6). MAQSAMP(QMC@DSP2) contains a sample command set similar to Figure 73 on page 375.

**Figure 73: Command set: Display summary destination queues report**

```
DISPLAY IMSID=PR7B,TYPE=DEST_QUEUES
   SELECT DESTINATION=MSC+2
END
DISPLAY IMSID=PR7B,TYPE=DEST_QUEUES
   SELECT QTYPE=(Q1,Q7,Q9)
END
DISPLAY IMSID=PR7B,TYPE=DEST_QUEUES
   SELECT QTYPE=(Q2,Q6)
END
```

Figure 74 on page 375 shows a command set requesting a message record prefix and text in hexadecimal format. For this sample, the ENTRYID was determined from a previously-executed Message Advisor DISPLAY command. MAQSAMP(QMC@DSP4) contains a sample command set similar to Figure 74 on page 375.

**Figure 74: Command set: Display a message record**

```
DISPLAY IMSID=PR7B,TYPE=RECORD,
   ENTRYID=02000507BA23FCAB3532F0A7
END
```

Figure 75 on page 375 shows a command set requesting message queue statistics. MAQSAMP(QMC@DSP1) contains a sample command set similar to Figure 75 on page 375.

**Figure 75: Command set: Display message queue statistics**

```
DISPLAY IMSID=R61P,TYPE=STATISTICS
END
```

For more information about building and executing an SQ DISPLAY command set and for information about using the SQ DISPLAY command to create reports, see the *Message Advisor for IMS User Guide.*
SQ DISPLAY TYPE=DESTINATION command set description

Only the DISPLAY TYPE=DESTINATION keywords that apply to shared queues are described in this section.

For a complete description of the DISPLAY TYPE=DESTINATION command set, see the section titled Display.

SCOpe keyword

SCOpe is an optional keyword for the DISPLAY TYPE=DESTINATION command and limits display processing to destinations that match the specified scope. You can specify a list of values for the SCOpe keyword.

Table 377 on page 376 lists the SCOpe keyword parameters. Figure 71 on page 373 illustrates the queues that are processed for each parameter.

Table 377: SCOpe keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLQ</td>
<td>includes both local and shared queues in display processing</td>
</tr>
<tr>
<td>LOCALQ</td>
<td>limits display processing to local queues</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can use LCLQ as an alias for LOCALQ.</td>
</tr>
<tr>
<td>OVERFLOWQ</td>
<td>limits display processing to the overflow shared queues</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can use OVFQ as an alias for OVERFLOWQ.</td>
</tr>
<tr>
<td>PRIMARYQ</td>
<td>limits display processing to the primary shared queues</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can use PRIQ as an alias for PRIMARYQ.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| SHAREDQ   | limits display processing to shared queues (both primary and overflow)  
Note: You can use SHRQ as an alias for SHAREDQ.  
SHAREDQ is mutually exclusive with PRIMARYQ and OVERFLOWQ. |

**Figure 76: SCOpe keyword parameters and associated queues processed**

**SQ DISPLAY TYPE=DEST_QUEUES command set description**

Only the `DISPLAY TYPE=DEST_QUEUES` keywords that apply to shared queues are described in this section.

For a complete description of the `DISPLAY TYPE=DEST_QUEUES` command set, see the section titled Display.
**SCOpe keyword**

SCOpe is an optional keyword for the `DISPLAY TYPE=DEST_Queues` command and limits display processing to destinations that match the specified scope. You can specify a list of values for the SCOpe keyword.

Table 378 on page 378 lists the SCOpe keyword parameters. Figure 71 on page 373 illustrates the queues that are processed for each parameter.

**Table 378: SCOpe keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLQ</td>
<td>includes both local and shared queues in display processing</td>
</tr>
<tr>
<td>LOCALQ</td>
<td>limits display processing to local queues</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can use LCLQ as an alias for LOCALQ.</td>
</tr>
<tr>
<td>OVERFLOWQ</td>
<td>limits display processing to the overflow shared queues</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can use OVFQ as an alias for OVERFLOWQ.</td>
</tr>
<tr>
<td>PRIMARYQ</td>
<td>limits display processing to the primary shared queues</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can use PRIQ as an alias for PRIMARYQ.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| SHAREDQ   | limits display processing to shared queues (both primary and overflow)  
**Note:** You can use SHRQ as an alias for SHAREDQ.  
SHAREDQ is mutually exclusive with PRIMARYQ and OVERFLOWQ. |

**Figure 77: SCOpe keyword parameters and associated queues processed**

**SCOPE=**

- **ALLQ**
  - **SHAREDQ**
  - **SHRQ**
  - **PRIMARYQ**
  - **PRIQ**
  - **OVERFLOWQ**
  - **OVFQ**
  - **LOCALQ**
  - **LCLQ**

**SQ DISPLAY TYPE=MESSAGE command set description**

Only the **DISPLAY TYPE=MESSAGE** keywords that apply to shared queues are described in this section.

For a complete description of the **DISPLAY TYPE=MESSAGE** command set, see the section titled Display.
**SCOpe keyword**

SCOpe is an optional keyword for the `DISPLAY TYPE=MESSAGE` command and limits display processing to destinations that match the specified scope. You can specify a list of values for the SCOpe keyword.

Table 379 on page 380 lists the SCOpe keyword parameters. Figure 71 on page 373 illustrates the queues that are processed for each parameter.

**Table 379: SCOpe keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLQ</td>
<td>includes both local and shared queues in display processing</td>
</tr>
</tbody>
</table>
| LOCALQ    | limits display processing to local queues  
**Note:** You can use LCLQ as an alias for LOCALQ. |
| OVERFLOWQ | limits display processing to the overflow shared queues  
**Note:** You can use OVFQ as an alias for OVERFLOWQ. |
| PRIMARYQ  | limits display processing to the primary shared queues  
**Note:** You can use PRIQ as an alias for PRIMARYQ. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| SHAREDQ   | limits display processing to shared queues (both primary and overflow)  
  **Note:** You can use SHRQ as an alias for SHAREDQ.  
  SHAREDQ is mutually exclusive with PRIMARYQ and OVERFLOWQ. |

**Figure 78: SCOpe keyword parameters and associated queues processed**

```plaintext
SCOPE=

ALLQ

SHAREDQ   SHRQ

LOCALQ   LCLQ

PRIMARYQ   PRIQ

OVERFLOWQ   OVFQ
```

**SQ DISPLAY TYPE=RECORD command set description**

Only the `DISPLAY TYPE=RECORD` keywords that apply to shared queues are described in this section.

For a complete description of the `DISPLAY TYPE=RECORD` command set, see the section titled Display.

**ENTRYID keyword**

ENTRYID is a required keyword for the `DISPLAY TYPE=RECORD` command. Use the ENTRYID keyword to display a specific shared queues record.
You can use the SCOpe keyword to indicate whether Message Advisor is to search the primary or the overflow structure. If SCOpe includes both the primary and the overflow structure, Message Advisor searches the primary structure first. If Message Advisor finds the entry ID, it displays that message. Message Advisor only searches the overflow structure if it does not find the entry ID on the primary structure.

Table 380 on page 382 lists the ENTRYID keyword parameter.

**Table 380: ENTRYID keyword parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxxxxxxxx</td>
<td>the specific shared queues record to be displayed</td>
</tr>
</tbody>
</table>

**Note:** xxxxxxxxxx is a value that you specify.

**SCOpe keyword**

SCOpe is an optional keyword for the Display TYPE=RECORD command and limits display processing to destinations that match the specified scope. You can specify a list of values for the SCOpe keyword.

Table 381 on page 382 lists the SCOpe keyword parameters. Figure 71 on page 373 illustrates the queues that are processed for each parameter.

**Table 381: SCOpe keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLQ</td>
<td>includes both local and shared queues in display processing</td>
</tr>
<tr>
<td>LOCALQ</td>
<td>limits display processing to local queues</td>
</tr>
<tr>
<td>OVERFLOWQ</td>
<td>limits display processing to the overflow shared queues</td>
</tr>
<tr>
<td>PRIMARYQ</td>
<td>limits display processing to the primary shared queues</td>
</tr>
</tbody>
</table>

**Note:** You can use LCLQ as an alias for LOCALQ. You can use OVFQ as an alias for OVERFLOWQ. You can use PRIQ as an alias for PRIMARYQ.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| SHAREDQ   | limits display processing to shared queues (both primary and overflow)  
**Note:** You can use SHRQ as an alias for SHAREDQ.  
SHAREDQ is mutually exclusive with PRIMARYQ and OVERFLOWQ. |

**Figure 79: SCOpe keyword parameters and associated queues processed**

```text
SCOPE=

ALLQ

SHAREDQ
SHRQ

PRIMARYQ
PRIQ

OVERFLOWQ
OVFQ

LOCALQ
LCLQ
```

**SQ DISPLAY TYPE=STATISTICS command set description**

There is no difference between shared and non-shared queues for the `DISPLAY TYPE=STATISTICS` command.

For a complete description of the `DISPLAY TYPE=STATISTICS` command set, see the section titled Display.

**SQ DISPLAY TYPE=ANALYZER command set description**

The `DISPLAY TYPE=ANALYZER` command applies to shared queues only.
For a complete description of the `DISPLAY TYPE=ANALYZER` command set, see the section titled Display.

**REQUEUE command for shared queues**

In an SQ environment, you can use the REQUEUE command with the UNLOAD and DEQUEUE commands to unload messages queued to a specific LTERM destination, and then later requeue the messages to the same or another LTERM.

See “REQUEUE command” on page 227 for detailed information.

For more information about building and executing an SQ REQUEUE command set, see the *Message Advisor for IMS User Guide*.

**Sample SQ REQUEUE command sets**

Message Advisor provides sample SQ REQUEUE command sets that can serve as templates when building your own SQ REQUEUE command sets. The MAQSAMP library contains the sample command sets.

Table 382 on page 384 identifies the library members that contain the sample SQ REQUEUE command sets.

**Table 382: Sample SQ REQUEUE command sets**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMC@REQ1</td>
<td>requeues selected messages within an interval</td>
</tr>
<tr>
<td>QMC@REQ2</td>
<td>selects/rejects messages</td>
</tr>
<tr>
<td>QMC@REQ3</td>
<td>changes affinity on transactions</td>
</tr>
</tbody>
</table>

*Note*

Many of the samples include field-level prompts (e.g., IMSID=?). When you execute a command set through the Message Advisor Interactive System Productivity Facility (ISPF) interface, you will be prompted to provide a valid value for any field that contains a prompt.

If you execute a command set outside of the Message Advisor ISPF interface, you will *not* be prompted to replace the ?. Therefore, you must remember to replace all ? with a valid value before submitting the request.
**Shared queues REQUEUE considerations**

In an SQ environment, IMS recovery processing and requeue considerations change dramatically.

Because queues no longer belong to individual IMS systems in an SQ environment, IMS SQ systems no longer dump message queues to the log on a SNAPQ, DUMPQ, or PURGE. As a result, these checkpoints are of little concern regarding SQ REQUEUE command set processing.

Because **REQUEUE TYPE=COLD** command sets do not have the same effect in an SQ environment, Message Advisor allows the **CONV=ALL** keyword or the **CONV=NONE** keyword on all **REQUEUE TYPE=EREFAIL** and **REQUEUE TYPE=REPROCESS** command sets.

The following sample command sets are similar (but not identical) to the samples provided in MAQSAMP.

**Figure 80 on page 385** shows a command set for requeuing selected messages during a specific interval. The sample is similar to the sample located in MAQSAMP(QMC@REQ1).

**Figure 80: Command set: REQUEUE TYPE=FILE during a specific interval**

```plaintext
REQUEUE IMSID=PR7B,TYPE=FILE,REPORT=(SUMM,DEST),CONV=ALL,MSGTYPE=ALL
   INPUT DSN=TME.RIP.MAQV2SQ.D2003275.ALLDEST
   INTERVAL START=2003275/0830126,STOP=2003275/1130455
END
```

In **Figure 80 on page 385**, Message Advisor will select messages of all types (provided that the messages fall within the specified interval) for the requeue operation. The MSGTYPE keyword defaults to ALL, so Message Advisor will requeue all messages. **TYPE=FILE** indicates that Message Advisor will use an unload file as input to the requeue. **CONV=ALL** indicates that if any conversations exist in the input file, Message Advisor will requeue them.

**Figure 81 on page 385** shows a command set for requeuing messages of QTYPE TRAN1. This sample is similar to the sample located in MAQSAMP(QMC@REQ2).

**Figure 81: Command set: REQUEUE TYPE=FILE for requeuing messages of QTYPE TRAN1**

```plaintext
REQUEUE IMSID=PR81,TYPE=FILE,REPORT=(SUMM,DEST)
   INPUT DSN=TME.RIP.MAQV2SQ.D2003725.ALLDEST
   SELECT QTYPE=(Q1)
   REJECT QTYPE=(ALLTERM)
END
```

In **Figure 81 on page 385**, Message Advisor will only select transactions from the ready queue for the requeue operation. Message Advisor will not requeue any serial or suspended transactions. The **REJECT** subcommand ensures that Message Advisor
will not requeue any messages that are related to LTERM types. Omitting the REJECT subcommand would provide the same results.

Figure 82 on page 386 shows a command set for requeueing messages of all transaction types. This sample is similar to the sample located in MAQSAMP(QMC@REQ3).

Figure 82: Command set: REQUEUE TYPE=FILE for requeueing messages of all transaction types

REQUEUE IMSID=PR81,TYPE=FILE,REPORT=(SUMM,DEST)
   INPUT DSN=TME.RIP.MAQV1SQ.D2003275.ALLDEST
   SEL QTYPE=(ALLTRAN)
   CHANGE NEWAFFINITY=&IMS
END

In Figure 82 on page 386, Message Advisor will requeue transactions of type TRAN1, TSERL4, and TSUSP3. Message Advisor will change the affinity of the selected messages to match the IMSID of the processing IMS.

For more information about building and executing an SQ REQUEUE command set, see the Message Advisor for IMS User Guide.

UNLOAD command for shared queues

The SQ UNLOAD command lets you unload messages from active IMS shared message queues to a data set, based on criteria that you specify. Message Advisor does not dequeue the messages that it unloads; instead, Message Advisor leaves the unloaded messages on the IMS message queues. The SQ DEQUEUE command’s MODE=UNLOAD DEQUEUE feature allows you to unload and then immediately dequeue messages.

For more information about building and executing an SQ UNLOAD command set, see the Message Advisor for IMS User Guide.

Sample SQ UNLOAD command sets

Message Advisor provides sample SQ UNLOAD command sets that can serve as templates when building your own SQ UNLOAD command sets.

The MAQSAMP library contains the sample command sets.

Table 383 on page 387 identifies the library members that contain the sample SQ UNLOAD command sets.
### Table 383: Sample SQ UNLOAD command sets

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMC@UNL1</td>
<td>unloads all messages from a single destination</td>
</tr>
<tr>
<td>QMC@UNL2</td>
<td>unloads messages from multiple destinations</td>
</tr>
</tbody>
</table>

The following sample command sets are similar, but not identical to, the samples provided in MAQSAMP.

**Figure 83 on page 387** shows a command set for unloading all messages in all queues from a specific destination. MAQSAMP(QMC@UNL1) contains a command set similar to Figure 83 on page 387.

The command set shown in Figure 83 on page 387 unloads all messages from destination $R61PP01. Because the command set does not include an OUTPUT statement, Message Advisor unloads the messages to the default unload data set.

**Figure 83: Command set: Unload from one destination**

```
UNLOAD IMSID=R61P,DESTINATION=$R61PP01
END
```

**Figure 84 on page 387** shows a command set for unloading all messages from multiple destinations using the FORCE keyword. MAQSAMP(QMC@UNL2) contains a sample command set similar to Figure 84 on page 387.

The command set shown in Figure 84 on page 387 unloads all messages from MSNAME destinations (QTYPE=MSN) beginning with $R61P (DESTINATION= $R61P*). Because the command set does not include an OUTPUT statement, Message Advisor unloads the messages to the default unload data set.

**Figure 84: Command set: Unload from multiple destinations**

```
UNLOAD IMSID=R61P,FORCE=YES
SELECT DESTINATION=$R61P*
SELECT QTYPE=MSN
END
```

For more information about building and executing an SQ UNLOAD command set, see the *Message Advisor for IMS User Guide*.

### SQ UNLOAD command set description

Only the UNLOAD keywords that apply to shared queues are described in this section.

For a complete description of the UNLOAD command set, see “UNLOAD command” on page 281.
**SCOpe keyword**

SCOpe is an optional keyword for the UNLoad command and limits unload processing to destinations that match the specified scope. You can specify a list of values for the SCOpe keyword.

Table 384 on page 388 lists the SCOpe keyword parameters. Figure 71 on page 373 illustrates the queues that are processed for each parameter.

**Table 384: SCOpe keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLQ</td>
<td>includes both local and shared queues in display processing</td>
</tr>
</tbody>
</table>
| LOCALQ    | limits display processing to local queues  
**Note:** You can use LCLQ as an alias for LOCALQ. |
| OVERFLOWQ | limits display processing to the overflow shared queues  
**Note:** You can use OVFQ as an alias for OVERFLOWQ. |
| PRIMARYQ  | limits display processing to the primary shared queues  
**Note:** You can use PRIQ as an alias for PRIMARYQ. |
**SELECT and REJECT subcommands for shared queues**

When used in an SQ environment, the SELECT and REJECT subcommands function the same as they do for non-shared queues with the exception of the following keywords, which only apply to shared queues:

- **AFFINITY**
- **ENTRYID** (mutually exclusive with the DRRN keyword for non-shared queues)
- **KEY**
- QTYPE (mutually exclusive with the DESTYPE and QUEUE keywords for non-shared queues)

See “SELECT and REJECT subcommands” on page 313 for descriptions of all other SELECT and REJECT keywords.

**Note**
You can specify the DESTYPE keyword when processing the shared queues to match only messages for destinations that are defined on the current IMS system (in most cases). Table 385 on page 390 lists the DESTYPE keyword parameters and their matching destinations for shared queues.

**Table 385: DESTYPE keyword parameters and matching destinations for shared queues**

<table>
<thead>
<tr>
<th>DESTYPE=</th>
<th>matches these destinations on shared queues:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>all queue types-regardless of whether the destination is locally defined</td>
</tr>
<tr>
<td>APPC/TPNAME</td>
<td>QTYPE 7-matches only if the APPC or TPNAME is defined on the current IMS</td>
</tr>
<tr>
<td>BMP</td>
<td>ALLTRAN-matches only BMPs that are defined on the current IMS</td>
</tr>
<tr>
<td>CNT</td>
<td>QTYPE 5-matches only CNTs that are on the current IMS</td>
</tr>
<tr>
<td>DEADQ</td>
<td>QTYPE 5-matches only DEADQ CNTs on the current IMS</td>
</tr>
<tr>
<td>DYNAMIC</td>
<td>QTYPE 5-matches only dynamic CNTs that are on the current IMS</td>
</tr>
<tr>
<td>HELD</td>
<td>none-matches only destinations on local queues</td>
</tr>
<tr>
<td>LTERMS</td>
<td>ALLTERM-matches only LTERMs that are defined on the current IMS</td>
</tr>
<tr>
<td>MPP</td>
<td>ALLTRAN-matches only MPPs that are defined on the current IMS</td>
</tr>
<tr>
<td>MSNAME</td>
<td>QTYPE 8-matches only MSNAMEs on the current IMS</td>
</tr>
<tr>
<td>OTMA</td>
<td>QTYPE 9-matches only if the OTMA is defined on the current IMS</td>
</tr>
<tr>
<td>RSMB</td>
<td>ALLTRAN-matches only RSMBs that are defined on the current IMS</td>
</tr>
<tr>
<td>SMB</td>
<td>ALLTRAN-matches only SMBs that are defined on the current IMS</td>
</tr>
<tr>
<td>STATIC</td>
<td>QTYPE 5-matches only static CNTs that are on the current IMS</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>none-matches only destinations on local queues</td>
</tr>
<tr>
<td>TRANSACTION</td>
<td>ALLTRAN-matches only transactions that are defined on the current IMS</td>
</tr>
<tr>
<td>UNDEFINED</td>
<td>all queue types-matches only if the destination is not defined on the current IMS</td>
</tr>
</tbody>
</table>

**Note**: Because UNDEFINED will match dynamic LTERMs that are not currently active, and may also match destinations that are defined on some other IMS, you should not use it by itself as a criteria for deleting messages.
AFFINITY keyword

Use the AFFINITY keyword to select or reject messages with a specific affinity. The AFFINITY keyword allows a seven-character affinity mask. Consider the following when specifying the AFFINITY keyword:

- The affinity is the IMSID for non-XRF systems. For XRF systems, the affinity is the first seven characters of the RSENAME.

- You can specify a list of affinities for the AFFINITY keyword value. The list can include the special values that are described in Table 386 on page 392. For example, `AFFINITY=(&NONE,IMSA)` returns only destinations with no affinity or with affinity to IMSA.

- If you do not specify the AFFINITY keyword, Message Advisor returns all destinations which otherwise match, with or without an affinity.

- You can specify `AFFINITY=&NONE` or `AFFINITY=' '` to return only destinations with no affinity.

- If you specify a mask for the AFFINITY keyword value, Message Advisor only returns destinations with an affinity that matches the mask. Message Advisor does NOT return destinations without an affinity (i.e., Message Advisor treats `AFFINITY=*` the same as it treats `AFFINITY=?*`).

- Local queue destinations (on non-shared queues systems or on shared queues systems) are considered to have no affinity and will only match a blank or a SELECT subcommand without an affinity.

- AFFINITY is a message-related keyword when used on a SELECT or REJECT subcommand with the REQueue command, but is a destination-related keyword for all other commands.

- If you specify either (but not both) &EXISTS or &ORPHAN, processing occurs as follows:
  - Message Advisor performs a FNDEST CREATE for destinations that it finds on the shared queues with an affinity for the current IMS.
  - If Message Advisor cannot find or create the destination, the destination matches &ORPHAN.

---

<table>
<thead>
<tr>
<th>DESTYPE=</th>
<th>matches these destinations on shared queues:</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSPCNT</td>
<td>QTYPE 5-matches only VSPCNTs on the current IMS</td>
</tr>
</tbody>
</table>
— If Message Advisor finds or creates the destination but the destination is not of the correct type for that queue, the destination matches &ORPHAN.

— For queue 7 and queue 9, the destination will also match &ORPHAN if the shared queues token does not match the IMS token.

— Destinations that exist or that Message Advisor can create as the correct type match &EXISTS.

The intent of the above processing is to allow finding messages on the shared queues which can no longer be processed, and are thus "orphaned."

Table 386 on page 392 lists the special parameters that are allowed for the AFFINITY keyword.

### Table 386: AFFINITY keyword special parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| mm...mm...mm | matches destinations with the specified affinity  
Note: mm...mm...mm is a value that you specify. You can specify a list of values or patterns. |
| &EXISTS       | matches only destinations with affinity to the current IMS and which CAN be created on the current IMS  
Note: &EXISTS is not allowed on SElect or REJect subcommands with the REQueue command. |
| &IMS          | matches only destinations with an affinity for the current IMS  
Note: &IMS is not allowed on SElect or REJect subcommands with the REQueue EXTRACT command.  
AFFINITY=(&EXISTS,&ORPHAN) is the same as AFFINITY=&IMS. |
| &NONE         | matches only destinations with no affinity                                                                                                                                                                   |
| &ORPHAN       | matches only destinations with affinity to the current IMS and which CANNOT be created on the current IMS  
Note: AFFINITY=&ORPHAN will not necessarily find all messages that cannot be processed. Messages without affinity will never match, since there may be another IMS that can process them even if the IMS is not currently active. There may also be LTERMs that can still be created by VT or ETO, which are no longer used. You may need to specify a time interval to locate these messages and LTERMs.  
&ORPHAN is not allowed on SElect or REJect subcommands with the REQueue command. |
ENTRYID keyword

ENTRYID is an optional keyword for the SESelect and REJect subcommands. You can specify a single 12-byte hexadecimal value to indicate the list entry ID for a shared queues message. Consider the following when specifying the ENTRYID keyword:

- The ENTRYID is the shared queues equivalent of a DRRN and the ENTRYID keyword is mutually exclusive with the DRRN keyword.
- You can specify the ENTRYID keyword on SESelect and REJect subcommands with the DISplay, DEQueue, and UNLoad commands, but not with the REQueue command.

Table 387 on page 393 lists the ENTRYID keyword parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxxxxxxxx</td>
<td>the list entry ID for a shared queues message</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td><strong>xxxxxxxxxxx</strong> is a value that you specify.</td>
</tr>
</tbody>
</table>

KEY keyword

KEY is an optional keyword for the SESelect and REJect subcommands and can be used to specify a single 16-byte shared queue key in hexadecimal. Consider the following when specifying the KEY keyword:

- The KEY keyword is mutually exclusive with all other destination-related keywords.
- Message Advisor for IMS uses the first and last byte of the key to determine the queue type.
- If the first byte has a value of X'01' to X'0B', Message Advisor for IMS uses the value as a client queue type. If the first byte has a value of X'40' or greater, Message Advisor for IMS uses the last byte to determine the private queue type. Other values for the first byte are invalid.
- For private queue types, when the last byte has a value of 'X01' through 'X0B' or 'X81' through 'X8B', Message Advisor for IMS uses the last nibble as the private queue type. Otherwise, the key is assumed to be for the control queue.
You can specify the KEY keyword on SELECT and REJECT subcommands with the DISPLAY, DEQueue, and UNLoad commands, but not with the REQueue command.

**QTYPE keyword**

QTYPE is an optional keyword for the SELECT and REJECT subcommands and limits processing to specific queues. You can specify a list of queue types with the QTYPE keyword. Consider the following when specifying the QTYPE keyword:

- If you only specify client queue types for the QTYPE keyword on a SELECT or REJECT subcommand, the DEQueue, UNLoad, and DISPLAY commands will search only the indicated queues for matching messages. For the REQueue command, the MSGRSQTY field in the message prefix determines the queue type for filtering.

- When a private queue keyword (COLDQ or LOCKQ) is used with QTYPE by itself or in combination with client queue types, only messages which are currently on the private queue AND which were originally on one of the listed client queues will match. For example, QTYPE=(COLDQ, SERIAL) will display only serial transactions on the cold queue. You must use two SELECT subcommands to select both the COLDQ and the SERIAL client queue.

- If you specify the QTYPE keyword on a SELECT or REJECT subcommand with a DISPLAY, DEQueue, or UNLoad command for non-shared queues, the subcommand will match nothing and an error may result.

- If you specify the QTYPE keyword on a SELECT or REJECT subcommand with a REQueue command, filtering will be based on the queue type in the message (if any). Because the queue type in the message can only be a client queue type, the COLDQ and LOCKQ parameters will not match. Also, messages that were created on a non-shared queues system will have a queue type of zero and will not match.

- When specified on SELECT or REJECT subcommands with the REQueue command, QTYPE is used only to select messages. You can requeue messages to any queue, regardless of the QTYPE keyword.

Table 388 on page 394 lists the QTYPE keyword parameters. Figure 86 on page 397 illustrates the queues that are processed for each parameter.

**Table 388: QTYPE keyword parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLDEST</td>
<td>processes all queues that are listed for the ALLTRAN and ALLTERM parameters</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Only client queues will match unless you explicitly select private queues.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| **ALLTERM** | limits processing to queues 5, 7, 8, and 9  
**Note:** You can use CNT as an alias for ALLTERM.  
Only client queues will match unless you explicitly select private queues. |
| **ALLTRAN** | limits processing to queues 1, 3, and 4  
**Note:** You can use SMB as an alias for ALLTRAN.  
Only client queues will match unless you explicitly select private queues. |
| **COLDQ** | private queue  
**Note:** Private queue types are not allowed on SESelect and REJect subcommands with the REQueue command.  
When you specify a private queue type for the QTYPE keyword by itself or in combination with a client queue type, only messages which are currently on the private queue AND which were originally on one of the listed client queues will match. For example, QTYPE=(COLDQ, SERIAL) will only display serial transactions on the cold queue. You must use two SELECT subcommands to select both the COLDQ and the SERIAL client queue. |
| **ControlQ** | private queue  
**Note:** Private queue types are not allowed on SESelect and REJect subcommands with the REQueue command.  
The control queue does not contain IMS messages, but contains CQS control information. You can use the DISelect commands to display these control elements. However, you cannot use other keywords except for the DESTination keyword, which you can use to select the key.  
**Note:** You can use CNTLQ as an alias for ControlQ. |
| **LOCKQ** | private queue  
**Note:** Private queue types are not allowed on SESelect and REJect subcommands with the REQueue command.  
When you specify a private queue type for the QTYPE keyword by itself or in combination with a client queue type, only messages which are currently on the private queue AND which were originally on one of the listed client queues will match. For example, QTYPE=(LOCKQ, SERIAL) will only display serial transactions on the lock queue. You must use two SELECT subcommands to select both the LOCKQ and the SERIAL client queue. |
| **Q1** | limits processing to queue 1  
**Note:** You can use 1, TRAN1, TRANRDY, and TRANSACT as aliases for Q1. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Q2        | limits processing to queue 2 (a staging queue)  
  *Note:* The staging queue types are not allowed on the REQueue command.  
  The staging queues contain continuation segments for IMS messages, but not the beginning of the message. You cannot use destination- or message-related selection keywords when selecting the staging queues.  
  *Note:* You can use 2 as an alias for Q2. |
| Q3        | limits processing to queue 3  
  *Note:* You can use 3, TSUSP3, and SUSPEND as aliases for Q3. |
| Q4        | limits processing to queue 4  
  *Note:* You can use 4, TSERL4, and SERIAL as aliases for Q4. |
| Q5        | limits processing to queue 5  
  *Note:* You can use 5, LTERM5, and LTRM5 as aliases for Q5. |
| Q6        | limits processing to queue 6 (a staging queue)  
  *Note:* The staging queue types are not allowed on the REQueue command.  
  The staging queues contain continuation segments for IMS messages, but not the beginning of the message. You cannot use destination- or message-related selection keywords when selecting the staging queues.  
  *Note:* You can use 6 as an alias for Q6. |
| Q7        | limits processing to queue 7  
  *Note:* You can use 7, APPC7, and TPNAME7 as aliases for Q7. |
| Q8        | limits processing to queue 8  
  *Note:* You can use 8, REMOTE8, and RMT8 as aliases for Q8. |
| Q9        | limits processing to queue 9  
  *Note:* You can use 9 and OTMA9 as aliases for Q9. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAGEQ</td>
<td>limits processing to queues 2 and 6&lt;br&gt;&lt;strong&gt;Note:&lt;/strong&gt; The staging queue types are not allowed on the REQueue command. The staging queues contain continuation segments for IMS messages, but not the beginning of the message. You cannot use destination- or message-related selection keywords when selecting the staging queues.</td>
</tr>
</tbody>
</table>
SELECT and REJECT subcommands for shared queues
COND pseudo command

This appendix describes the Message Advisor for IMS COND pseudo command, and also provides a syntax diagram and sample command set. For more information about adding COND command steps to Message Advisor requests, see the Message Advisor for IMS User Guide.

Overview

Using Interactive System Productivity Facility (ISPF) EDIT, you can build a Message Advisor request that consists of any combination of DEQUEUE, DISPLAY, REQUEUE, and UNLOAD command sets. You can accomplish multiple tasks with one request by including multiple command sets in the request. For example, you can issue one request that contains multiple command sets in order to unload, dequeue, and requeue all messages as a single task.

You can exercise additional control over a Message Advisor request that contains multiple command sets by assigning conditional steps to one or more of the command sets. Conditional steps let you perform the following tasks:

- request that Message Advisor execute a command set only if it successfully executes a previous command set

—OR—

- request that Message Advisor execute a command set only if it does not successfully execute a previous command set

When you assign conditional steps to a Message Advisor request, a COND pseudo command step is inserted between each command set or a series of command sets within the request. A COND step establishes a label (also called a stepname) for forward branching or for referencing purposes when comparing Message Advisor command set results.
COND command step syntax

The following figure shows the COND command step syntax.

**Figure 87: COND command step syntax**

---

Sample Message Advisor request with COND command steps

The following figure shows a sample request with DISPLAY, UNLOAD, DEQUEUE, and REQUEUE command sets. The request also contains conditional steps that specify the conditions under which Message Advisor executes each command set.
Figure 88: Sample Message Advisor request with COND command steps

COND LABEL=STEP1
  DISPLAY IMSID=PEL5,TYPE=DESTINATION
  SELECT DESTINATION=PRT100T,QUEUE=ALL,DESTYPE=ALL
  SELECT DESTINATION=PRT200T,QUEUE=ALL,DESTYPE=ALL
END

COND LABEL=STEP2
  DISPLAY IMSID=PEL5,TYPE=DESTINATION
  SELECT DESTINATION=PRT300T,QUEUE=ALL,DESTYPE=ALL
  SELECT DESTINATION=PRT400T,QUEUE=ALL,DESTYPE=ALL
END

COND LABEL=STEP3,IFCOND='STEP1.GT.0',BRANCH=STEP4
  DEQUEUE IMSID=PEL5,MODE=UNLOAD_DEQUEUE,FORCE=YES
  OUTPUT DSNNAME=BMC.QMR.UNLOAD
  SELECT DESTINATION=PRT100T,QUEUE=ALL,DESTYPE=ALL
  SELECT DESTINATION=PRT200T,QUEUE=ALL,DESTYPE=ALL
END

COND LABEL=STEP4,IFCOND='STEP2.EQ.0',BRANCH=FLUSH
  REQUEUE IMSID=PEL5,TYPE=FILE
  INPUT DSNNAME=BMC.QMR.UNLOAD2
  SELECT DESTINATION=PRT300T,QUEUE=ALL,DESTYPE=ALL
  SELECT DESTINATION=PRT400T,QUEUE=ALL,DESTYPE=ALL
END

COND command description

The COND command set consists of the following items:

- COND primary command with associated keywords and parameters
- END command
Note

Keyword and parameter descriptions are listed in alphabetical order: first, by keyword; then, by parameter within each keyword. Optional portions of commands, keywords, and parameters are shown in lowercase characters; defaults are underlined and listed first. Items in italic type represent variables that you must replace with a site-specific name or value.

BRANCH keyword

BRANCH is an optional keyword for the COND command. BRANCH determines the action that Message Advisor takes when the IFCOND in the same COND step returns a logical true condition.

Table 389 on page 402 lists the BRANCH keyword parameters.

Table 389: BRANCH keyword parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| stepname  | the value assigned to the LABEL keyword for the step that is to follow the current step  
**Note:** A backward branch results in a syntax error message. |
| FLUSH     | Message Advisor ignores all remaining steps |
| SKIPNEXT  | Message Advisor bypasses the next real command set |

IFCOND keyword

IFCOND is an optional keyword for the COND command. IFCOND determines whether Message Advisor processes the BRANCH and MAXRC keywords.

Table 390 on page 403 lists the IFCOND keyword parameter.
Table 390: IFCOND keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| 'field1.oper.field2' | when an IFCOND returns a logical true condition, Message Advisor processes the BRANCH and MAXRC keywords; when an IFCOND returns a false condition, Message Advisor ignores the BRANCH and MAXRC keywords.  
  field1 and field2 are either a preceding stepname (LABEL, MAXRC, or BYPASSED) or a numeric value.  
  .oper. is one of the following logical operators:  
  - .EQ. (equal)  
  - .NE. (not equal)  
  - .GT. (greater than)  
  - .GE. (greater than or equal)  
  - .LT. (less than)  
  - .LE. (less than or equal)  
  Note: If you do not specify the IFCOND keyword, Message Advisor processes BRANCH and MAXRC. |

LABEL keyword

LABEL is an optional keyword for the COND command. LABEL assigns a stepname for forward branching or for referring back to the return code value of a real Message Advisor step that follows the COND pseudo step. The return code for a bypassed step has an internal value of 1; however, MAXRC is not changed from 0 to 1 when Message Advisor bypasses a step.

Table 391 on page 403 lists the LABEL keyword parameter.

Table 391: LABEL keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| stepname  | the stepname for forward branching or for referring back to the return code value of a real Message Advisor step that follows the COND pseudo step.  
  Note: stepname is a value that you specify. |
MAXRC keyword

MAXRC is an optional keyword for the COND command. MAXRC indicates the highest return code for all command sets executed so far.

Table 392 on page 404 lists the MAXRC keyword parameter.

Table 392: MAXRC keyword parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stepname or numeric value</td>
<td>the highest return code for all command sets executed so far</td>
</tr>
<tr>
<td></td>
<td>You can reference MAXRC in the IFCOND keyword value, and you can use the</td>
</tr>
<tr>
<td></td>
<td>MAXRC keyword to explicitly set a lower or higher value. You can assign a</td>
</tr>
<tr>
<td></td>
<td>numeric value or a preceding stepname to MAXRC.</td>
</tr>
</tbody>
</table>

**Note:** The MAXRC keyword value cannot exceed 4096.

When a Message Advisor Server executes in batch mode, the last value of MAXRC determines the step termination code. In an ISPF invocation of Message Advisor command steps, the online interface reports the highest return code found; however, the logic flow of the steps is the same as if the request was executed in batch mode.
Message Advisor for IMS Application Interface

This appendix describes the Message Advisor for IMS Application Interface (API) feature, which identifies how Message Advisor communicates its data communications-related requests to IMS.

Overview

The Message Advisor API feature allows for programmatic execution, control, and interrogation of results from Message Advisor DISPLAY and IMSCMD command sets transmitted to an IMS system. The API interface only supports the DISPLAY and IMSCMD command sets.

DISPLAY command output can be generated in either an internal or an external format. The $QMRAPID macro in the MAQSAMP data set maps the formats. Other related MAQSAMP members are the $QMRAPI macro and the QMRAPIX sample program.

Function calls

The QMRAPIX program invokes the $QMRAPI macro to make function calls to module QMRAPIX0, which must be included in the link edit.

The function calls are as follows:

- INIT

  INIT ensures that a Message Advisor Server task of a given parameter name exists. If the server name does not exist (default server task name is QMRAPI00), program QMRAPIX0 issues an internal start command for the default or for the specified PROCNAME parameter. An executing batch job with the corresponding jobname will also suffice. Communication between a Message Advisor server and
the requesting address space uses cross-memory post and XCSA memory rather than Virtual Telecommunications Access Method (VTAM).

- **EXEC**
  EXEC passes the command set character string to the server.

- **GET**
  GET calls a series of functions that can be used to retrieve the normal output, which is in either an internal or an external format.

- **TERM**
  TERM is an optional function that logically terminates the API.

MAQCNTL member QMR#APIX executes the QMRAPIX sample program, and MAQCNTL member QMR#APIA assembles and links the program.

The MAQSAMP members QMRAPI and $QMRAPI and the MAQCNTL members QMR#APIX and QMR#APIA provide more information.

### Command definitions

The command definitions are as follows:

- **DISPLAY**
  DISPLAY generates a display command, which may be used by an EXEC call.

- **INCLUDE/EXCLUDE**
  INCLUDE/EXCLUDE generates an INCLUDE or EXCLUDE subcommand for a DISPLAY command.

- **END**
  END terminates a Message Advisor command.

- **IMSCMD**
  IMSCMD submits an IMS command to the IMS command processor.

### QMRAPI function call descriptions

Each of the QMRAPI function calls has associated keywords that let you define how the call is executed.
The function calls and their associated keywords are described in the topics of this section.

**QMRAPI INIT function call description**

The QMRAPI INIT function call initializes Message Advisor API control blocks, starts a Message Advisor Server if necessary, and prepares for subsequent EXEC function calls. Keyword descriptions for the INIT function call appear in this section in alphabetical order.

**ERROR keyword**

ERROR indicates the address of an instruction that will receive control if an error occurs (R15 greater than 0).

**MF keyword**

MF indicates a macro instruction. To generate reentrant code, specify the macro instruction twice: once with \texttt{MF=L} to generate the list form, and once with \texttt{MF=(E,address)} to generate the execute form. The \texttt{address} variable is the RX type address of the list form.

Following are the possible return values:

- R15=0: successful return
- R15=4: parameter error
- R15=8: \texttt{SERVER\_ID} Error
- R15=12: Message Advisor API already initialized
- R15=16: 'S \texttt{PROCNAME}' failed or did not become active within a one-minute elapsed time interval

**PROCNAME keyword**

PROCNAME indicates an RX type address or register. The address or register points to an 8-byte storage area that contains the system server task name that will be posted to service EXEC requests. If you specify less than five characters for the PROCNAME keyword value, QMRAPI00 is the default. This procedure will be internally started if it is not currently active.
QMRAPI EXEC function call description

The QMRAPI EXEC function call executes a Message Advisor command (DISPLAY or IMSCMD) and returns the output records and BMC messages. Keyword descriptions for the EXEC function call appear in this section in alphabetical order.

EIB keyword

EIB is required for the EXEC function call. EIB indicates the address of an RX type address or register. The address or register points to an EXEC interface block, which must be constructed as follows:

<table>
<thead>
<tr>
<th>RG DEC</th>
<th>ORG HEX</th>
<th>LENGTH</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>+0</td>
<td>+0</td>
<td>4</td>
<td>command length</td>
</tr>
<tr>
<td>+4</td>
<td>+4</td>
<td>4</td>
<td>command address</td>
</tr>
<tr>
<td>+8</td>
<td>+8</td>
<td>4</td>
<td>command address</td>
</tr>
<tr>
<td>+8</td>
<td>+8</td>
<td>4</td>
<td>output format</td>
</tr>
<tr>
<td>+12</td>
<td>+C</td>
<td>4</td>
<td>message length</td>
</tr>
<tr>
<td>+16</td>
<td>+10</td>
<td>4</td>
<td>message address</td>
</tr>
</tbody>
</table>

The command length and address fields describe a storage area that must contain the input command. The $QMRAPI DISPLAY function can create this command area.

The output format field must contain a 1 or a 2 (F'1' or F'2'). F'1' indicates internal output. The $QMRAPI macro can map output blocks created with the F'1' format. F'2' indicates external output. EBCDIC formats output lines created with F'2'.

The message length and message address fields must describe an output storage area that will be filled with any messages that Message Advisor generates. Because all messages will be expanded or truncated to 80 bytes, the length should be a multiple of 80.

ERROR keyword

ERROR indicates the address of an instruction that will receive control if an error occurs (R15 greater than 0). If you specify both the ERROR and WARNING keywords, the ERROR address only receives control when R15 is greater than or equal to 8.
**MF keyword**

MF indicates a macro instruction. To generate reentrant code, specify the macro instruction twice: once with \( MF=L \) to generate the list form, and once with \( MF=(E, address) \) to generate the execute form. The \( address \) variable is the RX type address of the list form.

Following are the possible return values:

- R15=0: successful return
- R15=4: warning message returned or no output available
- R15=8: error message returned
- R15=12: Message Advisor API task terminated
- R15=16: input command syntax error
- R15=20: Message Advisor DISPLAY syntax error
- R15=24: Message Advisor DISPLAY abended
- R15=28: Message Advisor API was not initialized
- R15=32: request was cancelled

**WARNING keyword**

WARNING indicates the address of an instruction that will receive control if a minor error occurs (R15=4).

**QMRAPI GET function call description**

The QMRAPI GET function call extracts individual records from output produced by an EXEC Message Advisor API call. Keyword descriptions for the GET function call appear in this section in alphabetical order.

**AREA keyword**

AREA is required for the GET function call. AREA indicates an RX type address or register that points to the output storage area.
AREALENG keyword

AREALENG is required for the GET function call. AREALENG indicates a symbol, decimal number, or register that identifies the length of the output storage area provided for the output record.

EOR keyword

EOR is required for the GET function call. EOR indicates the address of an instruction that will receive control if no more records are available (R15=4).

ERROR keyword

ERROR indicates the address of an instruction that will receive control if no more records are available or an error occurs (R15 greater than 0). If you specify both the EOR and ERROR keywords, the ERROR address only receives control when R15 is greater than or equal to 8.

MF keyword

MF indicates a macro instruction. To generate reentrant code, specify the macro instruction twice: once with $\text{MF}=L$ to generate the list form, and once with $\text{MF}=(E,\text{address})$ to generate the execute form. The address variable is the RX type address of the list form.

Following are the possible return values:

- R15=0: successful return
- R15=4: no more records available
- R15=8: Message Advisor API was not initialized
- R15=12: area length value too small

QMRAPI TERM function call description

The QMRAPI TERM function call terminates Message Advisor subtasks and cleans up common storage areas. Keyword descriptions for the TERM function call appear in this section in alphabetical order.
**ERROR keyword**

ERROR indicates the address of an instruction that will receive control if an error occurs (R15 greater than 0). If you specify both the ERROR and WARNING keywords, the ERROR address only receives control when R15 is greater than or equal to 8.

---

**Note**

TERM always generates reentrant code; MF is not allowed.

Following are the possible return values:

- R15=0: successful return
- R15=4: storage freemain failure
- R15=8: Message Advisor API was not initialized

**WARNING keyword**

WARNING indicates the address of an instruction that will receive control if a minor error occurs (R15=4).

---

**QMRAPI command definition descriptions**

Each of the QMRAPI command definitions has associated keywords that let you define how the command is executed. The command definitions and their associated keywords are described in this section.

**DISPLAY command definition description**

The QMRAPI DISPLAY command definition generates a display command for use by an EXEC call. The macro must appear in csect, and a $QMRAPI END instruction must follow the macro. The DISPLAY command will not be generated until the $QMRAPI END instruction is encountered. Keyword descriptions for the DISPLAY command definition appear below in alphabetical order.
DEVICE_TYPE keyword

DEVICE_TYPE indicates the device type to be selected. The following device types are valid: 3286, FIN, 3614, 3277, LUT6, 3790, SLU4, SLU1, SLU2, NTO, AVM, and SLUP.

IMSID keyword

IMSID is required for the DISPLAY command definition. IMSID indicates the IMS system to which this DISPLAY command definition applies.

NODE_NAME keyword

NODE_NAME indicates the name of the node to display.

Masking is allowed through wild card characters (?) and (*). A question mark represents any single character, and an asterisk represents multiple trailing characters. A question mark can appear in any position, but an asterisk must appear as the last character.

RETURNEMPTY keyword

RETURNEMPTY indicates whether or not nodes without queued messages are returned. The default value is NO.

INCLUDE/EXCLUDE command definition description

The QMRAPI INCLUDE/EXCLUDE command definition generates an INCLUDE or EXCLUDE subcommand for a DISPLAY command. You can use multiple INCLUDE and EXCLUDE subcommands after the $QMRAPI DISPLAY instruction and before the $QMRAPI END instruction.

INCLUDE and EXCLUDE subcommands refine the selection criteria for node names. You can use the subcommands in place of the NODE_NAME keyword on the DISPLAY command, or you can use them in addition to the keyword. You can use as many INCLUDE and EXCLUDE subcommands as you desire.

Message Advisor always processes INCLUDE subcommands first, regardless of the subcommand order. Node names that match the INCLUDE criteria become part of an included subset. A node name must match all criteria on a specific INCLUDE subcommand to become part of the included subset; that is, the node name must match the specified range and the transaction ID (if specified).
Message Advisor processes the INCLUDE subcommands independently. If the node name matches any one of the INCLUDE subcommands, it becomes part of the included subset. Also, any node names that match the NODE_NAME keyword on the DISPLAY command become part of the included subset. If there are no INCLUDE subcommands and no NODE_NAME keyword, then all node names become part of the included subset.

After Message Advisor determines the included subset, it processes the EXCLUDE subcommands. Any node names in the included subset which match the exclude criteria on any EXCLUDE subcommand are eliminated from the subset. Again, Message Advisor processes the EXCLUDE subcommands independently. A node name must match all criteria on a specific EXCLUDE subcommand to be eliminated.

Message Advisor displays any node names remaining in the included subset.

Keyword descriptions for the INCLUDE/EXCLUDE command definition appear in this section in alphabetical order.

**RANGE_FROM keyword**

RANGE_FROM indicates the first node name within a range of node names to be included or excluded. If you do not specify the RANGE_TO keyword, Message Advisor will include or exclude this node name and all alphabetically larger node names.

Masking is allowed through wild card characters (?) and (*). A question mark represents any single character, and an asterisk represents multiple trailing characters. A question mark can appear in any position, but an asterisk must appear as the last character.

**RANGE_TO keyword**

RANGE_TO indicates the last node name within a range of node names to be included or excluded.

Masking is allowed through wild card characters (?) and (*). A question mark represents any single character, and an asterisk represents multiple trailing characters. A question mark can appear in any position, but an asterisk must appear as the last character.

**TRANID keyword**

TRANID indicates that all node names with this destination name preset in CTBPRSTN are to be included or excluded.
Masking is allowed through wild card characters (?) and (*). A question mark
represents any single character, and an asterisk represents multiple trailing
characters. A question mark can appear in any position, but an asterisk must appear
as the last character.

**END command definition description**

The QMRAPI END command definition ends a Message Advisor command. The
$QMRAPI END instruction must follow each $QMRAPI DISPLAY instruction. There
are no keywords associated with the END command definition.

**IMSCMD command definition description**

The QMRAPI IMSCMD command definition submits an IMS command to the IMS
command processor. The QMRAPI GET function call returns IMS command outputs.

Keyword descriptions for the IMSCMD command definition appear in this section in
alphabetical order.

**COMMAND keyword**

COMMAND is required for the IMSCMD command definition. COMMAND
indicates the IMS command to be executed.

**IMSID keyword**

IMSID is required for the IMSCMD command definition. IMSID indicates the IMS
system to which this IMSCMD command definition applies.
**Glossary**

This glossary contains terms relating to the following BMC Software products:

**A**

**action menu**

The submenu that is accessible from the Action menu on the menu bar. Action menu commands differ based on the node that you select in the console tree. Action menu items can also be accessed by right-clicking on tree nodes.

**active entry**

A main entry that has a program start time but no program termination time in the PDX directory. This condition indicates that the job has failed or is still running.

**active instances node**

Node used to issue commands—such as display settings, set temporary journal and trace options, dynamically reload options and exits, reset security and statistics, and switch journals. Only Energizer and IMS Connect address spaces have commands.

**administrator authority**

The authorization an ETA user must have in order to create, modify, and delete user access profiles.

**affinities**

Messages that have unique processing characteristics. Affinities are used by the Affinity Manager to select the datastores that can process the message.

**Affinity Manager**
One of the DataStore Router managers. Determines whether incoming transaction's attributes match any predefined Energizer affinity definitions. If there is a match, the Affinity Manager compiles a list of datastores that can process the message. If there is no match, the Affinity Manager compiles a list of all active datastores that have been defined to Energizer and IMS Connect.

ALOT

See autologoff interval on page 416.

ASOT

See autosignoff interval on page 416.

authorization profile

A method to limit access to, and use of, specific BMC product functions and components. See also user authorization profile on page 425.

authorized CPU ID

The identification number associated with a specific CPU. CPU ID passwords enable you to use a specific BMC Software product on that CPU.

autologoff interval

A user-specified interval that determines when inactive terminals and printers will be automatically logged off IMS.

autosignoff interval

A user-specified interval that determines when inactive terminals and printers will be automatically signed off IMS.

Autosignon

An MAQ feature that allows terminals and printers to bypass the ETO requirement that all devices sign on to IMS.
BMCLINK

An inter-region control facility provided with DELTA IMS products that allows the DELTA IMS user to communicate with an IMS control region.

BMCRESLB

A DD name for the IMS RESLIB data set used in the IMS control region if IMS RESLIB is a LINKLIST data set.

BMCXLINK

An inter-region control facility provided with the DELTA PLUS and ETA products that allows DELTA PLUS and ETA users to communicate with an IMS control region.

command

An order for an action to take place or a statement used to request a function. A command consists of the command name and its parameters.

command set

A Message Advisor command consisting of a primary command, subcommand(s), keywords, parameters, and an END command. Some subcommands can be repeated multiple times within a command set.

configuration

This node is used to customize Energizer by adding eLinks, eGroups, IMS Connects, datastores, DataStore Router definitions, and Exit Services definitions (exits).

console

A Windows application that is used to manage BMC Software products through one interface.
CPU ID authorization

The process of authorizing a BMC Software product so that it will run on a CPU for which you have a license and a valid CPU ID password.

CPU ID password

A unique string of characters that lets you run a specific BMC Software product on a specific CPU.

customer exit

Message exits that were written outside of Energizer (in assembler language) and are maintained by you. The exits cannot use the full capabilities of Energizer and must be defined to Energizer to use DataStore Routing.

D

DataStore Router

Directs a transaction to the datastore that is best equipped to process a transaction.

DEADQ

A value that selects or rejects messages that are on the IMS dead letter queue (with ETO). DEADQ is accepted by all Message Advisor commands that use the DESTYPE= keyword.

descriptor

A skeleton from which an IMS control block is dynamically built. There are four types of descriptors: LOGON, USER, MFS device, and MSC.

descriptor list

Information created and stored through a product that allows for the dynamic addition or deletion of descriptors on an IMS system.

DYNAMIC
A value that selects or rejects messages destined to ETO or VTF LTERMs. DYNAMIC is accepted by all Message Advisor commands that use the DESTYPE= keyword.

dynamic LTERM

Any LTERM in your IMS network not defined to IMS in the IMSGEN, but rather by DELTA PLUS VIRTUAL TERMINAL or ETA. If an LTERM assigned to a dynamic terminal is not found among IMSGEN-defined LTERMs or existing dynamic LTERMs, IMS creates a new dynamic LTERM with the name it requires.

dynamic printer

Any printer not defined in an IMSGEN, but rather defined to IMS by DELTA PLUS VIRTUAL TERMINAL or ETA. Dynamic terminal node names and LTERM names can be contained in a TSS table. The table allows possible dynamic printer LTERM names to be validated and, when needed, translated into the dynamic printer node names.

dynamic terminal

Any terminal not defined in an IMSGEN, but rather created at logon time by DELTA PLUS VIRTUAL TERMINAL or ETA. Dynamic terminal control blocks reside in the IMS control region only while that dynamic terminal is logged on to IMS. Dynamic terminal control blocks are deleted when dynamic terminals are logged off, thereby freeing virtual storage for another dynamic terminal user.

E

eGroup

A logical grouping of IMS Connects that use the same routing method. Because the eGroup name will be used as an XCF group name, the eGroup name must be unique. It must not match any existing XCF group name.

eLink

Energizer address space that provides the communications link between the UIM server and the IMS Connects. The eLink is used to change the environment and obtain system-related information.

EXER Subroutine
IMS Connect term. When IMS Connect detects an error in the output buffer after execution of the previous READ subroutine completes, control is passed to the EXER subroutine in the same user exit where the READ subroutine is executed. For more information, see the IMS Connect Guide and Reference manual.

Exit Services

Energizer for IMS Connect component that expands the basic functionality of IMS Connect—such as dynamic reload capabilities, virtual exits, and security exits.

G

global options

Options that govern how DELTA PLUS operates on all IMS systems that use DELTA PLUS. DELTA PLUS consists of various elements associated through VTAM and/or normal IMS inter-region communication. The information you specify in the DELTA PLUS global options keep these elements operating according to standards that you establish. These global options apply to all IMS systems that use DELTA PLUS.

Group options

Options that enable a user-defined group of IMS systems to be treated as one IMS system. A group of IMS systems would typically be an IMS Datasharing Group, IMS Shared Queues Group, or a group of logically related or duplicated IMS systems.

I

IMSID options

Options that specify customization information for each IMS system that uses DELTA PLUS.

INIT Subroutine

IMS Connect term. After a message exit has been successfully loaded, the INIT subroutine for that message exit is called. For more information, see the IBM IMS Connect Guide and Reference manual.
K

classification
A name or symbol that identifies an option associated with a Message Advisor command or subcommand. Message Advisor keywords are followed by an equal (=) sign and are separated by a comma (,).

L

Load Balancer
Routing method that is used by the DataStore Router. Determines where to route the message based upon one of the routing methods: Workload Manager (WLM) Sysplex Routing Services or Statistical.

M

message exits
Exits that can be created in assembler language and are defined to Energizer (Customer Exits), or exits that are created and maintained in Energizer (Virtual Exits). See also user message exit on page 425.

Message Advisor Server
A functional unit that provides a shared interface to IMS control regions. The Message Advisor Server provides a method by which the content of IMS message queues can be managed. It can run as a started task or in batch mode.

messages pane
The bottom pane in the console. It displays system status and warnings. Message information includes: severity, status, message source, message ID, and message.

middleware tier
Tier that resides on the mainframe and contains the UIM server. The UIM server handles the communication between the client and the product.
navigation pane

The left pane of the console. It displays the hierarchical organization of your enterprise environment. The hierarchical structure lets you drill down through the components. The components in the navigation pane have right-click menus.

node

(1) An IMS resource that represents a physical VTAM terminal. The node is represented by a VTCB control block. (2) For Energizer, a location on the tree that represents a component. Each node on the tree has a corresponding icon.

override options

Options set at the entry or PSB level.

parameter

A keyword variable used with a command or subcommand to affect its result.

pop-up menu

Menu items that can be accessed by right-clicking on tree nodes. Pop-up menu commands differ based on the node that you select in the console tree.

processing options

Data that is stored in the options library. By using the console Save button translates field values into the proper format to store in the options library.
An agreement between your organization and BMC Software stating the extent of your legal right to use a specific product and the system and CPU on which you will run the product.

**product tier**

The product tier resides on the mainframe and contains BMC Software products.

**request**

A unit of work that contains one or more command sets. Requests are stored as members in the Message Advisor request library.

**session**

Series of commands that come from the same client and belong to the same logical sequence.

**spare element pool**

A group of unused terminal, LTERM, and subpool control blocks used by DELTA IMS DC and DELTA IMS DB/DC to add terminals, LTERMs, and subpools between IMSGENs.

**task pane**

The right pane in the console. It is your work area. When you select an action from a menu in the console, the window that corresponds to the selected action displays in the task pane. Depending on the selected action, tabs display reports or data entry fields.

**TERM Subroutine**

IMS Connect term. When IMS Connect is shutting down, control is passed, in turn, to the TERM subroutine in each message exit that is active. For more information, see the *IMS Connect Guide and Reference* manual.
Translate Subsystem Services

A DELTA PLUS VIRTUAL TERMINAL and ETA generalized table lookup feature that is used to create and use tables of data to specify IMS customization options.

tree

A representation of the hierarchical organization of your enterprise environment. The tree is used to drill down through the hierarchy to access nodes.

tree node

A location on the tree that represents a component. Each node on the tree has a corresponding icon.

U

UIM server

User Interface Middleware server. The UIM server is common to a family of BMC Software IMS products. The UIM server is a TCP/IP application that facilitates communication between client workstations and the mainframe.

UIMx

Common component for all products that use the UIM server. The term is used during the installation process.

undock

Action that reattaches a window to the task pane (contents of the window are the same in dock and undock modes).

unknown destination

A destination to which output has been queued, but which does not exist on the IMS system because the LTERM was not IMSGEN-defined or has not been dynamically created. The DELTA PLUS VIRTUAL TERMINAL and ETA products allow you to control the creation of unknown destinations.
Unsolicited Output feature

A DELTA PLUS VIRTUAL TERMINAL and ETA feature that is used to specify whether unknown destinations should be created on an IMS system. This feature can also be used to specify options for unknown destinations (if you allow them to be created) and for dynamic printer LTERMs and the associated SPQB and VCNT control block structures.

UPDS

See user profile data set on page 425.

USER

In documentation, this term denotes the IMS user element.

user access profile

Information used by the internal security feature of DELTA PLUS and ETA. User access profiles specify the product features that a user (specified by user ID) can utilize on a specified IMS system.

user authorization profile

A method of limiting access to, and use of, specific BMC Software product functions and components.

user message exit

IMS Connect term. A message exit that is created and maintained by using assembler language. These exits process the requests that IMS Connect receives from the client or datastore.

user profile data set

For the ETA and DELTA PLUS products, a partitioned data set used to store the ETA and DELTA PLUS user access profiles. The user access profiles control access to ETA and DELTA PLUS functions.

user/SPQB

In DELTA IMS documentation, denotes the IMS user element.
virtual exit

Message exit that was created in Energizer, but without using assembler language. The exits are maintained and updated by using the console.

virtual LTERM

In an IMS network, any LTERM that is not defined to IMS in the IMSGEN but is instead created when needed. If an LTERM assigned to a virtual terminal does not exist among IMSGEN-defined LTERMs or virtual LTERMs, a new virtual LTERM with the required name can be created.

virtual printer

Any SLU 1- or 328x-type printer that is not defined in an IMSGEN but is instead created when needed and later deleted. A possible virtual printer LTERM name can be validated and, when needed, translated into the virtual printer node name. Virtual printer LTERMs are created when needed, and additional elements are created shortly after output is ready to be sent to the printer.

virtual terminal

Any 3270- or SLU 2-type terminal that is not defined in an IMSGEN but is instead created at logon time and deleted after the user logs off IMS. Installations can specify that the IMS terminal control block for a virtual terminal resides in the IMS control region.

virtual terminal control block

A control block created at logon exit time that consists of a prefix and copies of the appropriate model control blocks. A suffix of one word is added to each virtual communications line block. This suffix contains a code that identifies the control block as virtual.

VSPCNT

A value that selects or rejects messages destined to ISC destinations. VSPCNT is accepted by all Message Advisor commands that use the DESTYPE= keyword.
WorkLoad Governor

An Energizer feature that protects the availability of datastores by limiting the number of transactions passing through IMS Connect.
Index

#MSGS keyword
  on QPF_OPTIONS ENFORCE statement 120
  on QPF_OPTIONS PROCESS statement 131
#RECORDS keyword
  on QPF_OPTIONS ENFORCE statement 121
  on QPF_OPTIONS PROCESS statement 131
%USED keyword
  on QPF_OPTIONS ENFORCE statement 122
  on QPF_OPTIONS PROCESS statement 132

A

ABEND_CODE keyword 108
ACTION keyword
  on QPF_ACTION command 54
  on QPF_OPTIONS ENFORCE statement 109
  on QPF_OPTIONS OVERFLOW_PHASE subcommand 132
  on QPF_OPTIONS PROCESS statement 123
action menu
  definition 415
active entry
  definition 415
active instances node
  definition 415
ACTIVE keyword
  on QPF_LIST command 67
  on QPF_OPTIONS command 87
administrator authority
  definition 415
affinities
  definition 415
AFFINITY keyword
  on SELECT and REJECT subcommands 321
  on SELECT and REJECT subcommands for shared queues 391
Affinity Manager
  definition 415

ALOT
  definition 416
APPEND keyword 305
AREA keyword 409
AREALENG keyword 410
ASOT
  definition 416
authorization profile
  definition 416
authorized CPU ID
  definition 416
AUTO_CMD keyword
  on QPF_OPTIONS MONITOR_PHASE subcommand 96
  on QPF_OPTIONS OVERFLOW_PHASE subcommand 133
  on QPF_OPTIONS PROTECT_PHASE subcommand 102
AUTO_CMD_COLDCOM keyword 153
AUTO_CMD_COLDSYS keyword 154
AUTO_REQ keyword 155
AUTO_REQ_CMD keyword 155
AUTO_REQ_COLDCOM keyword 156
AUTO_REQ_COLDSYS keyword 157
AUTODQ_MASTER keyword
  on QPF_OPTIONS MONITOR_PHASE subcommand 96
AUTODQ_SECONDARY keyword
  on QPF_OPTIONS MONITOR_PHASE subcommand 97
autologoff interval
  definition 416
autosignoff interval
  definition 416
Autosignon
  definition 416

B

batch method of building command sets 35
BLKSIZE keyword
   on OUTPUT, EXTRACT, and SCRAP subcommands 305
   on QPF_OPTIONS UNLOAD_DSN subcommand 139
BMCLINK
   definition 417
BMCRESLB
   definition 417
BMCXLINK
   definition 417
BRANCH keyword 402

C

CHANGE subcommand
   description 258
   overview 254
   syntax diagram 258
CHEckpoint_dsn keyword 157
CHKPT keyword 238
CMD keyword
   on IMSCMD command 222
   on PLEXCMD command 224
CNT keyword 323
command
   definition 417
command definition descriptions
   QMRAPI DISPLAY 411
   QMRAPI END 414
   QMRAPI IMSCMD 414
   QMRAPI INCLUDE/EXCLUDE 412
command descriptions
   COND pseudo command 401
   CUSTOMIZE 152
   DEQUEUE 182
   DEQUEUE for shared queues 371, 372
   DISPLAY for shared queues 374
   DISPLAY TYPE=ANALYZER 219
   DISPLAY TYPE=ANALYZER for shared queues 384
   DISPLAY TYPE=DEST_QUEUES 204
   DISPLAY TYPE=DEST_QUEUES for shared queues 377
   DISPLAY TYPE=DESTINATION 200
   DISPLAY TYPE=DESTINATION for shared queues 376
   DISPLAY TYPE=MESSAGE 210
   DISPLAY TYPE=MESSAGE for shared queues 379
   DISPLAY TYPE=RECORD 214
   DISPLAY TYPE=RECORD for shared queues 381
   DISPLAY TYPE=STATISTICS 218
   DISPLAY TYPE=STATISTICS for shared queues 383
   PLEXCMD 224
   QPF_OPTIONS 86
   REQUEUE 237
   UNLOAD 286
   UNLOAD for shared queues 386, 387
COMMAND keyword 414
command set
   definition 417
command sets
   building through batch method 35
   building through ISPF interface 34
   command syntax considerations 37
   executing
      interactive method 34
      WTOR method 36
   overview 29
commands
COND pseudo command
  overview 399
COND pseudo command,
sample command set 401
CUSTOMIZE
  overview 147
  sample command sets 151
  SERVER_OPTIONS subcommand
description 168
DEQUEUE
  overview 177
  sample command sets 180
DEQUEUE for shared queues
description 372
  sample command sets 371
DISPLAY
  overview 193
  sample command sets 198
DISPLAY for shared queues
  sample command sets 374
DISPLAY TYPE=ANALYZER for shared queues
description 384
DISPLAY TYPE=DEST_QUES for shared queues
description 377
DISPLAY TYPE=DESTINATION for shared queues
description 376
DISPLAY TYPE=MESSAGE for shared queues
description 379
DISPLAY TYPE=RECORD for shared queues
description 381
DISPLAY TYPE=STATISTICS for shared queues
description 383
IMSCMD
  overview 221
  sample command sets 222
PLEXCMD
description 224
  overview 221
  sample command set 224
  syntax diagram 223
QPF_ACTION
  initiate an action 48
  overview 47
  sample command sets 53
  STOP action commands by device type 49
QPF_LIST
display a problem list 64
  overview 63
  sample command sets 65
QPF_OPTIONS
  description 86
  overview 71
  PROCESS keywords usage example 83
  PROCESS statement usage
  considerations 122
  PROTECT keywords usage example 82
  sample command sets 79
  syntax diagram 73
  UNLOAD_DSN keywords usage
  example 84
REQUEUE
description 237
  overview 227
  requeue from one system to another
  228
  sample command sets 235
  SPILL subcommand description 275
  SPILL subcommand overview 275
  support for uncataloged SLDS 230
  syntax diagram 233
REQUEUE for shared queues
  sample command sets 384
  usage considerations 385
UNLOAD
  description 286
  overview 281
  sample command sets 284
  syntax diagram 283
UNLOAD for shared queues
description 387
  sample command sets 386
COMMITMODE keyword 259, 323
  common keywords
  for SELECT, REJECT, CHANGE and GROUP
  subcommands 318
COMPRESS keyword 239
COND pseudo command
description 401
  overview 399
  sample command set 401
  syntax diagram 400
conditional steps
  on DEQUEUE command 178
  on DISPLAY command 195
  on REQUEUE command 231
  on UNLOAD command 282
configuration  definition 417
console  definition 417
conventions 17
CONVERSATIONS keyword
  on DEQUEUE command 183
  on REQUEUE command 239
  on UNLOAD command 286
CONVID keyword 323
CPU ID authorization  definition 418
CPU ID password  definition 418
customer exit  definition 418
CUSTOMIZE command
  description 152
    IMS_OPTIONS subcommand  description 153
    LIST_OPTIONS subcommand  description 167
overview 147
sample command sets 151
  activate automatic requeue 151
  list Message Advisor Servers and IMS systems for which options have been defined 151
  list options for a specific IMS ID 151
  list options for a specific Message Advisor Server 151
SERVER_OPTIONS subcommand  description 168
syntax diagram 149
uppercase-only feature 148
virtual LTERM creation feature 148
CYLS_PRIM keyword
  on OUTPUT, EXTRACT, and SCRAP subcommands 306
  on QPF_OPTIONS UNLOAD_DSN subcommand 139
  on REQUEUE SPILL subcommand 276
CYLS_SEC keyword
  on OUTPUT, EXTRACT, and SCRAP subcommands 306
  on QPF_OPTIONS UNLOAD_DSN subcommand 140
  on REQUEUE SPILL subcommand 276
D
DATA keyword 326
DATACLAS keyword
  on OUTPUT, EXTRACT, and SCRAP subcommands 307
  on QPF_OPTIONS UNLOAD_DSN subcommand 140
  on REQUEUE SPILL subcommand 276
DATASIZE keyword 326
DataStore Router  definition 418
DBD keyword 327
DDNAME keyword
  on OUTPUT, EXTRACT, and SCRAP subcommands 307
  on REQUEUE command 272
DEADQ  definition 418
DEADQ keyword 328
DEFAULT_SERVER keyword 158
DELETE keyword
  on CUSTOMIZE IMS_OPTIONS subcommand 158
  on CUSTOMIZE SERVER_OPTIONS subcommand 168
DEQUEUE command
  and conditional steps 178
  description 182
  for shared queues
    description 372
    sample command sets 371
overview 177
sample command sets 180
  dequeue from multiple destinations 180
  dequeue from one destination 180
DEQUEUE command for shared queues
  description 371
descriptor  definition 418
descriptor list  definition 418
DESTINATION keyword
  on QPF_OPTIONS ENFORCE statement 111
  on QPF_OPTIONS PROCESS statement 124
  on SELECT and REJECT subcommands 328
DESTYPE keyword
for shared queues 329
parameters and matching destinations 389
on QPF_OPTIONS ENFORCE statement 111
on QPF_OPTIONS PROCESS statement 124
DETAIL keyword 167
DEVICE_TYPE keyword 412
DISCARD keyword 240
DISP keyword
  on OUTPUT, EXTRACT, and SCRAP subcommands 307
  on QPF_OPTIONS UNLOAD_DSN subcommand 140
DISPLAY command
  and conditional steps 195
  for shared queues
description 374
  sample command sets 374
overview 193
sample command sets
  display a formatted message record 198
display a message record 198
display detailed destination report 198
display message queue statistics 198
display summary destination queues report 198
overview 198
syntax diagram 195
DISPLAY command for shared queues
sample command sets
  display a message record 374
display detailed destination report 374
display message queue statistics 374
display summary destination queues report 374
DISPLAY TYPE=ANALYZER command
description 219
for shared queues
description 384
DISPLAY TYPE=DEST_QUEUES command
description 204
for shared queues
description 377
DISPLAY TYPE=DESTINATION command
description 200
for shared queues
description 376
DISPLAY TYPE=MESSAGE command
description 210
for shared queues
description 379
DISPLAY TYPE=RECORD command
description 214
for shared queues
description 381
DISPLAY TYPE=STATISTICS command
description 218
for shared queues
description 383
documentation information 16
DRRN keyword
  on DISPLAY TYPE=RECORD command 214
  on SELECT and REJECT subcommands 332
DSNAME keyword
  on OUTPUT, EXTRACT, and SCRAP subcommands 308
symbolic keywords allowed 308
  on QPF_OPTIONS UNLOAD_DSN subcommand 141
  on REQUEUE command 273
DUMP keyword 87
DYNAMIC
definition 418
dYNAMIC keyword 333
dynamic LTERM
definition 419
dynamic printer
definition 419
dynamic terminal
definition 419
E
eGroup
definition 419
EIB keyword 408
electronic documentation 16
eLink
definition 419
ENFORCE statements
  keywords
    usage examples 84
    usage considerations 108
ENTRYID keyword
on DISPLAY TYPE=RECORD command 215
on DISPLAY TYPE=RECORD command for shared queues 382
on SELECT and REJECT subcommands 334
on SELECT and REJECT subcommands for shared queues 393
EOR keyword 410
ERROR keyword
  on QMRAPI EXEC function call 408
  on QMRAPI GET function call 411
  on QMRAPI INIT function call 407
  on REQUEUE command 241
EXER Subroutine
definition 419
EXIT keyword 242
Exit Services
definition 420
EXTRACT subcommand
description 304
overview 303
EXTRACT_DSN keyword 158

G
global options
definition 420
Group options
definition 420
GROUP subcommand
description 271
overview 270
syntax diagram 271
GRS_QNAME keyword 169

H
HELD keyword 335
Help
  online 16
HIPER_ASM% keyword 88
HIPER_PAGES keyword 89
HIPER_USAGE keyword 90
HIPERASSIST keyword 89

I
IFCOND keyword 402
IMS versions 18
IMS_OPTIONS subcommand
description 153
IMSCMD command 222
  execute an IMS command 222
  overview 221
  sample command sets 222
    execute an IMS command 222
    syntax diagram 222
IMSid keyword
  on QPF_ACTION command 56
IMSID keyword
on CUSTOMIZE IMS_OPTIONS subcommand 159
on CUSTOMIZE LIST_OPTIONS subcommand 167
on DEQUEUE command 187
on DISPLAY TYPE=ANALYZER command 219
on DISPLAY TYPE=DEST_QUEUES command 205
on DISPLAY TYPE=DESTINATION command 200
on DISPLAY TYPE=MESSAGE command 211
on DISPLAY TYPE=RECORD command 215
on DISPLAY TYPE=STATISTICS command 218
on IMSCMD command 223
on QMRAPI DISPLAY command definition 412
on QMRAPI IMSCMD command definition 414
on QPF_LIST command 67
on QPF_OPTIONS command 90
on REQUEUE command 243
on UNLOAD command 289

IMSID options
definition 420
INCORE keyword
and REQUEUE performance 231
description 243
INIT Subroutine
definition 420
INPUT command
  INPUT subcommand
    overview 272
  INPUT subcommand
description 272
  overview 272
  syntax diagram 272
INTERNAL_PREFIX keyword 244
INTERVAL subcommand
description 299
  overview 293
  syntax diagram 298
  time stamp format for START and STOP keywords 295
  time zone label for START and STOP keywords 296
  time zone offset for START and STOP keywords 296

ISPF
  and building command sets 34

J

JOBNAME keyword 112

K

KEY keyword
  on SELECT and REJECT subcommands 334
  on SELECT and REJECT subcommands for shared queues 393
  keyword
definition 421
  keywords
OTMA_LIMIT 115

#MSGS
  on QPF_OPTIONS ENFORCE statement 120
  on QPF_OPTIONS PROCESS statement 131

#RECORDS
  on QPF_OPTIONS ENFORCE statement 121
  on QPF_OPTIONS PROCESS statement 131

%USED
  on QPF_OPTIONS ENFORCE statement 122
  on QPF_OPTIONS PROCESS statement 131

ABEND_CODE 108

ACTION
  on QPF_ACTION command 54
  on QPF_OPTIONS ENFORCE statement 109
  on QPF_OPTIONS OVERFLOW_PHASE subcommand 132
  on QPF_OPTIONS PROCESS statement 123

ACTIVE
  on QPF_LIST command 67
  on QPF_OPTIONS command 87

AFFINITY
  on SELECT and REJECT subcommands 321
  on SELECT and REJECT subcommands for shared queues 391

APPEND 305
 AREA 409
 AREALENG 410
 AUTO_CMD
  on QPF_OPTIONS MONITOR_PHASE subcommand 96
  on QPF_OPTIONS OVERFLOW_PHASE subcommand 133
  on QPF_OPTIONS PROTECT_PHASE subcommand 102

AUTO_CMD_COLDCOM 153
 AUTO_CMD_COLDSYS 154
 AUTO_REQ 155
 AUTO_REQ_CMD 155
 AUTO_REQ_COLDCOM 156

AUTO_REQ_COLDSYS 157

AUTODQ_MASTER
  on QPF_OPTIONS MONITOR_PHASE subcommand 96

AUTODQ_SECONDARY
  on QPF_OPTIONS MONITOR_PHASE subcommand 97

BLKSIZE
  on OUTPUT, EXTRACT, andSCRAP subcommands 305
  on QPF_OPTIONS UNLOAD_DSN subcommand 139

BRANCH 402
CHEckpoint_dsn 157
CHKPT 238
CMD
  on IMSCMD command 222
  on PLEXCMD command 224

CNT 323
COMMAND 414

COMMITMODE 259, 323
  common for SELECT, REJECT, CHANGE, and GROUP subcommands 318
  common to SELECT, REJECT, CHANGE, and GROUP subcommands 254, 318

COMPRESS 239

CONVERSATIONS
  on DEQUEUE command 183
  on REQUEUE command 239
  on UNLOAD command 286

CONVID 323
CYLS_PRIM
  on OUTPUT, EXTRACT, andSCRAP subcommands 306
  on QPF_OPTIONS UNLOAD_DSN subcommand 139
  on REQUEUE SPILL subcommand 276

CYLS_SEC
  on OUTPUT, EXTRACT, andSCRAP subcommands 306
  on QPF_OPTIONS UNLOAD_DSN subcommand 140
  on REQUEUE SPILL subcommand 276

DATA 326
DATAACLAS
  on OUTPUT, EXTRACT, andSCRAP subcommands 307
  on QPF_OPTIONS UNLOAD_DSN subcommand 140
  on REQUEUE SPILL subcommand 276
DATASIZE 326
DBD 327
DDNAME
  on OUTPUT, EXTRACT, and SCRAP subcommands 307
  on REQUEUE command 272
DEADQ 328
DEFAULT_SERVER 158
DELETE
  on CUSTOMIZE IMS_OPTIONS subcommand 158
  on CUSTOMIZE SERVER_OPTIONS subcommand 168
DESTINATION
  on QPF_OPTIONS ENFORCE statement 111
  on QPF_OPTIONS PROCESS statement 124
  on SELECT and REJECT subcommands 328
DESTYPE
  on QPF_OPTIONS ENFORCE statement 111
  on QPF_OPTIONS PROCESS statement 124
  DESTYPE for shared queues 329
  parameters and matching destinations 389
DETAIL 167
DEVICE_TYPE 412
DISCARD 240
DISP
  on OUTPUT, EXTRACT, and SCRAP subcommands 307
  on QPF_OPTIONS UNLOAD_DSN subcommand 140
DRRN
  on DISPLAY TYPE=RECORD command 214
  on SELECT and REJECT subcommands 332
DSNAME
  on OUTPUT, EXTRACT, and SCRAP subcommands 308
  on QPF_OPTIONS UNLOAD_DSN subcommand 141
  on REQUEUE command 273
  symbolic keywords allowed on OUTPUT, EXTRACT, and SCRAP subcommands 308
DUMP 87
DYNAMIC 333
EIB 408
ENTRYID
  on DISPLAY TYPE=RECORD command 215
  on DISPLAY TYPE=RECORD command for shared queues 382
  on SELECT and REJECT subcommands 334
  on SELECT and REJECT subcommands for shared queues 393
EOR 410
ERROR
  on QMRAPI EXEC function call 408
  on QMRAPI GET function call 411
  on QMRAPI INIT function call 407
  on REQUEUE command 241
EXIT 242
EXTRACT_DSN 158
FORCE
  on DEQUEUE command 185
  on QPF_ACTION command 56
  on QPF_OPTIONS PROCESS statement 125
  on UNLOAD command 288
FORMAT
  on DISPLAY TYPE=MESSAGE command 210
  on DISPLAY TYPE=RECORD command 216
  on QPF_OPTIONS command 88
GRS_QNAME 169
HELD 335
HIPER_ASM% 88
HIPER_PAGES 89
HIPER_USAGE 90
HIPERASSIST 89
IFCOND 402
IMSID
  on QPF_ACTION command 56
IMSID
<table>
<thead>
<tr>
<th>Command/Keyword</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>on CUSTOMIZE IMS_OPTIONS subcommand</td>
<td>159</td>
</tr>
<tr>
<td>on CUSTOMIZE LIST_OPTIONS subcommand</td>
<td>167</td>
</tr>
<tr>
<td>on DEQUEUE command</td>
<td>187</td>
</tr>
<tr>
<td>on DISPLAY TYPE=ANALYZER command</td>
<td>219</td>
</tr>
<tr>
<td>on DISPLAY TYPE=DEST_QUEUES command</td>
<td>205</td>
</tr>
<tr>
<td>on DISPLAY TYPE=DESTINATION command</td>
<td>200</td>
</tr>
<tr>
<td>on DISPLAY TYPE=MESSAGE command</td>
<td>211</td>
</tr>
<tr>
<td>on DISPLAY TYPE=RECORD command</td>
<td>215</td>
</tr>
<tr>
<td>on DISPLAY TYPE=STATISTICS command</td>
<td>218</td>
</tr>
<tr>
<td>on IMSCMD command</td>
<td>223</td>
</tr>
<tr>
<td>on QMRAPI DISPLAY command definition</td>
<td>412</td>
</tr>
<tr>
<td>on QMRAPI IMSCMD command definition</td>
<td>414</td>
</tr>
<tr>
<td>on QPF_LIST command</td>
<td>67</td>
</tr>
<tr>
<td>on QPF_OPTIONS command</td>
<td>90</td>
</tr>
<tr>
<td>on REQUEUE command</td>
<td>243</td>
</tr>
<tr>
<td>on UNLOAD command</td>
<td>289</td>
</tr>
<tr>
<td>INCORE</td>
<td>243</td>
</tr>
<tr>
<td>INTERNAL_PREFIX</td>
<td>244</td>
</tr>
<tr>
<td>JOBNAME</td>
<td>112</td>
</tr>
<tr>
<td>KEY</td>
<td></td>
</tr>
<tr>
<td>on SELECT and REJECT subcommands</td>
<td>334</td>
</tr>
<tr>
<td>on SELECT and REJECT subcommands for shared queues</td>
<td>393</td>
</tr>
<tr>
<td>LABEL</td>
<td></td>
</tr>
<tr>
<td>on COND pseudo command</td>
<td>403</td>
</tr>
<tr>
<td>on INTERVAL subcommand</td>
<td>299</td>
</tr>
<tr>
<td>on REQUEUE CHANGE subcommand</td>
<td>259</td>
</tr>
<tr>
<td>on REQUEUE GROUP subcommand</td>
<td>271</td>
</tr>
<tr>
<td>on SELECT and REJECT subcommands</td>
<td>335</td>
</tr>
<tr>
<td>LAST USED</td>
<td>126</td>
</tr>
<tr>
<td>LIMIT</td>
<td></td>
</tr>
<tr>
<td>on DISPLAY TYPE=DEST_QUEUES command</td>
<td>205</td>
</tr>
<tr>
<td>on DISPLAY TYPE=DESTINATION command</td>
<td>201</td>
</tr>
<tr>
<td>on DISPLAY TYPE=MESSAGE command</td>
<td>211</td>
</tr>
<tr>
<td>LIST_QUEUES</td>
<td>205</td>
</tr>
<tr>
<td>LOG</td>
<td></td>
</tr>
<tr>
<td>on DEQUEUE command</td>
<td>187</td>
</tr>
<tr>
<td>on DEQUEUE command for shared queues</td>
<td>372</td>
</tr>
<tr>
<td>LOGCODE</td>
<td></td>
</tr>
<tr>
<td>on CUSTOMIZE LIST_OPTIONS subcommand</td>
<td>160</td>
</tr>
<tr>
<td>on QPF_OPTIONS command</td>
<td>91</td>
</tr>
<tr>
<td>LTERM</td>
<td>336</td>
</tr>
<tr>
<td>LUNAME</td>
<td></td>
</tr>
<tr>
<td>on QPF_OPTIONS ENFORCE statement</td>
<td>113</td>
</tr>
<tr>
<td>on QPF_OPTIONS PROCESS statement</td>
<td>126</td>
</tr>
<tr>
<td>on SELECT and REJECT subcommands</td>
<td>336</td>
</tr>
<tr>
<td>MAX_DSNS</td>
<td>142</td>
</tr>
<tr>
<td>MAX_REGION</td>
<td>91</td>
</tr>
<tr>
<td>MAXDATA</td>
<td>244</td>
</tr>
<tr>
<td>MAXMSGS</td>
<td></td>
</tr>
<tr>
<td>on DISPLAY TYPE=MESSAGE command</td>
<td>211</td>
</tr>
<tr>
<td>on REQUEUE command</td>
<td>245</td>
</tr>
<tr>
<td>MAXRC</td>
<td>404</td>
</tr>
<tr>
<td>MEMBER</td>
<td>56</td>
</tr>
<tr>
<td>MF</td>
<td></td>
</tr>
<tr>
<td>on QMRAPI EXEC function call</td>
<td>409</td>
</tr>
<tr>
<td>on QMRAPI GET function call</td>
<td>410</td>
</tr>
<tr>
<td>on QMRAPI INIT function call</td>
<td>407</td>
</tr>
<tr>
<td>MFSNAME</td>
<td>337</td>
</tr>
<tr>
<td>MGMTCLAS</td>
<td></td>
</tr>
<tr>
<td>on OUTPUT, EXTRACT, and SCRAP subcommands</td>
<td>309</td>
</tr>
<tr>
<td>on QPF_OPTIONS UNLOAD_DSN subcommand</td>
<td>143</td>
</tr>
<tr>
<td>on REQUEUE SPILL subcommand</td>
<td>277</td>
</tr>
<tr>
<td>MODE</td>
<td></td>
</tr>
<tr>
<td>on DEQUEUE command</td>
<td>188</td>
</tr>
<tr>
<td>on QPF_OPTIONS PROTECT_PHASE subcommand</td>
<td>103</td>
</tr>
<tr>
<td>on REQUEUE command</td>
<td>245</td>
</tr>
<tr>
<td>MODEL_DSNNAME</td>
<td></td>
</tr>
</tbody>
</table>
on OUTPUT, EXTRACT, and SCRAP subcommands 310
on QPF_OPTIONS UNLOAD_DSN subcommand 143
MOVEPAGE 92
MSNAME 340
NAME
on QPF_ACTION command 57
on QPF_OPTIONS ENFORCE statement 113
NETID 58
NEWAFFINITY 260
NEWCOMMITMODE 260
NEWDEST 260
NEWER_THAN
on INTERVAL subcommand 299
on SELECT and REJECT subcommands 340
NEWER_THAN and OLDER_THAN usage considerations 297
NEWLUNAME 261
NEWMFSNAME 262
NEWNODENAME 262
NEWOIMS 262
NEWORQMANAGER 264
NEWORQNAME 263
NEWORQTMEMBER 264
NEWORQPIPE 264
NEWORIGIN 263
NEWPIMS 265
NEWQUEUE 265
NEWSAFGROUP 266
NEWSYSID 266
NEWTIMESTAMP 267
NEWTMEMBER 267
NEWTPPIPE 268
NEWTPNAME 268
NEWUSERID 269
NODE 114
NODE_NAME
on QMRAPI DISPLAY command definition 412
NODENAME on SELECT and REJECT subcommands 340
NOTIFY
on QPF_OPTIONS command 92
on QPF_OPTIONS ENFORCE statement 114
on QPF_OPTIONS MONITOR_PHASE 97
on QPF_OPTIONS OVERFLOW_PHASE subcommand 134
on QPF_OPTIONS PROCESS statement 127
on QPF_OPTIONS PROTECT_PHASE subcommand 103
NOTIFY_INTERVAL
on QPF_OPTIONS MONITOR_PHASE subcommand 98
on QPF_OPTIONS OVERFLOW_PHASE subcommand 134
on QPF_OPTIONS PROTECT_PHASE subcommand 104
NOTIFY%
on QPF_OPTIONS MONITOR_PHASE subcommand 98
on QPF_OPTIONS OVERFLOW_PHASE subcommand 134
on QPF_OPTIONS PROTECT_PHASE subcommand 104
NRE_SNAPQ 160
OFFSET 341
OIMS 342
OLDER_THAN
on INTERVAL subcommand 300
on SELECT and REJECT subcommands 342
OLDS 246
ORQMANAGER 344
ORQNAME 344
ORQTMEMBER 346
ORQTPipe 345
ORigin 343
ORIGINS 67
OTMAQ 346
OTOKEN 347
PIMS 347
PLEXID 225
PREFIX 277
PROBLEMS 68
PROCNAME 407
PROTECT 93
PSB
  on QPF_OPTIONS ENFORCE statement 115
  on SELECT and REJECT subcommands 348
PTOKEN 349
QPOSITION
description 349
  relative position 350
QTYPE
  on SELECT and REJECT subcommands 351
  on SELECT and REJECT subcommands for shared queues 394
QUEUE
  on QPF_OPTIONS PROCESS statement 128
  on SELECT and REJECT subcommands 356
RANGE_FROM 413
RANGE_TO 413
RATE
  on CUSTOMIZE IMS_OPTIONS subcommand 160
  on DEQUEUE command 188
  on DISPLAY TYPE=DEST_QUEUES command 206
  on DISPLAY TYPE=DESTINATION command 201
  on DISPLAY TYPE=MESSAGE command 212
  on REQUEUE command 246
  on UNLOAD COMMAND 290
RCNT 357
RECON_DYNALLOC 162
RECON1_DSN 161
RECON2_DSN 161
RECON3_DSN 162
REGION
  on QPF_ACTION command 59
  on SELECT and REJECT subcommands 358
REGION_ID 359
REPLACE 269
REPORT
  on DEQUEUE command 188
  on REQUEUE command 247
  on UNLOAD command 290
REQ_PROMPT
on CUSTOMIZE SERVER_OPTIONS subcommand 169
on REQUEUE command 248
RESET 93
RESET_TIME
  on QPF_OPTIONS MONITOR_PHASE subcommand 99
  on QPF_OPTIONS OVERFLOW_PHASE subcommand 136
  on QPF_OPTIONS PROTECT_PHASE subcommand 105
RESET%
  on QPF_OPTIONS MONITOR_PHASE subcommand 99
  on QPF_OPTIONS OVERFLOW_PHASE subcommand 135
  on QPF_OPTIONS PROTECT_PHASE subcommand 105
RETRY 143
RETURN_EMPTY 412
ROUTE 225
RSMB 359
SAFGROUP 360
SCOPE
on DEQUEUE command 189
on DEQUEUE command for shared queues 372
on DISPLAY TYPE=DEST_QUEUES command 206
on DISPLAY TYPE=DEST_QUEUES command for shared queues 378
on DISPLAY TYPE=DESTINATION command 201
on DISPLAY TYPE=DESTINATION command for shared queues 376
on DISPLAY TYPE=MESSAGE command 212
on DISPLAY TYPE=MESSAGE command for shared queues 380
on DISPLAY TYPE=RECORD command 216
on DISPLAY TYPE=RECORD command for shared queues 382
on UNLOAD command 291
on UNLOAD command for shared queues 388
START
on INTERVAL subcommand 300
on SELECT and REJECT subcommands 362
START and STOP
usage considerations 294
STATIC 363
STATUS 128
STATUS_CODE 116
STOP
on INTERVAL subcommand 301
on SELECT and REJECT subcommands 363
STORCLAS
on OUTPUT, EXTRACT, and SCRAP subcommands 310
on QPF_OPTIONS UNLOAD_DSN subcommand 144
on REQUEUE SPILL subcommand 278
SWAP 172
SYSID 364
SYSTEM_WAIT 136
TEMP_RECORDS 117
THRESHOLD%
on QPF_OPTIONS ENFORCE statement 117
on QPF_OPTIONS MONITOR_PHASE subcommand 100
on QPF_OPTIONS OVERFLOW_PHASE subcommand 137
on QPF_OPTIONS PROCESS statement 129
on QPF_OPTIONS PROTECT_PHASE subcommand 106
TITLE 173
TMEMBER
on QPF_OPTIONS ENFORCE statement 118
on QPF_OPTIONS PROCESS statement 129
on SELECT and REJECT subcommands 365
TPIPE
on QPF_OPTIONS ENFORCE statement 118
on QPF_OPTIONS PROCESS statement 130
on SELECT and REJECT subcommands 366
SCRAP 249
SCRAP_DSN 163
SCRAPCODE 360
SEGMENT 361
SEGNO 116
SEGSIZE 116
SERVER_APPLID 170
SERVER_NAME
on CUSTOMIZE LIST_OPTIONS subcommand 168
on CUSTOMIZE SERVER_OPTIONS subcommand 171
SKIP_ORIGINS 68
SKIP_PROBLEMS 68
SMB 362
SORT_BY 208
SORTCORE 171
SPILL1_DSN 163
SPILL2_DSN 164
SPILL3_DSN 164
SPILL4_DSN 165
SQ_MAXQBUF
on QPF_OPTIONS command 94
SSCTNAME 172
<table>
<thead>
<tr>
<th>Keyword</th>
<th>On which command/option</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPNAME</td>
<td>On QPF_ACTION command</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>On QPF_OPTIONS ENFORCE</td>
<td>118</td>
</tr>
<tr>
<td></td>
<td>On QPF_OPTIONS PROCESS</td>
<td>130</td>
</tr>
<tr>
<td></td>
<td>On SELECT and REJECT subcommands</td>
<td>367</td>
</tr>
<tr>
<td>TRACE</td>
<td>On CUSTOMIZE SERVER_OPTIONS subcommand</td>
<td>173</td>
</tr>
<tr>
<td></td>
<td>On QPF_OPTIONS command</td>
<td>94</td>
</tr>
<tr>
<td>TRACE_SIZE</td>
<td></td>
<td>95</td>
</tr>
<tr>
<td>TRANID</td>
<td></td>
<td>413</td>
</tr>
<tr>
<td>TRANSACTION</td>
<td>On QPF_OPTIONS ENFORCE</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>On SELECT and REJECT subcommands</td>
<td>367</td>
</tr>
<tr>
<td>TYPE</td>
<td>On DEQUEUE command</td>
<td>190</td>
</tr>
<tr>
<td></td>
<td>On DISPLAY TYPE=ANALYZER command</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td>On DISPLAY TYPE=DEST_QUEUES command</td>
<td>209</td>
</tr>
<tr>
<td></td>
<td>On DISPLAY TYPE=DESTINATION command</td>
<td>203</td>
</tr>
<tr>
<td></td>
<td>On DISPLAY TYPE=MESSAGE command</td>
<td>213</td>
</tr>
<tr>
<td></td>
<td>On DISPLAY TYPE=RECORD command</td>
<td>217</td>
</tr>
<tr>
<td></td>
<td>On DISPLAY TYPE=STATISTICS command</td>
<td>218</td>
</tr>
<tr>
<td></td>
<td>On QPF_ACTION command</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>On QPF_LIST command</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>On QPF_OPTIONS command</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>On QPF_OPTIONS ENFORCE</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>On QPF_OPTIONS UNLOAD_DSN subcommand</td>
<td>144</td>
</tr>
<tr>
<td></td>
<td>On REQUEUE command</td>
<td>250</td>
</tr>
<tr>
<td>UNIT</td>
<td>On OUTPUT, EXTRACT, and SCRAO subcommands</td>
<td>311</td>
</tr>
<tr>
<td></td>
<td>On QPF_OPTIONS UNLOAD_DSN subcommand</td>
<td>145</td>
</tr>
<tr>
<td></td>
<td>On REQUEUE command</td>
<td>274</td>
</tr>
<tr>
<td></td>
<td>On REQUEUE SPILL subcommand</td>
<td>278</td>
</tr>
<tr>
<td>UNLOAD_DSN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>On CUSTOMIZE IMS_OPTIONS subcommand</td>
<td>165</td>
</tr>
</tbody>
</table>

**UOWID** 367
**UPPERCASE** 173
**USER_PREFIX** 252
**USERID** 368
**VALIDATE_SRC** 253
**VIRTUAL_CREATE** 166

**VOLSER**
- On OUTPUT, EXTRACT, and SCRAO subcommands 311
- On QPF_OPTIONS UNLOAD_DSN subcommand 145
- On REQUEUE command 274
- On REQUEUE SPILL subcommand 278

**VSPCNT** 369
**VTAM_LOGMODE** 174
**VTAM_LU_PREFIX** 174

**WAIT**
- On PLEXCMD command 226
- On QPF_ACTION command 61

**WARNING**
- On QMRAPI EXEC function call 409
- On QMRAPI TERM function call 411

**WTO_CMDS** 166
**WTO_DESC**
- On CUSTOMIZE SERVER_OPTIONS subcommand 175
- On QPF_OPTIONS MONITOR_PHASE subcommand 101
- On QPF_OPTIONS OVERFLOW_PHASE subcommand 137
- On QPF_OPTIONS PROTECT_PHASE subcommand 107

**WTO_ROUTCDE**
- On CUSTOMIZE SERVER_OPTIONS subcommand 175
- On QPF_OPTIONS MONITOR_PHASE subcommand 101
- On QPF_OPTIONS OVERFLOW_PHASE subcommand 138
- On QPF_OPTIONS PROTECT_PHASE subcommand 107

**WTOR**
on QPF_OPTIONS MONITOR_PHASE
subcommand 102
on QPF_OPTIONS
OVERFLOW_PHASE
subcommand 138
on QPF_OPTIONS PROTECT_PHASE
subcommand 107
WTOR_HIDE 175
ZERO_TIME 253

L

LABEL keyword
on COND pseudo command 403
on INTERVAL subcommand 299
on REQUEUE CHANGE subcommand 259
on REQUEUE GROUP subcommand 271
on SELECT and REJECT subcommands 335
LAST_USED keyword 126
LIMIT keyword
on DISPLAY TYPE=DEST_QUEUES command 205
on DISPLAY TYPE=DESTINATION command 201
on DISPLAY TYPE=MESSAGE command 211
LIST_OPTIONS subcommand 167
LIST_QUEUES keyword 205
Load Balancer
definition 421
LOG keyword
on DEQUEUE command 187
on DEQUEUE command for shared queues 372
LOGCODE keyword
on CUSTOMIZE LIST_OPTIONS subcommand 160
on QPF_OPTIONS command 91
LTERM keyword 336
LUNAME keyword
on QPF_OPTIONS ENFORCE statement 113
on QPF_OPTIONS PROCESS statement 126
on SELECT and REJECT subcommands 336

M

MAX_DSNS keyword 142
MAX_REGION keyword 91
MAXDATA keyword 244
MAXMSGS keyword
on DISPLAY TYPE=MESSAGE command 211
on REQUEUE command 245
MAXRC keyword 404
MEMBER keyword 56
Message Advisor
overview 19
Message Advisor Server
definition 421
message exits
definition 421
messages pane
definition 421
MF keyword
on QMRAPI EXEC function call 409
on QMRAPI GET function call 410
on QMRAPI INIT function call 407
MFSNAME keyword 337
MGMTCLAS keyword
on OUTPUT, EXTRACT, and SCRAP
subcommands 309
on QPF_OPTIONS UNLOAD_DSN
subcommand 143
on REQUEUE SPILL subcommand 277
middleware tier
definition 421
MODE keyword
on DEQUEUE command 188
on QPF_OPTIONS PROTECT_PHASE
subcommand 103
on REQUEUE command 245
MODEL_DSNAME keyword
on OUTPUT, EXTRACT, and SCRAP
subcommands 310
on QPF_OPTIONS UNLOAD_DSN
subcommand 143
MOVEPAGE keyword 92
MSGTYPE keyword
description 338
MSNAME keyword 340

N

NAME keyword
on QPF_ACTION command 57
on QPF_OPTIONS ENFORCE statement 113
navigation pane
definition 422
NETID keyword 58
NEWAFFINITY keyword 260
NEWCOMMITMODE keyword 260
NEWDEST keyword 260
NEWER_THAN and OLDER_THAN keywords
usage considerations 297
NEWER_THAN keyword
on INTERVAL subcommand 299
on SELECT and REJECT subcommands 340
NEWLUNAME keyword 261
NEWMFSNAME keyword 262
NEWNODENAME keyword 262
NEWOIMS keyword 262
NEWORGQMANAGER keyword 264
NEWORGQNAME keyword 263
NEWORGTMEMBER keyword 264
NEWORGTPIPE keyword 264
NEWORIGIN keyword 263
NEWPIMS keyword 265
NEWQUEUE keyword 265
NEWSAFGROUP keyword 266
NEWSYSID keyword 266
NEWTIMESTAMP keyword 267
NEWTMEMBER keyword 267
NEWTPPIPE keyword 268
NEWTPNAME keyword 268
NEWUSERID keyword 269
node
definition 422
NODE keyword 114
NODE_NAME keyword
on QMRAPI DISPLAY command definition 412
NODENAME keyword
on SELECT and REJECT subcommands 340
NOTIFY keyword
on QPF_OPTIONS command 92
on QPF_OPTIONS ENFORCE statement 114
on QPF_OPTIONS MONITOR_PHASE 97
on QPF_OPTIONS OVERFLOW_PHASE
subcommand 134
on QPF_OPTIONS PROCESS statement 127
on QPF_OPTIONS PROTECT_PHASE
subcommand 103
NOTIFY_INTERVAL keyword
on QPF_OPTIONS MONITOR_PHASE
subcommand 98
on QPF_OPTIONS OVERFLOW_PHASE
subcommand 135
on QPF_OPTIONS PROTECT_PHASE
subcommand 105
NOTIFY% keyword
on QPF_OPTIONS MONITOR_PHASE
subcommand 98
on QPF_OPTIONS OVERFLOW_PHASE
subcommand 134
on QPF_OPTIONS PROTECT_PHASE
subcommand 104
NRE_SNAPQ keyword 160

O
OFFSET keyword 341
OIMS keyword 342
OLDER_THAN keyword
on INTERVAL subcommand 300
on SELECT and REJECT subcommands 342
OLDS keyword 246
online Help 16
ORGQMANAGER keyword 344
ORGQNAME keyword 344
ORGTMEMBER keyword 346
ORGTPIPE keyword 345
ORIGIN keyword 343
ORIGINS keyword 67
OTMA_LIMIT keyword 115
OTMAQ keyword 346
OTOKEN keyword 347
OUTPUT subcommand
description 304
overview 303
OUTPUT, EXTRACT, and SCRAP subcommands
description 304
overview 303
syntax diagram 304
OVERFLOW_PHASE
keywords usage example 85
override options
definition 422

P
parameter
definition 422
PIMS keyword 347
PLEXCMD
overview 221
sample command set
execute and IMSPLEX command 224
PLEXCMD command
Q

QMR (Queue Manager Utilities)
  task locations 41
QMR WTOR command
  sample command sets
    cancel a QMR task by using an operator command 28

QMRAPI

DISPLAY command definition
  description 411
END command definition
  description 414
EXEC function call
  description 408
function calls
  overview 405
GET function call
  description 409
IMSCMD command definition
  description 414
INCLUDE/EXCLUDE command definition
  description 412
INIT function call
  description 407
overview 405
TERM function call
  description 410
QPF (Queue Protection Facility)

Q

QMR (Queue Manager Utilities)
  task locations 41
QMR WTOR command
  sample command sets
    cancel a QMR task by using an operator command 28

QMRAPI
display a problem list
   through a QPF_LIST command set 64
   through the QPF processing WTOR
   LIST command 64
display current QPF options 65
display the QPF problem list
   from the Message Advisor ISPF
   interface panels 64
initiate an action
   through a problem list displayed on
   the Message Advisor ISPF
   interface panels 48
   through a QPF processing WTOR
   action command 48
   through a QPF_ACTION command set
   48
QPF_ACTION command
   STOP action commands by device type 49
QPF_OPTIONS command
   description 86
   ENFORCE statement keywords usage
   examples 84
   ENFORCE statement usage
   considerations 108
   general keywords 81, 86
   MONITOR_PHASE
      keywords usage example 82
   OVERFLOW_PHASE
      keywords usage example 85
   overview 71
   PROCESS keywords usage example 83
   PROCESS statement usage considerations 122
   PROTECT keywords usage example 82
   sample command sets 79
      activate QPF options 80
      activate QPF_OPTIONS 80
   syntax diagram 73
   UNLOAD_DSN keywords usage example 84
   UNLOAD_DSN subcommand
      usage considerations 139
QPOSITION keyword 349
   relative position 350
QTYPE keyword
   on SELECT and REJECT subcommands 351
   on SELECT and REJECT subcommands for
   shared queues 394
QUEUE keyword
   on QPF_OPTIONS PROCESS statement 128
   on SELECT and REJECT subcommands 356

R

RANGE_FROM keyword 413
RANGE_TO keyword 413
RATE keyword
on CUSTOMIZE IMS_OPTIONS subcommand
160
on DEQUEUE command 188
on DISPLAY TYPE=DEST_QUEUES command
206
on DISPLAY TYPE=DESTINATION command
201
on DISPLAY TYPE=MESSAGE command 212
on REQUEUE command 246
on UNLOAD command 290
RCNT keyword 357
RECON_DYNALLOC keyword 162
RECON1_DSN keyword 161
RECON2_DSN keyword 161
RECON3_DSN keyword 162
REGION keyword
  on QPF_ACTION command 59
  on SELECT and REJECT subcommands 358
REGION_ID keyword 359
related publications 16
REPLACE keyword 269
REPORT keyword
  on DEQUEUE command 188
  on REQUEUE command 247
  on UNLOAD command 290
REQ_PROMPT keyword
  on CUSTOMIZE SERVER_OPTIONS
    subcommand 169
  on REQUEUE command 248
request
  definition 423
requeue
  from one system to another 228
REQUEUE command
and conditional steps 231
CHANGE subcommand
  description 258
  overview 254
  syntax diagram 258
description 237
for shared queues
  sample command sets 384
  usage considerations 385
GROUP subcommand
  description 271
  overview 270
  syntax diagram 271
in a shared queues environment 384
input for requeues 229
INPUT subcommand
  description 272
  syntax diagram 272
overview 227
performance and INCORE keyword 231
requeue from one system to another 228
sample command sets
  overview 235
  requeue all messages 235
  requeue all messages following /ERE
    failure 235
  requeue all messages using a specific
    checkpoint 235
  requeue all messages using a specific
    input file 235
  requeue messages to a specific LTERM 235
  requeue transaction messages 235
SPILL subcommand
  description 275
  overview 275
  syntax diagram 275
support for uncataloged SLDS 230
syntax diagram 233
user exit 230
REQUEUE command for shared queues
sample command sets
  REQUEUE TYPE=FILE 385
RESET keyword 93
RESET_TIME keyword
on QPF_OPTIONS MONITOR_PHASE subcommand 99
on QPF_OPTIONS OVERFLOW_PHASE subcommand 136
on QPF_OPTIONS PROTECT_PHASE subcommand 105
RESET% keyword
on QPF_OPTIONS MONITOR_PHASE subcommand 99
on QPF_OPTIONS OVERFLOW_PHASE subcommand 135
on QPF_OPTIONS PROTECT_PHASE subcommand 105
RETRY keyword 143
RETURN_EMPTY keyword 412
ROUTE keyword 225
RSMB keyword 359

S

SAFGROUP keyword 360
sample command sets

COND pseudo command, Message Advisor request with COND command steps 401
CUSTOMIZE command 151
activate automatic requeue 151
list Message Advisor Servers and IMS systems for which options have been defined 151
list options for a specific IMS ID 151
list options for a specific Message Advisor Server 151
DEQUEUE command 180
dequeue from multiple destinations 180
dequeue from one destination 180
DEQUEUE command for shared queues 371
DISPLAY command 198
display a formatted message record 198
display a message record 198
display detailed destination report 198
display message queue statistics 198
display summary destination queues report 198
DISPLAY command for shared queues 374
display a message record 374
display detailed destination report 374
display message queue statistics 374
display summary destination queues report 374
IMSCMD command 222
execute an IMS command 222
PLEXCMD command 224
execute an IMSPLEX command 224
QMR WTOR command
cancel a QMR task by using an operator command 28
QPF_ACTION command 53
initiate a QPF action 53
QPF_LIST command 65
display the QPF problem list 65
QPF_OPTIONS command
activate QPF options 80
activate QPF_OPTIONS 80
overview 79
REQUEUE command
overview 235
requeue all messages 235
requeue all messages following /ERE failure 235
requeue all messages using a specific checkpoint 235
requeue all messages using a specific input file 235
requeue messages to a specific LTERM 235
requeue transaction messages 235
REQUEUE command for shared queues 384
REQUEUE TYPE=FILE 385
UNLOAD command 284
unload from multiple destinations 284
unload from one destination 284
UNLOAD command for shared queues 386
unload from multiple destinations 386
unload from one destination 386
SCOPE keyword
  on DEQUEUE command 189
  on DEQUEUE command for shared queues 372
  on DISPLAY TYPE=DEST_QUEUES command 206
  on DISPLAY TYPE=DEST_QUEUES command for shared queues 378
  on DISPLAY TYPE=DESTINATION command 201
  on DISPLAY TYPE=DESTINATION command for shared queues 376
  on DISPLAY TYPE=MESSAGE command 212
  on DISPLAY TYPE=MESSAGE command for shared queues 380
  on DISPLAY TYPE=RECORD command 216
  on DISPLAY TYPE=RECORD command for shared queues 382
  on UNLOAD command 291
  on UNLOAD command for shared queues 388
SCRAP keyword 249
SCRAP subcommand
description 304
overview 303
SCRAP_DSN keyword 163
SCRAPCODE keyword 360
SEGMENT keyword 361
SEGNO keyword 116
SEGSIZE keyword 116
SELECT and REJECT subcommands
description 320
for shared queues
  usage considerations 389
invalid command and keyword combinations 318
overview 313
syntax diagram 314
SERVER_APPLID keyword 170
SERVER_NAME keyword
  on CUSTOMIZE LIST_OPTIONS subcommand 168
  on CUSTOMIZE SERVER_OPTIONS subcommand 171
SERVER_OPTIONS subcommand
description 168
session
definition 423
shared queues
  DEQUEUE command
description 371, 372
  DISPLAY command
description 374
  DISPLAY TYPE=ANALYZER command
description 384
  DISPLAY TYPE=DEST_QUEUES command
description 377
  DISPLAY TYPE=DESTINATION command
description 376
  DISPLAY TYPE=MESSAGE command
description 379
  DISPLAY TYPE=RECORD command
description 381
  DISPLAY TYPE=STATISTICS command
description 383
  REQUEUE command
  usage considerations 385
  UNLOAD command
description 386, 387
SKIP_ORIGINS keyword 68
SKIP_PROBLEMS keyword 68
SMB keyword 362
SORT_BY keyword 208
SORTCORE keyword 171
spare element pool
definition 423
SPILL subcommand
description 275
  syntax diagram 275
SPILL1_DSN keyword 163
SPILL2_DSN keyword 164
SPILL3_DSN keyword 164
SPILL4_DSN keyword 165
SSCTNAME keyword 172
START and STOP keywords
  time stamp format 295
  time zone label 296
  time zone offset 296
  usage considerations 294
START keyword
  on INTERVAL subcommand 300
  on SELECT and REJECT subcommands 362
STATIC keyword 363
STATUS keyword 128
STATUS_CODE keyword 116
STOP keyword
  on INTERVAL subcommand 301
  on SELECT and REJECT subcommands 363
STORCLAS keyword
  on OUTPUT, EXTRACT, and SCRAP
    subcommands 310
  on QPF_OPTIONS UNLOAD_DSN
    subcommand 144
  on REQUEUE SPILL subcommand 278
subcommand descriptions
  CHANGE 258
  EXTRACT 304
  GROUP 271
  IMS_OPTIONS 153
  INPUT 272
  INTERVAL subcommand 299
  LIST_OPTIONS 167
  OUTPUT 304
  OUTPUT, EXTRACT, and SCRAP 304
  SCRAP 304
  SELECT and REJECT 320
  SERVER_OPTIONS 168
  SPILL 275
subcommands

CHANGE
description 258
  syntax diagram 258
EXTRACT
description 304
    overview 303
GROUP
description 271
    overview 270
    syntax diagram 271
IMS_OPTIONS
description 153
INPUT
description 272
    overview 272
    syntax diagram 272
INTERVAL
description 299
    overview 293
    syntax diagram 298
keywords common to SELECT, REJECT,
CHANGE, and GROUP subcommands 254, 318
LIST_OPTIONS
description 167
OUTPUT
description 304
    overview 303
OUTPUT, EXTRACT, and SCRAP
    overview 303
    syntax diagram 304
SCRAP
description 304
    overview 303
SELECT and REJECT
description 320
    invalid command and keyword
      combinations 318
    overview 313
    syntax diagram 314
SELECT and REJECT for shared queues
    usage considerations 389
SELECT, REJECT, CHANGE, and GROUP
    common keywords 318
SERVER_OPTIONS
description 168
SPILL
description 275
    overview 275
    syntax diagram 275
UNLOAD_DSN
  usage considerations 139
SWAP keyword 172
symbolic keywords
  invalid data set name examples 37
  valid data set names 37
syntax diagrams
  CHANGE subcommand 258
  COND pseudo command 400
  CUSTOMIZE command 149
  DISPLAY command 195
  GROUP subcommand 271
  IMSCMD command 222
  INPUT subcommand 272
  INTERVAL subcommand 298
  OUTPUT, EXTRACT, and SCRAP subcommands 304
  PLEXCMD command 223
  QPF_OPTIONS command 73
  REQUEUE command 233
  SELECT and REJECT subcommands 314
  SPILL subcommand 275
  UNLOAD command 283
SYSID keyword 364
SYSTEM_WAIT keyword 136

T
  task pane
    definition 423
  TEMP_RECORDS keyword 117
  TERM Subroutine
    definition 423
  THRESHOLD% keyword
    on QPF_OPTIONS ENFORCE statement 117
    on QPF_OPTIONS MONITOR_PHASE subcommand 100
    on QPF_OPTIONS OVERFLOW_PHASE subcommand 137
    on QPF_OPTIONS PROCESS statement 129
    on QPF_OPTIONS PROTECT_PHASE subcommand 106
time stamp, START and STOP keywords 295
time zone label, START and STOP keywords 296
time zone offset, START and STOP keywords 296
TITLE keyword 173
TMEMBER keyword
  on QPF_OPTIONS ENFORCE statement 118
  on QPF_OPTIONS PROCESS statement 129
  on SELECT and REJECT subcommands 365
TPIPE keyword
  on QPF_OPTIONS ENFORCE statement 118
  on QPF_OPTIONS PROCESS statement 130
  on SELECT and REJECT subcommands 366
TPNAME keyword
  on QPF_ACTION command 59
  on QPF_OPTIONS ENFORCE statement 118
  on QPF_OPTIONS PROCESS statement 130
  on SELECT and REJECT subcommands 367
TRACE keyword
  on CUSTOMIZE SERVER_OPTIONS subcommand 173
  on QPF_OPTIONS command 94
  on QPF_ACTION command 59
  on QPF_OPTIONS ENFORCE statement 118
TRACE_SIZE keyword 95
TRANID keyword 413
TRANSACTION keyword
  on QPF_OPTIONS ENFORCE statement 119
  on QPF_OPTIONS PROCESS statement 130
  on SELECT and REJECT subcommands 367
Translate Subsystem Services
  definition 424
  tree
    definition 424
    tree node
    definition 424
  TYPE and NAME positional parameters
    allowed values for destinations 21
    allowed values for input devices 21
    allowed values for regions 21
  TYPE keyword
    on DEQUEUE command 190
    on DISPLAY TYPE=ANALYZER command 220
    on DISPLAY TYPE=DEST_QUEUES command 209
    on DISPLAY TYPE=DESTINATION command 203
    on DISPLAY TYPE=MESSAGE command 213
    on DISPLAY TYPE=RECORD command 217
    on DISPLAY TYPE=STATISTICS command 218
    on QPF_ACTION command 60
    on QPF_LIST command 69
    on QPF_OPTIONS command 95
    on QPF_OPTIONS ENFORCE statement 119
    on QPF_OPTIONS UNLOAD_DSN subcommand 144
    on REQUEUE command 250
U

UIM server
definition 424

UIMx
definition 424

uncataloged SLDS
  REQUEUE command support 230
undock
definition 424

UNIT keyword
  on OUTPUT, EXTRACT, and SCRAP
    subcommands 311
  on QPF_OPTIONS UNLOAD_DSN
    subcommand 144
  on REQUEUE command 274
  on REQUEUE SPILL subcommand 278

unknown destination
definition 424

UNLOAD command
  and conditional steps 282
    description 286
  for shared queues
    description 386, 387
    sample command sets 386
  overview 281
  sample command sets 284
    unload from multiple destinations 284
    unload from one destination 284
  syntax diagram 283

UNLOAD command for shared queues
  sample command sets
    unload from multiple destinations 386
    unload from one destination 386

UNLOAD_DSN keyword
  on CUSTOMIZE IMS_OPTIONS subcommand 165

UNLOAD_DSN subcommand
  usage considerations 139

Unsolicited Output feature
  definition 425

UOWID keyword 367

UPDS
definition 425

UPPERCASE keyword 173
  uppercase-only feature with CUSTOMIZE command 148

USER
definition 425
  user access profile
    definition 425
  user authorization profile
    definition 425
  user exit
    for REQUEUE command 230
  user message exit
    definition 425
  user profile data set
    definition 425
  USER_PREFIX keyword 252
  user/SPQB
    definition 425
  USERID keyword 368

V

VALIDATE_SRC keyword 253
  virtual exit
    definition 426
  virtual LTERM
    definition 426
    virtual LTERM creation feature with CUSTOMIZE command 148
  virtual printer
    definition 426
  virtual terminal
    definition 426
    virtual terminal control block
    definition 426
  VIRTUAL_CREATE keyword 166

VOLSER keyword
  on OUTPUT, EXTRACT, and SCRAP
    subcommands 311
  on QPF_OPTIONS UNLOAD_DSN
    subcommand 145
  on REQUEUE command 274
  on REQUEUE SPILL subcommand 278

VSPCNT
definition 426

VSPCNT keyword 369

VTAM_LOGMODE keyword 174

VTAM_LU_PREFIX keyword 174

W

WAIT keyword
on PLEXCMD command 226
on QPF_ACTION command 61
WARNING keyword
  on QMRAPI EXEC function call 409
  on QMRAPI TERM function call 411
WorkLoad Governor
definition 427
WTO_CMDS keyword 166
WTO_DESC keyword
  on CUSTOMIZE SERVER_OPTIONS
    subcommand 175
  on QPF_OPTIONS MONITOR_PHASE
    subcommand 101
  on QPF_OPTIONS OVERFLOW_PHASE
    subcommand 137
  on QPF_OPTIONS PROTECT_PHASE
    subcommand 107
WTO_ROUTCDE keyword
  on CUSTOMIZE SERVER_OPTIONS
    subcommand 175
  on QPF_OPTIONS MONITOR_PHASE
    subcommand 101
  on QPF_OPTIONS OVERFLOW_PHASE
    subcommand 138
  on QPF_OPTIONS PROTECT_PHASE
    subcommand 107
WTOR
executing a command set 36
WTOR keyword
  on QPF_OPTIONS MONITOR_PHASE
    subcommand 102
  on QPF_OPTIONS OVERFLOW_PHASE
    subcommand 138
  on QPF_OPTIONS PROTECT_PHASE
    subcommand 107
WTOR_HIDE keyword 175

Z

ZERO_TIME keyword 253