

WebSphere MQ for z/OS



# System Setup Guide

*Version 7.0*



WebSphere MQ for z/OS



# System Setup Guide

*Version 7.0*

**Note**

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

**First edition (April 2008)**

This edition of the book applies to the following products:

- IBM WebSphere MQ for z/OS, Version 7.0

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

© Copyright International Business Machines Corporation 1993, 2008. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

**Figures . . . . . v**

**Tables . . . . . vii**

## **Chapter 1. Customizing your queue managers . . . . . 1**

Preparing for customization . . . . .	1
Installable features . . . . .	1
Libraries that exist after installation. . . . .	2
Customizing your queue managers . . . . .	3
Before you start . . . . .	4
Customization summary . . . . .	5
Task 1: Identify the z/OS system parameters . . . . .	7
Task 2: APF authorize the WebSphere MQ load libraries . . . . .	7
Task 3: Update the z/OS link list and LPA . . . . .	8
Task 4: Update the z/OS program properties table. . . . .	10
Task 5: Define the WebSphere MQ subsystem to z/OS . . . . .	10
Task 6: Create procedures for the WebSphere MQ queue manager . . . . .	14
Task 7: Create procedures for the channel initiator. . . . .	15
Task 8: Define the WebSphere MQ subsystem to a z/OS WLM service class . . . . .	17
Task 9: Set up the DB2 environment . . . . .	17
Task 10: Set up the Coupling Facility . . . . .	19
Task 11: Implement your ESM security controls . . . . .	19
Task 12: Update SYS1.PARMLIB members . . . . .	21
Task 13: Customize the initialization input data sets . . . . .	21
Task 14: Create the bootstrap and log data sets . . . . .	24
Task 15: Define your page sets . . . . .	25
Task 16: Add the WebSphere MQ entries to the DB2 data-sharing group . . . . .	25
Task 17: Tailor your system parameter module . . . . .	26
Task 18: Tailor the channel initiator parameters . . . . .	45
Task 19: Set up Batch, TSO, and RRS adapters . . . . .	46
Task 20: Set up the operations and control panels . . . . .	47
Task 21: Include the WebSphere MQ dump formatting member. . . . .	49
Task 22: Suppress information messages. . . . .	49

## **Chapter 2. Migrating from a previous version . . . . . 51**

## **Chapter 3. Testing your queue manager 53**

Running the basic installation verification program . . . . .	53
Overview of the CSQ4IVP1 application . . . . .	53
Preparing to run CSQ4IVP1 . . . . .	53
Running CSQ4IVP1. . . . .	54
Checking the results of CSQ4IVP1. . . . .	54
Testing for queue-sharing groups . . . . .	57

Preparing to run CSQ4IVP1 for a queue-sharing group . . . . .	57
Running CSQ4IVP1 for a queue-sharing group . . . . .	58
Checking the results of CSQ4IVP1 for a queue-sharing group . . . . .	58
Testing for distributed queuing. . . . .	58
Overview of CSQ4IVPX job . . . . .	58
Preparing to run CSQ4IVPX. . . . .	58
Running CSQ4IVPX . . . . .	60
Checking the results of CSQ4IVPX. . . . .	60
Testing for C, C++, COBOL, PL/I, and CICS . . . . .	61

## **Chapter 4. Customizing for CICS . . . . 63**

Setting up the CICS adapter . . . . .	63
Resource definition . . . . .	63
System definition . . . . .	65
Completing the connection from CICS . . . . .	66
Controlling CICS application connections . . . . .	67
Customizing the CICS adapter . . . . .	67
Customizing the CICS bridge . . . . .	68
Setting up CICS . . . . .	69
Setting up WebSphere MQ . . . . .	69
Controlling CICS bridge throughput . . . . .	71

## **Chapter 5. Customizing for IMS . . . . . 73**

Setting up the IMS adapter . . . . .	73
Defining WebSphere MQ to IMS . . . . .	74
Defining WebSphere MQ queue managers to the IMS adapter . . . . .	77
Setting up the IMS trigger monitor . . . . .	78
Customizing the IMS bridge. . . . .	79

## **Chapter 6. Monitoring performance and resource usage . . . . . 81**

Introduction to monitoring . . . . .	81
Getting snapshots of WebSphere MQ. . . . .	81
Using CICS adapter statistics . . . . .	82
Using WebSphere MQ trace . . . . .	82
Using WebSphere MQ online monitoring . . . . .	84
Using WebSphere MQ events . . . . .	84
Using System Management Facility . . . . .	85
Using other products with WebSphere MQ. . . . .	86
Investigating performance problems . . . . .	87
Interpreting WebSphere MQ performance statistics . . . . .	88
Layout of an SMF type 115 record. . . . .	89
The SMF header. . . . .	89
Self-defining sections . . . . .	90
Examples of SMF statistics records. . . . .	91
Processing type 115 SMF records . . . . .	93
Storage manager data records . . . . .	93
Log manager data records . . . . .	94
Message manager data records . . . . .	95
Data manager data records . . . . .	95
Buffer manager data records. . . . .	95
Lock manager data records . . . . .	98

DB2 manager data records . . . . .	98
Coupling Facility manager data records . . . . .	103
Interpreting WebSphere MQ accounting data . . . . .	104
Layout of an SMF type 116 record . . . . .	104
Processing type 116 SMF records . . . . .	107
Common WebSphere MQ SMF header . . . . .	107
Thread cross reference data. . . . .	108
Message manager data records . . . . .	109
Sample subtype zero accounting record . . . . .	109
Thread-level and queue-level data records. . . . .	110
<b>Chapter 7. Setting up security . . . . .</b>	<b>123</b>
Using RACF classes and profiles . . . . .	123
Using RACF security classes . . . . .	123
RACF profiles . . . . .	124
Switch profiles . . . . .	126
Profiles used to control access to WebSphere MQ resources . . . . .	132
Profiles for connection security . . . . .	133
Profiles for queue security . . . . .	135
Profiles for processes . . . . .	145
Profiles for namelists . . . . .	146
Profiles for alternate user security . . . . .	147
Profiles for context security. . . . .	149
Profiles for command security. . . . .	151
Profiles for topic security . . . . .	158
Profiles for command resource security. . . . .	161
Using the RESLEVEL security profile . . . . .	163
The RESLEVEL profile . . . . .	163
RESLEVEL and batch connections . . . . .	164
RESLEVEL and system functions . . . . .	165
RESLEVEL and CICS connections . . . . .	165
RESLEVEL and IMS connections . . . . .	166
RESLEVEL and channel initiator connections . . . . .	167
RESLEVEL and intra-group queuing. . . . .	168
RESLEVEL and the user IDs checked . . . . .	169
User IDs for security checking. . . . .	170
User IDs for connection security . . . . .	170
User IDs for command security and command resource security . . . . .	170
User IDs for resource security (MQOPEN, MQSUB, and MQPUT1) . . . . .	171
Blank user IDs and UACC levels . . . . .	178
WebSphere MQ security management . . . . .	178
User ID reverification. . . . .	179
User ID timeouts . . . . .	179

Refreshing queue manager security . . . . .	180
Displaying security status . . . . .	183
Security installation tasks . . . . .	184
Auditing considerations . . . . .	188
Customizing security. . . . .	190
Security problem determination . . . . .	190
Security considerations for distributed queuing . . . . .	192
The channel initiator . . . . .	192
Cluster support. . . . .	194
Security considerations for using WebSphere MQ with CICS . . . . .	195
Controlling the security of CICS transactions supplied by WebSphere MQ . . . . .	195
CICS adapter user IDs . . . . .	196
Security considerations for the CICS bridge . . . . .	197
Security considerations for using WebSphere MQ with IMS . . . . .	199
Using the OPERCMDS class . . . . .	199
Security considerations for the IMS bridge. . . . .	199
Example security scenarios . . . . .	203
The two queue managers scenario . . . . .	203
The queue-sharing group scenario . . . . .	210
WebSphere MQ security implementation checklist . . . . .	215

**Chapter 8. Upgrading and applying service to Language Environment or z/OS Callable Services . . . . . 219**

Running a REPORT CALLLIBS job . . . . .	220
Running a LINK CALLLIBS job . . . . .	221

**Chapter 9. Using OTMA exits in IMS 223**

Exit names . . . . .	223
Specifying the destination resolution user exit name . . . . .	223
Naming convention for IMS destination . . . . .	223
A sample scenario. . . . .	223
The pre-routing exit DFSYPRX0 . . . . .	223
The destination resolution user exit . . . . .	225

**Notices . . . . . 229**

**Index . . . . . 233**

**Sending your comments to IBM . . . . . 241**

---

## Figures

1. PPT additional entries needed for the WebSphere MQ queue manager . . . . .	10	19. SMF record 115, subtype 1 . . . . .	92
2. PPT additional entries needed for the WebSphere MQ channel initiator . . . . .	10	20. SMF record 115, subtype 2 . . . . .	93
3. Sample IEFSSNss statements for defining subsystems. . . . .	12	21. Part of an SMF record 116 showing the header and self-defining sections . . . . .	107
4. RACF commands for CSQ4IVP1 . . . . .	54	22. Example SMF type 116, subtype zero record . . . . .	110
5. Sample report from CSQ4IVP1 . . . . .	57	23. Example SMF type 116, subtype 1 record . . . . .	120
6. RACF commands for CSQ4IVP1 for a queue-sharing group . . . . .	58	24. Example SMF type 116, subtype 2 record . . . . .	121
7. RACF commands for CSQ4IVPX . . . . .	60	25. Example SMF type 116, subtype 1 record with no WQST data records . . . . .	121
8. Example output from CSQ4IVPX . . . . .	61	26. Checking for subsystem security . . . . .	128
9. JCL fragment for upgrading the CICS CSD . . . . .	64	27. Checking for queue manager level security . . . . .	129
10. Sample PLT for use with the CICS adapter . . . . .	65	28. Checking for queue-sharing group level security . . . . .	129
11. Sample INITPARM statement to set the default connection values for CICS . . . . .	65	29. Logic flow for WebSphere MQ security caching . . . . .	182
12. Linking to the adapter connect program, CSQQCON, from a PLT program. . . . .	68	30. Typical output from the DISPLAY SECURITY command . . . . .	184
13. Sample JCL to link-edit the dynamic call stub . . . . .	74	31. Sample output from RACFRW showing RESLEVEL general audit records . . . . .	189
14. CSQQDEFX macro syntax. . . . .	77	32. Example security scenario . . . . .	204
15. Layout of a queue manager definition table . . . . .	78	33. Example SMP/E JOBCARD job step . . . . .	221
16. Example transaction definition for CSQQTRMN . . . . .	79	34. Example SMP/E REPORT CALLLIBS job . . . . .	221
17. Example PSB definition for CSQQTRMN . . . . .	79	35. Example SMP/E LINK CALLLIBS job . . . . .	222
18. Part of an SMF record 115 showing the header and self-defining sections . . . . .	91	36. OTMA pre-routing exit assembler sample . . . . .	225
		37. Sample assembler DRU exit. . . . .	227





---

## Tables

1. WebSphere MQ libraries that exist after installation . . . . .	2	39. Access levels for close options on permanent dynamic queues . . . . .	140
2. National language feature libraries . . . . .	4	40. RACF authority to the dead-letter queue and its alias . . . . .	142
3. Customization summary . . . . .	5	41. Access required to the SYSTEM queues by WebSphere MQ . . . . .	143
4. Subsystem name to CPF associations . . . . .	12	42. MQOPEN, MQPUT1, MQSUB, and MQCLOSE options and the security authorization required . . . . .	143
5. Example of CPF subset and superset rules . . . . .	12	43. Access levels for process security . . . . .	146
6. Valid character set for CPF strings . . . . .	13	44. Access levels for namelist security . . . . .	147
7. Members of thlqual.SCSQPROC. . . . .	23	45. Access levels for alternate user security . . . . .	148
8. Default values of CSQ6SYSP parameters . . . . .	28	46. Access levels for context security . . . . .	150
9. Default values of CSQ6LOGP parameters . . . . .	35	47. MQSC commands, profiles, and their access levels . . . . .	152
10. Default values of CSQ6ARVP parameters . . . . .	38	48. PCF commands, profiles, and their access levels . . . . .	155
11. SSM specifications options . . . . .	76	49. Access level required for topic security to subscribe . . . . .	159
12. Setting and displaying attributes to control online monitoring . . . . .	84	50. Access level required to profiles for topic security for closure of a subscribe operation . . . . .	160
13. SMF record 115 header description. . . . .	89	51. Access level required to profiles for topic security for a publish operation . . . . .	160
14. Offsets to self-defining sections . . . . .	90	52. Checks made at different RACF access levels for batch connections . . . . .	165
15. Problem symptoms that can be examined using log manager statistics . . . . .	94	53. Checks made at different RACF access levels for CICS connections . . . . .	166
16. DB2 statistics record (Q5ST) . . . . .	98	54. Checks made at different RACF access levels for IMS connections . . . . .	167
17. Coupling facility statistics record (QEST) . . . . .	103	55. Checks made at different RACF access levels for channel initiator connections . . . . .	168
18. SMF record header description. . . . .	104	56. Checks made at different RACF access levels for the intra-group queuing agent. . . . .	168
19. Offsets to self-defining sections . . . . .	105	57. User ID checking against profile name for batch connections . . . . .	171
20. Structure of the common WebSphere MQ SMF header record QWHS . . . . .	107	58. User ID checking against profile name for CICS-type user IDs . . . . .	171
21. Structure of the thread cross reference for a CICS system . . . . .	108	59. User ID checking against profile name for IMS-type user IDs . . . . .	172
22. Structure of the thread cross reference for an IMS system . . . . .	109	60. How the second user ID is determined for the IMS connection . . . . .	173
23. Layout of the task-related information (WTAS) structure . . . . .	111	61. User IDs checked against profile name for TCP/IP channels . . . . .	174
24. Layout of the Queue (WQ) structure . . . . .	113	62. User IDs checked against profile name for LU 6.2 channels . . . . .	175
25. Layout of the Task Id structure (WTID) . . . . .	118	63. User IDs checked against profile name for LU 6.2 and TCP/IP server-connection channels . . . . .	176
26. Structure of the WTIDCORI for a CICS system . . . . .	119	64. User IDs checked against profile name for intra-group queuing . . . . .	177
27. Structure of the WTIDCORI for an IMS system . . . . .	119	65. RACF access to data sets associated with a queue manager . . . . .	186
28. Meaning of channel names . . . . .	119	66. RACF access to data sets associated with distributed queuing . . . . .	186
29. RACF classes used by WebSphere MQ . . . . .	123	67. CICS bridge monitor security . . . . .	198
30. Switch profiles for subsystem level security . . . . .	127	68. CICS bridge task security . . . . .	198
31. Switch profiles for queue-sharing group or queue manager level security . . . . .	128	69. Security profiles for the example scenario . . . . .	207
32. Valid security switch combinations for queue manager level security . . . . .	129		
33. Valid security switch combinations for queue-sharing group level security . . . . .	130		
34. Valid security switch combinations for queue manager and queue-sharing group level security . . . . .	130		
35. Other valid security switch combinations that switch both levels of checking <b>on</b> . . . . .	130		
36. Switch profiles for resource checking . . . . .	131		
37. Access levels for queue security using the MQOPEN or MQPUT1 calls . . . . .	136		
38. Access levels for queue security using the MQSUB call . . . . .	137		

70. Sample security profiles for the batch application on queue manager QM1 . . . . .	208	76. Sample security profiles for the batch application on queue manager QM1 . . . . .	214
71. Sample security profiles for queue manager QM2 using TCP/IP and not SSL . . . . .	208	77. Sample security profiles for queue manager QM2 using TCP/IP . . . . .	214
72. Sample security profiles for queue manager QM2 using LU 6.2 . . . . .	209	78. Sample security profiles for queue manager QM2 using LU 6.2 . . . . .	215
73. Sample security profiles for queue manager QM2 using TCP/IP and SSL . . . . .	209	79. Service has been applied or the product has been upgraded to a new release . . . . .	219
74. Sample security profiles for the CICS application on queue manager QM1 . . . . .	210	80. One of the products has been updated to a new release in a new SMP/E environment and libraries . . . . .	220
75. Security profiles for the example scenario	213		

---

# Chapter 1. Customizing your queue managers

---

## Preparing for customization

The *WebSphere MQ for z/OS Program Directory* lists the contents of the WebSphere® MQ installation tape, the program and service level information for WebSphere MQ, and describes how to install WebSphere MQ under z/OS® using the System Modification Program Extended (SMP/E).

When you have installed WebSphere MQ, you must carry out a number of tasks before you can make it available to users. Refer to the following sections for a description of these tasks:

- “Customizing your queue managers” on page 3
- Chapter 3, “Testing your queue manager,” on page 53
- Chapter 7, “Setting up security,” on page 123

If you are migrating from a previous version of WebSphere MQ for z/OS, you do not need to perform most of the customization tasks. Refer to *WebSphere MQ Migration Information* for information about the tasks you have to perform.

## Installable features

WebSphere MQ for z/OS comprises the following features:

- Base** This is required; it comprises all the main functions, including
- Administration and utilities
  - Support for CICS®, IMS™ and batch type applications using the WebSphere MQ Application Programming Interface, or C++
  - Distributed queuing facility (supporting both TCP/IP and APPC communications)

### National language features

These contain error messages and panels in all the supported national languages. Each language has a language letter associated with it. The languages and letters are:

- |          |                           |
|----------|---------------------------|
| <b>C</b> | Simplified Chinese        |
| <b>E</b> | U.S. English (mixed case) |
| <b>K</b> | Japanese                  |
| <b>U</b> | U.S. English (uppercase)  |

You must install the US English (mixed case) option. You can also install one or more other languages. (The installation process for other languages requires US English (mixed case) to be installed, even if you are not going to use US English (mixed case).)

### Client Attachment feature

This is optional; it is only required if you are going to attach clients to your subsystem. When you have installed this feature, there are no

configuration parameters to set before you can attach clients to WebSphere MQ for z/OS. Administration for clients is available even if you do not install this feature.

### Java™ Support feature

This is optional; it is only required if you want to use Java and the Java Message Service. This is described in WebSphere MQ Using Java.

## Libraries that exist after installation

WebSphere MQ is supplied with a number of separate load libraries. Table 1 shows the libraries that might exist after you have installed WebSphere MQ.

Table 1. WebSphere MQ libraries that exist after installation

Name	Description
thlqual.SCSQANLC	Contains the load modules for the Simplified Chinese version of WebSphere MQ.
thlqual.SCSQANLE	Contains the load modules for the U.S. English (mixed case) version of WebSphere MQ.
thlqual.SCSQANLK	Contains the load modules for the Japanese version of WebSphere MQ.
thlqual.SCSQANLU	Contains the load modules for the U.S. English (uppercase) version of WebSphere MQ.
thlqual.SCSQASMS	Contains source for assembler sample programs.
thlqual.SCSQAUTH	The main repository for all WebSphere MQ product load modules; it also contains the default parameter module, CSQZPARM . This library must be APF-authorized and in PDS-E format.
thlqual.SCSQCICS	Contains extra load modules that must be included in the CICS DFHRPL concatenation.
thlqual.SCSQCLST	Contains CLISTs used by the sample programs.
thlqual.SCSQCOBC	Contains COBOL copybooks, including copybooks required for the sample programs.
thlqual.SCSQCOBS	Contains source for COBOL sample programs.
thlqual.SCSQCPPS	Contains source for C++ sample programs.
thlqual.SCSQC37S	Contains source for C sample programs.
thlqual.SCSQC370	Contains C headers, including headers required for the sample programs.
thlqual.SCSQDEFS	Contains side definitions for C++ and the DB2® DBRMs for shared queuing.
thlqual.SCSQEXEC	Contains REXX™ execs to be included in the SYSEXEC or SYSPROC concatenation if you are using the WebSphere MQ operations and control panels.
thlqual.SCSQHPPS	Contains header files for C++.
thlqual.SCSQINST	Contains JCL for installation jobs.
thlqual.SCSQLINK	Early code library. Contains the load modules that are loaded at system initial program load (IPL). The library must be APF-authorized.
thlqual.SCSQLOAD	Load library. Contains load modules for non-APF code, user exits, utilities, samples, installation verification programs, and adapter stubs. The library does not need to be APF-authorized and does not need to be in the link list.
thlqual.SCSQMACS	Contains Assembler macros including: sample macros, product macros, and system parameter macros.
thlqual.SCSQMAPS	Contains CICS mapsets used by sample programs.
thlqual.SCSQMSGC	Contains ISPF messages to be included in the ISPMLIB concatenation if you are using the Simplified Chinese language feature for the WebSphere MQ operations and control panels.
thlqual.SCSQMSGE	Contains ISPF messages to be included in the ISPMLIB concatenation if you are using the U.S. English (mixed case) language feature for the WebSphere MQ operations and control panels.
thlqual.SCSQMSGK	Contains ISPF messages to be included in the ISPMLIB concatenation if you are using the Japanese language feature for the WebSphere MQ operations and control panels.

Table 1. WebSphere MQ libraries that exist after installation (continued)

Name	Description
thlqual.SCSQMSGU	Contains ISPF messages to be included in the ISPMLIB concatenation if you are using the U.S. English (uppercase) language feature for the WebSphere MQ operations and control panels.
thlqual.SCSQMVR1	Contains the load modules for distributed queuing . This library must be APF-authorized and in PDS-E format.
thlqual.SCSQPLIC	Contains PL/I include files.
thlqual.SCSQPLIS	Contains source for PL/I sample programs.
thlqual.SCSQPMLA	Contains IPCS panels, for the dump formatter, to be included in the ISPPLIB concatenation. Also contains panels for WebSphere MQ sample programs.
thlqual.SCSQPMLC	Contains ISPF panels to be included in the ISPPLIB concatenation if you are using the Simplified Chinese language feature for the WebSphere MQ operations and control panels.
thlqual.SCSQPMLE	Contains ISPF panels to be included in the ISPPLIB concatenation if you are using the U.S. English (mixed case) language feature for the WebSphere MQ operations and control panels.
thlqual.SCSQPMLK	Contains ISPF panels to be included in the ISPPLIB concatenation if you are using the Japanese language feature for the WebSphere MQ operations and control panels.
thlqual.SCSQPMLU	Contains ISPF panels to be included in the ISPPLIB concatenation if you are using the U.S. English (uppercase) language feature for the WebSphere MQ operations and control panels.
thlqual.SCSQPROC	Contains sample JCL and default system initialization data sets.
thlqual.SCSQSMLC	Contains the load modules for the Simplified Chinese versions of the WebSphere MQ modules that are required for special purpose function (for example the early code).
thlqual.SCSQSMLE	Contains the load modules for the U.S. English (mixed case) versions of the WebSphere MQ modules that are required for special purpose function (for example the early code).
thlqual.SCSQSMLK	Contains the load modules for the Japanese versions of the WebSphere MQ modules that are required for special purpose function (for example the early code).
thlqual.SCSQSMLU	Contains the load modules for the U.S. English (uppercase) versions of the WebSphere MQ modules that are required for special purpose function (for example the early code).
thlqual.SCSQTMLC	Contains ISPF tables to be included in the ISPTLIB concatenation if you are using the Simplified Chinese language feature for the WebSphere MQ operations and control panels.
thlqual.SCSQTMLE	Contains ISPF tables to be included in the ISPTLIB concatenation if you are using the U.S. English (mixed case) language feature for the WebSphere MQ operations and control panels.
thlqual.SCSQTMLK	Contains ISPF tables to be included in the ISPTLIB concatenation if you are using the Japanese language feature for the WebSphere MQ operations and control panels.
thlqual.SCSQTMLU	Contains ISPF tables to be included in the ISPTLIB concatenation if you are using the U.S. English (uppercase) language feature for the WebSphere MQ operations and control panels.
<b>Attention:</b> Do <i>not</i> modify or customize any of these libraries. If you want to make changes, copy the libraries, and make your changes to the copies.	

## Customizing your queue managers

This chapter leads you through the various stages of customizing WebSphere MQ after you have successfully installed it. The installation process is described in the *WebSphere MQ for z/OS Program Directory*.

Samples are supplied with WebSphere MQ to help you with your customization. The sample data set members have names beginning with the four characters CSQ4 and are in the library thlqual.SCSQPROC.

## Before you start

Before you perform the customization tasks in this chapter, there are a number of configuration options that you should consider because they affect the performance and resource requirements of WebSphere MQ for z/OS. These options are discussed in the WebSphere MQ for z/OS Concepts and Planning Guide.

In the description of each task in this section, and in Table 3 on page 5, we indicate whether:

- The task is part of the process of customizing WebSphere MQ. That is, you perform the task once when you customize WebSphere MQ on the z/OS system. (In a parallel sysplex, you must perform the task for each z/OS system in the sysplex, and ensure that each z/OS system is set up identically.)
- The task is part of adding a queue manager. That is, you perform the task once for each queue manager when you add that queue manager.
- You need to perform the task when migrating. If you are migrating from a previous version of WebSphere MQ for z/OS, you might not need to perform all these tasks.

You should review the tasks when you apply corrective maintenance to WebSphere MQ and when you install a new version or release of WebSphere MQ.

None of the tasks require you to IPL your z/OS system, provided that you use commands to change the various z/OS system parameters, and perform “Task 12: Update SYS1.PARMLIB members” on page 21 as suggested.

We strongly recommend that all z/OS systems in a sysplex are set up identically, so that queue managers can be quickly created on any system in an emergency.

### Identify the national language support libraries

You need to specify the appropriate national language support libraries in the JCL that you will use for running WebSphere MQ (as described in the following sections). Each language is identified by a language letter:

- C** Simplified Chinese
- E** U.S. English (mixed case)
- K** Japanese
- U** U.S. English (uppercase)

Table 2 shows the names of the libraries for the language features; the language letter is the last letter of the library name.

Table 2. National language feature libraries

Description	Japanese	Simplified Chinese	U.S. English (mixed case)	U.S. English (uppercase)
Load modules	thlqual.SCSQANLK	thlqual.SCSQANLC	thlqual.SCSQANLE	thlqual.SCSQANLU
ISPF messages	thlqual.SCSQMSGK	thlqual.SCSQMSGC	thlqual.SCSQMSGE	thlqual.SCSQMSGU
ISPF panels	thlqual.SCSQPNLK	thlqual.SCSQPNLK	thlqual.SCSQPNE	thlqual.SCSQPNLU
Special purpose function (for example, early code)	thlqual.SCSQSNLK	thlqual.SCSQSNLC	thlqual.SCSQSNLE	thlqual.SCSQSNLU
ISPF tables	thlqual.SCSQTBLK	thlqual.SCSQTBLK	thlqual.SCSQTBLE	thlqual.SCSQTBLU

## Customization summary

The following table lists all the steps required to customize WebSphere MQ for z/OS. It also indicates the following:

- Whether the step has to be performed once only, or repeated for each queue manager.
- Whether you need to repeat the step for each queue-sharing group, or omit the step if you are not using queue-sharing groups. If you do not use queue-sharing groups initially, but subsequently want to do so, see *WebSphere MQ Migration* for the steps to take.
- Whether the step is required if you are migrating from a previous version of WebSphere MQ. Some steps might be needed, depending on what you decide about data set and queue manager names; these are marked 'Review'. For full details of migration, see *Migrating from a previous version WebSphere MQ Migration*.

Table 3. Customization summary

Task	Page	Required when migrating	Repeat for each queue manager	Queue-sharing groups
z/OS customization tasks				
"Task 1: Identify the z/OS system parameters" on page 7	"Task 1: Identify the z/OS system parameters" on page 7	Review	–	–
"Task 2: APF authorize the WebSphere MQ load libraries" on page 7	"Task 2: APF authorize the WebSphere MQ load libraries" on page 7	Review	–	–
"Task 3: Update the z/OS link list and LPA" on page 8	"Task 3: Update the z/OS link list and LPA" on page 8	Review	–	–
"Task 4: Update the z/OS program properties table" on page 10	"Task 4: Update the z/OS program properties table" on page 10	–	–	–
"Task 5: Define the WebSphere MQ subsystem to z/OS" on page 10	"Task 5: Define the WebSphere MQ subsystem to z/OS" on page 10	–	X	–
"Task 6: Create procedures for the WebSphere MQ queue manager" on page 14	"Task 6: Create procedures for the WebSphere MQ queue manager" on page 14	Review	X	–
"Task 7: Create procedures for the channel initiator" on page 15	"Task 7: Create procedures for the channel initiator" on page 15	Review	X	–

Table 3. Customization summary (continued)

Task	Page	Required when migrating	Repeat for each queue manager	Queue-sharing groups
"Task 8: Define the WebSphere MQ subsystem to a z/OS WLM service class" on page 17	"Task 8: Define the WebSphere MQ subsystem to a z/OS WLM service class" on page 17	–	X	–
"Task 9: Set up the DB2 environment" on page 17	"Task 9: Set up the DB2 environment" on page 17	Review	–	Omit if not used
"Task 10: Set up the Coupling Facility" on page 19	"Task 10: Set up the Coupling Facility" on page 19	Review	–	Repeat for each
"Task 11: Implement your ESM security controls" on page 19	"Task 11: Implement your ESM security controls" on page 19	Review	X	X
"Task 12: Update SYS1.PARMLIB members" on page 21	"Task 12: Update SYS1.PARMLIB members" on page 21	Review	–	–
WebSphere MQ customization tasks				
"Task 13: Customize the initialization input data sets" on page 21	"Task 13: Customize the initialization input data sets" on page 21	X	X	–
"Task 14: Create the bootstrap and log data sets" on page 24	"Task 14: Create the bootstrap and log data sets" on page 24	–	X	–
"Task 15: Define your page sets" on page 25	"Task 15: Define your page sets" on page 25	–	X	–
"Task 16: Add the WebSphere MQ entries to the DB2 data-sharing group" on page 25	"Task 16: Add the WebSphere MQ entries to the DB2 data-sharing group" on page 25	Review	X	Repeat for each
"Task 17: Tailor your system parameter module" on page 26	"Task 17: Tailor your system parameter module" on page 26	X	X	–
"Task 18: Tailor the channel initiator parameters" on page 45	"Task 18: Tailor the channel initiator parameters" on page 45	X	X	–
"Task 19: Set up Batch, TSO, and RRS adapters" on page 46	"Task 19: Set up Batch, TSO, and RRS adapters" on page 46	Review	–	–
"Task 20: Set up the operations and control panels" on page 47	"Task 20: Set up the operations and control panels" on page 47	Review	–	–
"Task 21: Include the WebSphere MQ dump formatting member" on page 49	"Task 21: Include the WebSphere MQ dump formatting member" on page 49	X	–	–



Table 3. Customization summary (continued)

Task	Page	Required when migrating	Repeat for each queue manager	Queue-sharing groups
"Task 22: Suppress information messages" on page 49	"Task 22: Suppress information messages" on page 49	–	–	–

## Task 1: Identify the z/OS system parameters

- You need to perform this task once for each z/OS system where you want to run WebSphere MQ.
- You might need to perform this task when migrating from a previous version. For details, see *WebSphere MQ Migration Information*.

Some of the tasks involve updating the z/OS system parameters. You need to know which ones were specified when the system IPL was performed. SYS1.PARMLIB(IEASYSpp) contains a list of parameters that point to other members of SYS1.PARMLIB (where pp represents the z/OS system parameter list that was used to IPL the system).

The entries you need to find are:

**For "Task 2: APF authorize the WebSphere MQ load libraries":**

PROG=xx or APF=aa point to the Authorized Program Facility (APF) authorized library list (member PROGxx or IEFAPFaa)

**For "Task 3: Update the z/OS link list and LPA" on page 8:**

LNK=kk points to the link list (member LNKLSTkk) LPA=mm points to the LPA list (member LPALSTmm)

**For "Task 4: Update the z/OS program properties table" on page 10:**

SCH=xx points to the Program Properties Table (PPT) (member SCHEDxx)

**For "Task 5: Define the WebSphere MQ subsystem to z/OS" on page 10:**

SSN=ss points to the defined subsystem list (member IEFSSNss)

## Task 2: APF authorize the WebSphere MQ load libraries

- You need to perform this task once for each z/OS system where you want to run WebSphere MQ.
- If you are using queue-sharing groups, you must ensure that the settings for WebSphere MQ are identical on each z/OS system in the sysplex.
- You might need to perform this task when migrating from a previous version. For details, see *WebSphere MQ Migration Information*.

The WebSphere MQ load libraries thlqual.SCSQAUTH and thlqual.SCSQLINK must be APF-authorized. You must also APF-authorize the libraries for your national language feature (thlqual.SCSQANLx and thlqual.SCSQSNLx) and for the distributed queueing feature (thlqual.SCSQMVR1).

However, all load modules in the LPA are automatically APF-authorized. So are all members of the link list if the SYS1.PARMLIB member IEASYSpp contains the statement:

```
LNKAUTH=LNKLST
```

LNKAUTH=LNKLST is the default if LNKAUTH is not specified.

Depending on what you choose to put in the LPA or linklist (see “Task 3: Update the z/OS link list and LPA”), you might not need to put the libraries in the APF link list

**Note:** You must APF-authorize all the libraries that you include in the WebSphere MQ STEPLIB. If you put a library that is not APF-authorized in the STEPLIB, the whole library concatenation loses its APF authorization.

The APF lists are in the SYS1.PARMLIB member PROGxx or IEAAPFaa. The lists contain the names of APF authorized z/OS libraries. The order of the entries in the lists is not significant. See the *MVS Initialization and Tuning Reference* manual for information about APF lists.

If you use PROGxx members with dynamic format, you need only issue the z/OS command SET PROG= for the changes to take effect. Otherwise, if you use static format or IEAAPFaa members, you must IPL your system.

### Task 3: Update the z/OS link list and LPA

- You need to perform this task once for each z/OS system where you want to run WebSphere MQ.
- If you are using queue-sharing groups, you must ensure that the settings for WebSphere MQ are identical on each z/OS system in the sysplex.
- You might need to perform this task when migrating from a previous version. For details, see *WebSphere MQ Migration Information*.

#### Early code

**Note:** In previous versions and releases of this product, we recommended that you include the early code load modules in a library in the link list. This is no longer required, and we now recommend that you do not include early code libraries in the link list.

Put the libraries with the latest version, release, or maintenance level of the WebSphere MQ early code (which is loaded at system IPL) into the libraries used for the z/OS LPA, replacing older versions of these libraries. These libraries are specified in an LPALSTmm member of SYS1.PARMLIB.

The early code comprises the following load modules:

- CSQ3INI and CSQ3EPX in the library thqual.SCSQLINK
- CSQ3ECMX in the library thqual.SCSQSNLx, where x is your language letter.

WebSphere MQ includes a user modification that moves the contents of the thqual.SCSQSNLx library into the thqual.SCSQLINK and informs SMP/E. This user modification is called CSQ8ERLY and is described in the *WebSphere MQ for z/OS Program Directory*.

When you have updated the early code in the LPA libraries, it is available from the next z/OS IPL (with the CLPA option) to all queue manager subsystems added during IPL from definitions in IEFSSNs members in SYS1.PARMLIB.

You can make it available immediately without an IPL for any new queue manager subsystem added subsequently (as described in “Task 5: Define the WebSphere MQ subsystem to z/OS” on page 10) by adding it to the LPA as follows:

- If you did not use CSQ8ERLY, issue these z/OS commands:  

```
SETPROG LPA,ADD,MODNAME=(CSQ3INI,CSQ3EPX),DSNAME=thqual.SCSQLINK  
SETPROG LPA,ADD,MODNAME=(CSQ3ECMX),DSNAME=thqual.SCSQSNLx
```
- If you did use CSQ8ERLY, you can load the early code into the LPA using the following z/OS command:  

```
SETPROG LPA,ADD,MASK=*,DSNAME=thqual.SCSQLINK
```

If you have applied maintenance, or you intend to restart a queue manager with a later version or release of WebSphere MQ, the early code can be made available to queue manager subsystems that are already defined, provided the level of the early code when the z/OS system was IPLed was at least that of Version 5.3. To make it available, do this:

1. Add it to the LPA using z/OS SETPROG commands as described above.
2. Stop the queue manager, using the WebSphere MQ command STOP QMGR.
3. Ensure that the qmgr.REFRESH.QMGR security profile is set up. See Table 47 on page 152.
4. Refresh the early code for the queue manager using the WebSphere MQ command REFRESH QMGR TYPE(EARLY).
5. Restart the queue manager, using the WebSphere MQ command START QMGR.

The WebSphere MQ commands STOP QMGR, REFRESH QMGR, and START QMGR are described in the WebSphere MQ Script (MQSC) Command Reference.

If the early code was below the Version 5.3 level, you must IPL the z/OS system (with the CLPA option) to make the updated early code available to existing queue manager subsystems. Thereafter, it can be updated and made available without an IPL.

## Other code

All the WebSphere MQ supplied load modules in the following libraries are reentrant and can be placed in the LPA:

- SCSQAUTH
- SCSQANLx, where x is your language letter
- SCSQMVR1

However, **if you place the libraries in the LPA, whenever you apply maintenance, you will have to copy any changed modules manually into the LPA.** Therefore, it is preferable to put the WebSphere MQ load libraries in the link list, which can be updated after maintenance by issuing the z/OS command REFRESH LLA.

This is particularly recommended for SCSQAUTH so that you do not have to include it in several STEPLIBs. Only one language library, SCSQANLx should be placed in the LPA or link list. The link list libraries are specified in an LNKLSTkk member of SYS1.PARMLIB.

The distributed queuing facility and CICS bridge (but not the queue manager itself) need access to the Language Environment® (LE) runtime library SCEERUN. If you use either of these facilities, you need to include SCEERUN in the link list.

## Task 4: Update the z/OS program properties table

- You need to perform this task once for each z/OS system where you want to run WebSphere MQ.
- If you are using queue-sharing groups, you must ensure that the settings for WebSphere MQ are identical on each z/OS system in the sysplex.
- You do not need to perform this task when migrating from a previous version.

If it is not already present, you must add the following entry to the program properties table (PPT), which you can find in SYS1.PARMLIB(SCHEDxx).

```
PPT PGMNAME(CSQYASCP) /* CSQ - THIS IS REQUIRED FOR WEBSPPHERE MQ */
CANCEL /* CAN BE CANCELLED */
KEY(7) /* STORAGE PROTECTION KEY */
SWAP /* PROGRAM IS SWAPPABLE */
NOPRIV /* NOT PRIVILEGED */
DSI /* REQUIRES DATA SET INTEGRITY */
PASS /* NOT ALLOWED TO BYPASS PASS PROT */
SYST /* SYSTEM TASK SO NOT TIMED */
AFF(NONE) /* NO PROCESSOR AFFINITY */
NOPREF /* NO PREFERRED STORAGE FRAMES */
```

Figure 1. PPT additional entries needed for the WebSphere MQ queue manager

The NOSWAP attribute used in earlier releases of WebSphere MQ is no longer necessary, because the WebSphere MQ queue manager controls swapping itself. However, if you have a heavily-loaded WebSphere MQ network and response time is critical, it might be advantageous to make the WebSphere MQ channel initiator non-swappable, by adding the following further PPT entry, at the risk of impacting the performance of the rest of your z/OS system:

```
PPT PGMNAME(CSQXJST) /* CSQ - MAKE WEBSPPHERE MQ MOVER NON-SWAPPABLE */
CANCEL /* CAN BE CANCELLED */
KEY(8) /* STORAGE PROTECTION KEY */
NOSWAP /* PROGRAM IS NON-SWAPPABLE */
```

Figure 2. PPT additional entries needed for the WebSphere MQ channel initiator

Issue the z/OS command SET SCH= for these changes to take effect.

## Task 5: Define the WebSphere MQ subsystem to z/OS

- Repeat this task for each WebSphere MQ queue manager.
- You do not need to perform this task when migrating from a previous version.

## Updating the subsystem name table

The subsystem name table of z/OS, which is taken initially from the SYS1.PARMLIB member IEFSSNss, contains the definitions of formally defined z/OS subsystems. To define each WebSphere MQ subsystem, you must add an entry to this table, either by changing the IEFSSNss member of SYS1.PARMLIB, or, preferably, by using the z/OS command SETSSI.

If you use the SETSSI command, the change takes effect immediately, and there is no need to IPL your system. You should update SYS1.PARMLIB as well, as described in “Task 12: Update SYS1.PARMLIB members” on page 21 so that the changes remain in effect after subsequent IPLs.

The SETSSI command to dynamically define a WebSphere MQ subsystem is:

```
SETSSI ADD,S=ssid,I=CSQ3INI,P='CSQ3EPX,cpf,scope'
```

The corresponding information in IEFSSNss can be specified in one of two ways:

- The keyword parameter form of the WebSphere MQ subsystem definition in IEFSSNss. This is the recommended method.

```
SUBSYS SUBNAME(ssid) INITRTN(CSQ3INI) INITPARM('CSQ3EPX,cpf,scope')
```

- The positional parameter form of the WebSphere MQ subsystem definition.

```
ssid,CSQ3INI,'CSQ3EPX,cpf,scope'
```

Do not mix the two forms in one IEFSSNss member. If different forms are required, use a separate IEFSSNss member for each type, adding the SSN operand of the new member to the IEASYSpp SYS1.PARMLIB member. To specify more than one SSN, use SSN=(aa,bb,...) in IEASYSpp.

In the examples above,

**ssid** The subsystem identifier. It can be up to four characters long. All characters must be alphanumeric (uppercase A through Z, 0 through 9), it must start with an alphabetic character. The queue manager will have the same name as the subsystem, therefore you can only use characters that are allowed for both z/OS subsystem names and WebSphere MQ object names.

**cpf** The command prefix string (see “Defining command prefix strings” on page 12 for information about CPFs).

**scope** The system scope, used if you are running in a z/OS sysplex (see “CPFs in a sysplex environment” on page 13 for information about system scope).

Figure 3 shows several examples of IEFSSNss statements.

```
CSQ1,CSQ3INI,'CSQ3EPX,+mqs1cpf,S'
CSQ2,CSQ3INI,'CSQ3EPX,+mqs2cpf,S'
CSQ3,CSQ3INI,'CSQ3EPX,++,S'
```

Figure 3. Sample IEFSSNss statements for defining subsystems

**Note:** Once you have created objects in a subsystem, you cannot change the subsystem name or use the page sets from one subsystem in another subsystem. To do either of these, you must unload all the objects and messages from one subsystem and reload them into another.

Table 4 gives a number of examples showing the associations of subsystem names and command prefix strings (CPFs), as defined by the statements in Figure 3.

Table 4. Subsystem name to CPF associations

WebSphere MQ subsystem name	CPF
CSQ1	+mqs1cpf
CSQ2	+mqs2cpf
CSQ3	++

**Note:** The ACTIVATE and DEACTIVATE functions of the z/OS command SETSSI are not supported by WebSphere MQ.

## Defining command prefix strings

You should adopt a system-wide convention for your CPFs for all subsystems to avoid conflicts. You should adhere to the following guidelines:

- Define a CPF as string of up to eight characters.
- Do not use a CPF that is already in use by any other subsystem, and avoid using the JES backspace character defined on your system as the first character of your string.
- Define your CPF using characters from the set of valid characters listed in Table 6 on page 13.
- Do not use a CPF that is an abbreviation for an already defined process or that might be confused with command syntax. For example, a CPF such as ‘D’ conflicts with z/OS commands such as DISPLAY. To avoid this happening, you should use one of the special characters (shown in Table 6 on page 13) as the first or only character in your CPF string.
- Do not define a CPF that is either a subset or a superset of an existing CPF. For an example, see Table 5.

Table 5. Example of CPF subset and superset rules

Subsystem name	CPF defined	Commands routed to...
MQA	!A	MQA
MQB	!B	MQB
MQC1	!C1	MQC1

Table 5. Example of CPF subset and superset rules (continued)

Subsystem name	CPF defined	Commands routed to...
MQC2	!C2	MQC2
MQB1	!B1	MQB

Commands intended for subsystem MQB1 (using CPF !B1) are routed to subsystem MQB because the CPF for this subsystem is !B, a subset of !B1. For example, if you entered the command:

```
!B1 START QMGR
```

subsystem MQB receives the command:

```
1 START QMGR
```

(which, in this case, it cannot deal with).

You can see which prefixes already exist by issuing the z/OS command DISPLAY OPDATA.

If you are running in a sysplex, z/OS diagnoses any conflicts of this type at the time of CPF registration (see “CPFs in a sysplex environment” for information about CPF registration).

Table 6 shows the characters that you can use when defining your CPF strings:

Table 6. Valid character set for CPF strings

Character set	Contents
Alphabetic	Uppercase A through Z, lowercase a through z
Numeric	0 through 9
National (see note)	@ \$ # (Characters that can be represented as hexadecimal values X'7C', X'5B', and X'7B', respectively)
Special	. / ( ) * & + - = < ! ! ; % _ ? : >
<p><b>Note:</b> The system recognizes the following hexadecimal representations of the national characters: @ as X'7C', \$ as X'5B', and # as X'7B'. In countries other than the U.S., the U.S. national characters represented on terminal keyboards might generate a different hexadecimal representation and cause an error. For example, in some countries the \$ character might generate an X'4A'.</p>	

## CPFs in a sysplex environment

If you are in a sysplex environment, WebSphere MQ registers your CPFs to enable you to enter a command from any console in the sysplex and route that command to the appropriate system for execution. The command responses are returned to the originating console.

### Defining the scope for sysplex operation:

Scope is used to determine the type of CPF registration performed by the WebSphere MQ subsystem when you are running WebSphere MQ in a sysplex environment.

Possible values for scope are as follows:

**M** System scope.

The CPF is registered with z/OS at system IPL time by WebSphere MQ and remains registered for the entire time that the z/OS system is active.

WebSphere MQ commands must be entered at a console connected to the z/OS image running the target subsystem, or you must use ROUTE commands to direct the command to that image.

You should use this option if you are not running in a sysplex.

**S** Sysplex started scope.

The CPF is registered with z/OS when the WebSphere MQ subsystem is started, and remains active until the WebSphere MQ subsystem terminates.

You must use ROUTE commands to direct the original START QMGR command to the target system, but all further WebSphere MQ commands can be entered at any console connected to the sysplex, and are routed to the target system automatically.

After WebSphere MQ termination, you must use the ROUTE commands to direct subsequent START commands to the target WebSphere MQ subsystem.

**X** Sysplex IPL scope.

The CPF is registered with z/OS at system IPL time by WebSphere MQ and remains registered for the entire time that the z/OS system is active.

WebSphere MQ commands can be entered at any console connected to the sysplex, and are routed to the image that is executing the target system automatically.

A WebSphere MQ subsystem with a CPF with scope of S can be defined on one or more z/OS images within a sysplex, so these images can share a single subsystem name table. However, you must ensure that the initial START command is issued on (or routed to) the z/OS image on which you want the WebSphere MQ subsystem to run. If you use this option, you can stop the WebSphere MQ subsystem and restart it on a different z/OS image within the sysplex without having to change the subsystem name table or re-IPL a z/OS system.

A WebSphere MQ subsystem with a CPF with scope of X can only be defined on one z/OS image within a sysplex. If you use this option, you must define a unique subsystem name table for each z/OS image requiring WebSphere MQ subsystems with CPFs of scope X.

If you want to use the z/OS automatic restart manager (ARM) to restart queue managers in different z/OS images automatically, every queue manager must be defined in each z/OS image on which that queue manager might be restarted. Every queue manager must be defined with a sysplex-wide, unique 4-character subsystem name with a CPF scope of S.

## **Task 6: Create procedures for the WebSphere MQ queue manager**

- *Repeat this task for each WebSphere MQ queue manager.*
- *You might need to perform this task when migrating from a previous version. For details, see [WebSphere MQ Migration Information](#).*



For each WebSphere MQ subsystem defined in the subsystem name table, create a cataloged procedure in a procedure library for starting the queue manager. The IBM-supplied procedure library is called SYS1.PROCLIB, but your installation might use its own naming convention.

The name of the queue manager started task procedure is formed by concatenating the subsystem name with the characters MSTR. For example, subsystem CSQ1 has the procedure name CSQ1MSTR. You need one procedure for each subsystem you define.

We recommend that a subsystem called CSQ1MSTR is created initially for installation verification and testing purposes.

Two sample started task procedures are provided in thlqual.SCSQPROC. Member CSQ4MSTR uses one page set for each class of message, member CSQ4MSRR uses multiple page sets for the major classes of message. Copy one of these procedures to member xxxxMSTR (where xxxx is the name of your WebSphere MQ subsystem) of your SYS1.PROCLIB or, if you are not using SYS1.PROCLIB, your procedure library. Copy the sample procedure to a member in your procedure library for each WebSphere MQ subsystem that you define.

When you have copied the members, you can tailor them to the requirements of each subsystem, using the instructions in the member. You can also use symbolic parameters in the JCL to allow the procedure to be modified when it is started. This is described with the start options in the WebSphere MQ for z/OS System Administration Guide. If you have several WebSphere MQ subsystems, you might find it advantageous to use JCL include groups for the common parts of the procedure, to simplify future maintenance.

You must concatenate thlqual.SCSQANLx (where x is the language letter for your national language) before thlqual.SCSQAUTH in the STEPLIB DD statement.

If you are using queue-sharing groups, the STEPLIB concatenation must include the DB2 runtime target library SDSNLOAD, and it must be APF-authorized. This library is only required in the STEPLIB concatenation if it is not accessible through the linklist or LPA.

Before you start the queue manager, set up WebSphere MQ data set and system security by:

- Authorizing the queue manager started task procedure to run under your external security manager.
- Authorizing access to the queue manager data sets.

For details about how to do this, see “Security installation tasks” on page 184.

You can add the exit library (CSQXLIB) to this procedure later if you want to use queue manager exits. You will need access to the Language Environment (LE) runtime library SCEERUN to do this; if it is not in your link list (SYS1.PARMLIB(LNKLSTkk)), concatenate it in the STEPLIB DD statement. You also need to stop and restart your queue manager.

## **Task 7: Create procedures for the channel initiator**

For each WebSphere MQ subsystem, tailor a copy of CSQ4CHIN. Depending on what other products you are using, you might need to allow access to other data sets.

- Repeat this task for each WebSphere MQ queue manager.
- You might need to perform this task when migrating from a previous version. For details, see *WebSphere MQ Migration Information*.

You need to create a channel-initiator started task procedure for each WebSphere MQ subsystem that is going to use distributed queuing.

To do this:

1. Copy the sample started task procedure `thlqual.SCSQPROC(CSQ4CHIN)` to your procedure library. Name the procedure `xxxxCHIN`, where `xxxx` is the name of your WebSphere MQ subsystem (for example, `CSQ1CHIN` would be the channel initiator started task procedure for queue manager `CSQ1`).
2. Make a copy for each WebSphere MQ subsystem that you are going to use.
3. Tailor the procedures to your requirements using the instructions in the sample procedure `CSQ4CHIN`. You can also use symbolic parameters in the JCL to allow the procedure to be modified when it is started. This is described with the start options in the *WebSphere MQ for z/OS System Administration Guide*.

Concatenate the library containing your national language feature (`thlqual.SCSQANLx` where `x` is the letter for your language) before `thlqual.SCSQAUTH` in the `STEPLIB DD` statement.

Concatenate the distributed queuing library `thlqual.SCSQMVR1`.

Access to the LE runtime library `SCEERUN` is required; if it is not in your link list (`SYS1.PARMLIB(LNKLSTkk)`), concatenate it in the `STEPLIB DD` statement.

4. Authorize the procedures to run under your external security manager.

You can add the exit library (`CSQXLIB`) to this procedure later if you want to use channel exits. You need to stop and restart your channel initiator to do this.

If you are using SSL, access to the system Secure Sockets Layer (SSL) runtime library is required. This library is called `SIEALNKE`. The library must be APF authorized.

If you are using TCP/IP, the channel initiator address space must be able to access the `TCPIP.DATA` data set that contains TCP/IP system parameters. The ways that the data set has to be set up depends on which TCP/IP product and interface you are using. They include:

- Environment variable, `RESOLVER_CONFIG`
- HFS file, `/etc/resolv.conf`
- `//SYSTCPD DD` statement
- `//SYSTCPDD DD` statement
- `jobname/userid.TCPIP.DATA`
- `SYS1.TCPPARMS(TCPDATA)`
- `zapname.TCPIP.DATA`

Some of these affect your started-task procedure JCL. For more information, see the following:

- *TCP/IP OpenEdition: Planning and Release Guide*
- *OS/390 OpenEdition: Planning*

- Your Unicenter TCPAccess Communication Server documentation

## Task 8: Define the WebSphere MQ subsystem to a z/OS WLM service class

- Repeat this task for each WebSphere MQ queue manager.
- You do not need to perform this task when migrating from a previous version.

To give WebSphere MQ appropriate performance priority in the z/OS system, you must assign the queue manager and channel initiator address spaces to an appropriate z/OS workload management (WLM) service class. If you do not do this explicitly then inappropriate defaults might apply.

Use the ISPF dialog supplied with WLM to

- extract the z/OS WLM policy definition from the WLM couple data set
- update this policy definition by adding queue manager and channel initiator started task procedure names to the chosen service class
- install the changed policy on the WLM couple data set

Then activate this policy using the z/OS command

```
V WLM,POLICY=poli c yname,REFRESH
```

See WebSphere MQ for z/OS Concepts and Planning Guide for more information on setting performance options.

## Task 9: Set up the DB2 environment

- Repeat this task for each DB2 data-sharing group.
- You might need to perform this task when migrating from a previous version. For details, see *Migrating from a previous version WebSphere MQ Migration Information*.
- Omit this task if you are not using queue-sharing groups.  
If you later want to use queue-sharing groups, see *WebSphere MQ Migration Information*.

You need to establish an environment in which WebSphere MQ can access and execute the DB2 plans that are used for queue-sharing groups.

The following steps must be performed for each new DB2 data-sharing group. All the sample JCL is in `thlqual.SCSQPROC`. (See *WebSphere MQ Migration Information* for information on how to migrate an existing data-sharing group.)

1. Customize and execute sample JCL CSQ45CSG to create the storage group that is to be used for the WebSphere MQ database, tablespaces, and tables.
2. Customize and execute sample JCL CSQ45CDB to create the database to be used by all queue managers that will connect to this DB2 data-sharing group.

3. Customize and execute sample JCL CSQ45CTS to create the tablespaces that will contain the queue manager and channel initiator tables used for queue-sharing groups (to be created in step 1).
4. Customize and execute sample JCL CSQ45CTB to create the 12 DB2 tables and associated indexes. Do not change any of the row names or attributes.
5. Customize and execute sample JCL CSQ45BPL to bind the DB2 plans for the queue manager, utilities, and channel initiator.
6. Customize and execute sample JCL CSQ45GEX to grant execute authority to the respective plans for the user IDs that will be used by the queue manager, utilities, and channel initiator. The user IDs for the queue manager and channel initiator are the user IDs under which their started task procedures run. The user IDs for the utilities are the user IDs under which the batch jobs can be submitted. The names of the appropriate plans are:

User	Plans
Queue manager	CSQ5A700, CSQ5C700, CSQ5D700, CSQ5K700, CSQ5L700, CSQ5M700, CSQ5P700, CSQ5R700, CSQ5S700, CSQ5T700, CSQ5U700, CSQ5W700
SDEFS function of the CSQUTIL batch utility	CSQ52700
CSQ5PQSG batch utility	CSQ5B600
CSQUZAP service utility	CSQ5Z600

In the event of a failure during DB2 setup, the following jobs can be customized and executed:

- CSQ45DTB to drop the tables and indexes.
- CSQ45DTS to drop the tablespaces.
- CSQ45DDB to drop the database.
- CSQ45DSG to drop the storage group.

**Note:** If these jobs fail because of a DB2 locking problem it is probably due to contention for a DB2 resource, especially if the system is being heavily used. Resubmit the jobs later. It is preferable to run these jobs when the system is lightly used or quiesced.

See the *DB2 for OS/390 Administration Guide* for more information about setting up DB2.

See the *WebSphere MQ for z/OS Concepts and Planning Guide* for information about DB2 table sizes.

## Task 10: Set up the Coupling Facility

- Repeat this task for each queue-sharing group.
- You might need to perform this task when migrating from a previous version. For details, see *WebSphere MQ Migration Information*.
- Omit this task if you are not using queue-sharing groups.  
If you later want to use queue-sharing groups, see *WebSphere MQ Migration Information*.

You need to define the Coupling Facility structures used by the queue managers in the queue-sharing group in the Coupling Facility Resource Management (CFRM) policy data set, using IXCMIAPU.

All the structures for the queue-sharing group start with the name of the queue-sharing group. You must have:

- An administrative structure called *qsg-name*CSQ\_ADMIN. This structure is used by WebSphere MQ itself and does not contain any user data.
- One or more structures used to hold messages for shared queues. These can have any name you choose up to 16 characters long.
  - The first four characters must be the queue-sharing group name. (If the queue-sharing group name is less than four characters long, it must be padded to four characters with @ symbols.)
  - The fifth character must be alphabetic and subsequent characters can be alphabetic or numeric. This part of the name (without the queue-sharing group name) is what you specify for the CFSTRUCT name when you define a shared queue, or a CF structure object.

You can use only alphabetic and numeric characters in the names of structures used to hold messages for shared queues, you cannot use any other characters (for example, the \_ character, which is used in the name of the administrative structure).

Sample control statements for IXCMIAPU are in data set `thlqual.SCSQPROC(CSQ4CFRM)`. Customize these and add them to your IXCMIAPU job for the Coupling Facility and run it.

When you have defined your structures successfully, activate the CFRM policy that is being used. To do this, issue the following z/OS command:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=policy-name
```

See the *WebSphere MQ for z/OS Concepts and Planning Guide* for information about planning CF structures and their sizes.

## Task 11: Implement your ESM security controls

- Repeat this task for each WebSphere MQ queue manager or queue-sharing group.
- You might need to perform this task when migrating from a previous version. For details, see *WebSphere MQ Migration Information*.

You must now consider how you are going to implement any security controls for WebSphere MQ.

If you use RACF® as your external security manager, see Chapter 7, “Setting up security,” on page 123, which describes how to implement these security controls.

If you are using queue-sharing groups, ensure that the user IDs associated with the queue manager, channel initiator, and the utilities (as specified in step 6 on page 18) have authority to establish a RRSF connection to each DB2 subsystem with which you want to establish a connection. The RACF profile to which the user ID requires READ access is *DB2ssid.RRSF* in the DSNR resource class.

If you are using the channel initiator, you must also do the following:

- If your subsystem has connection security active, define a connection security profile *ssid.CHIN* to your external security manager (see “Connection security profiles for the channel initiator” on page 135 for information about this).
- If you are using the Secure Sockets Layer (SSL) or a sockets interface, ensure that the user ID under whose authority the channel initiator is running is configured to use OpenEdition, as described in the *OS/390 OpenEdition Planning* manual.
- If you are using SSL, ensure that the user ID under whose authority the channel initiator is running is configured to access the key ring specified in the *SSLKEYR* parameter of the *ALTER QMGR* command.
- If you are using shared listeners and have set *DNSWLM=YES*, ensure that the user ID associated with the channel initiator has READ access to the *BPX.WLMSEVER* profile in the FACILITY class. If it does not have the required access, message *MSGCSQX473E* is issued to report the failure and the listener goes into retry state.

Those queue managers that will access the Coupling Facility list structures require the appropriate security access. The RACF class is FACILITY. The queue manager user ID requires ALTER access to the *IXLSTR.structure-name* profile.

If you are using RACF, provided you use the RACF STARTED class, you do not need to IPL your system (see “RACF authorization of started-task procedures” on page 185).

## Task 12: Update SYS1.PARMLIB members

- You need to perform this task once for each z/OS system where you want to run WebSphere MQ.
- If you are using queue-sharing groups, you must ensure that the settings for WebSphere MQ are identical on each z/OS system in the sysplex.
- You might need to perform this task when migrating from a previous version. For details, see *WebSphere MQ Migration Information*.

To ensure that your changes remain in effect after an IPL, update members of SYS1.PARMLIB as follows:

1. Update member IEFSSNss as described in “Task 5: Define the WebSphere MQ subsystem to z/OS” on page 10.
2. Change IEASYSpp so that the following are used when the system is IPLed:
  - the PROGxx or IEAAPFaa members used in “Task 2: APF authorize the WebSphere MQ load libraries” on page 7
  - the LNKLSTkk and LPALSTmm members used in “Task 3: Update the z/OS link list and LPA” on page 8
  - the SCHEDxx member used in “Task 4: Update the z/OS program properties table” on page 10
  - the IEFSSNss member used in “Task 5: Define the WebSphere MQ subsystem to z/OS” on page 10

## Task 13: Customize the initialization input data sets

- Repeat this task for each WebSphere MQ queue manager.
- You need to perform this task when migrating from a previous version. For details, see *WebSphere MQ Migration Information*.

Each WebSphere MQ queue manager gets its initial definitions from a series of commands contained in the WebSphere MQ *initialization input data sets*. These data sets are referenced by the DDnames CSQINP1 and CSQINP2 defined in the queue manager started task procedure.

Responses to these commands are written to the initialization output data sets referenced by the DDnames CSQOUT1 and CSQOUT2.

Sample initialization input data sets are supplied with WebSphere MQ, see the *WebSphere MQ for z/OS Concepts and Planning Guide* for information about them.

To preserve the originals, you should make working copies of each sample. Then you can tailor the commands in these working copies to suit your system requirements.

If you use more than one WebSphere MQ subsystem, you are recommended to include the subsystem name in the high-level qualifier of the initialization input data set name. This allows you to identify the WebSphere MQ subsystem associated with each data set more easily.

## Initialization data set formats

The initialization input data sets can be partitioned data set (PDS) members or sequential data sets. They can be a concatenated series of data sets. Define them with a record length of 80 bytes, where:

- Only columns 1 through 72 are significant. Columns 73 through 80 are ignored.
- Records with an asterisk (\*) in column 1 are interpreted as comments and are ignored.
- Blank records are ignored.
- Each command must start on a new record.
- A trailing - means continue from column 1 of the next record.
- A trailing + means continue from the first non-blank column of the next record.
- The maximum number of characters permitted in a command is 32 762.

The initialization output data sets are sequential data sets, with a record length of 125, a record format of VBA, and a block size of 629.

## Using the CSQINP1 sample

Data set thlqual.SCSQPROC holds two members which contain definitions of buffer pools, page set to buffer pool associations, and an ALTER SECURITY command. Member CSQ4INP1 uses one page set for each class of message, member CSQ4INPR uses multiple page sets for the major classes of message. The appropriate sample should be included in the CSQINP1 concatenation of your queue manager started task procedure.

### Note:

1. WebSphere MQ supports up to 16 buffer pools (zero through 15). The DEFINE BUFFPOOL command can only be issued from a CSQINP1 initialization data set. The definitions in the sample specify four buffer pools.
2. Each page set used by the queue manager must be defined in the CSQINP1 initialization data set by using the DEFINE PSID command. The page set definition associates a buffer pool ID with a page set. If no buffer pool is specified, buffer pool zero is used by default.  
Page set zero (00) must be defined. It contains all the object definitions. You can define up to 100 page sets for each queue manager.
3. The ALTER SECURITY command can be used to alter the security attributes TIMEOUT and INTERVAL. In CSQ4INP1, the default values are defined as 54 for TIMEOUT and 12 for INTERVAL.

See the WebSphere MQ for z/OS Concepts and Planning Guide for information about organizing buffer pools and page sets.

The buffer pool and page set definitions must be defined each time the queue manager is started. If you change them dynamically while the queue manager is running you should also update the CSQINP1 definitions so that the changes are retained when the queue manager is restarted.



## Using the CSQINP2 samples

This table lists the members of thlqual.SCSQPROC that can be included in the CSQINP2 concatenation of your queue manager started task procedure, with a description of their function.

Table 7. Members of thlqual.SCSQPROC

Member name	Description
CSQ4INSG	System object definitions.
CSQ4INSX	System object definitions.
CSQ4INSS	Customize and include this member if you are using queue-sharing groups.
CSQ4INSJ	Customize and include this member if you are using publish/subscribe using JMS.
CSQ4INSR	You might need to customize and include this member if you are using WebSphere Application Server or a Broker.
The following members contain definitions that you can customize for your own objects	
CSQ4INYC	Clustering definitions.
CSQ4INYD	Distributed queueing definitions.
CSQ4INYG	General definitions.
CSQ4INYR	Storage class definitions, using multiple page sets for the major classes of message.
CSQ4INYS	Storage class definitions, using one page set for each class of message.

You need to define objects once only, not each time that you start a queue manager, so it is not necessary to include these definitions in CSQINP2 every time. If you do include them every time, you are attempting to define objects that already exist, and you will get messages similar to the following:

```
CSQM095I +CSQ1 CSQMAQLC QLOCAL(SYSTEM.DEFAULT.LOCAL.QUEUE) ALREADY EXISTS
CSQM090E +CSQ1 CSQMAQLC FAILURE REASON CODE X'00D44003'
CSQ9023E +CSQ1 CSQMAQLC ' DEFINE QLOCAL' ABNORMAL COMPLETION
```

The objects are not damaged by this failure. If you want to leave the SYSTEM definitions data set in the CSQINP2 concatenation, you can avoid the failure messages by specifying the REPLACE attribute against each object.

## Using the CSQINPX sample

Sample thlqual.SCSQPROC(CSQ4INPX) contains a set of commands that you might want to execute each time the channel initiator starts. These are typically channel-related commands such as START LISTENER, which are required every time the channel initiator starts, rather than whenever the queue manager starts, and which are not allowed in the input data sets CSQINP1 or CSQINP2. You must customize this sample before use; you can then include it in the CSQINPX data set for the channel initiator.

The WebSphere MQ commands contained in the data set are executed at the end of channel initiator initialization, and output is written to the data set specified by the CSQOUTX DD statement. The output is similar to that produced by the

COMMAND function of the WebSphere MQ utility program (CSQUTIL). See the WebSphere MQ for z/OS System Administration Guide for information about the WebSphere MQ utility program.

You can specify any of the WebSphere MQ commands that can be issued from CSQUTIL, not only the channel commands. You can enter commands from other sources while CSQINPX is being processed. All commands are issued in sequence, regardless of the success of the previous command.

To specify a command response time, you can use the pseudo-command COMMAND as the first command in the data set. This takes a single optional keyword RESPTIME(*nnn*), where *nnn* is the time, in seconds, to wait for a response to each command. This is in the range 5 through 999; the default is 30.

If WebSphere MQ detects that the responses to four commands have taken too long, processing of CSQINPX is stopped and no further commands are issued. The channel initiator is not stopped, but message CSQU052E is written to the CSQOUTX data set, and message CSQU013E is sent to the console.

When WebSphere MQ has completed processing of CSQINPX successfully, message CSQU012I is sent to the console.

## Task 14: Create the bootstrap and log data sets

- Repeat this task for each WebSphere MQ queue manager
- You do not need to perform this task when migrating from a previous version.

Use the supplied program CSQJU003 to prepare the bootstrap data sets (BSDSs) and log data sets. You must run this job once for each WebSphere MQ queue manager you use. The sample JCL and Access Method Services (AMS) control statements to run CSQJU003 to create a single or dual logging environment are held in thlqual.SCSQPROC(CSQ4BSDS). Customize and run this job to create your BSDSs and logs and to preformat the logs.

The started task procedure, CSQ4MSTR, described in “Task 6: Create procedures for the WebSphere MQ queue manager” on page 14, refers to BSDSs in statements of the form:

```
//BSDS1 DD DSN=++HLQ++.BSDS01,DISP=SHR
//BSDS2 DD DSN=++HLQ++.BSDS02,DISP=SHR
```

The log data sets are referred to by the BSDSs.

### Note:

1. The BLKSIZE must be specified on the SYSPRINT DD statement in the CSQTLOG step. The BLKSIZE must be 629.
2. To help identify bootstrap data sets and log data sets from different queue managers, include the subsystem name in the high level qualifier of these data sets.

3. If you are using queue-sharing groups, you must define the bootstrap and log data sets with SHAREOPTIONS(2 3).

See WebSphere MQ for z/OS Concepts and Planning Guide for information about planning bootstrap and log data sets and their sizes.

## Task 15: Define your page sets

- Repeat this task for each WebSphere MQ queue manager.
- You do not need to perform this task when migrating from a previous version.

You must define separate page sets for each WebSphere MQ queue manager. `thlqual.SCSQPROC(CSQ4PAGE)` and `thlqual.SCSQPROC(CSQ4PAGR)` contain JCL and AMS control statements to define and format page sets. Member `CSQ4PAGE` uses one page set for each class of message, member `CSQ4PAGR` uses multiple page sets for the major classes of message. The JCL runs the supplied utility program `CSQUTIL`. Review the samples and customize them for the number of page sets you want and the sizes to use. See the WebSphere MQ for z/OS Concepts and Planning Guide for information about page sets and how to calculate suitable sizes.

The started task procedure `CSQ4MSTR` described in “Task 6: Create procedures for the WebSphere MQ queue manager” on page 14 refers to the page sets, in a statement of the form:

```
//CSQP00nn DD DISP=OLD,DSN=xxxxxxxx
```

where `nn` is the page set number between 00 and 99, and `xxxxxxxx` is the data set that you define.

### Note:

1. If you intend to use the dynamic page set expansion feature, ensure that secondary extents are defined for each page set. `thlqual.SCSQPROC(CSQ4PAGE)` shows how to do this.
2. To help identify page sets from different queue managers, include the subsystem name in the high level qualifier of the data set associated with each page set.
3. If you intend to allow the `FORCE` option to be used with the `FORMAT` function of the utility program `CSQUTIL`, you must add the `REUSE` attribute on the `AMS DEFINE CLUSTER` statement. This is described in the WebSphere MQ for z/OS System Administration Guide.
4. If your page sets are to be larger than 4 GB you must use the Storage Management System (SMS) `EXTENDED ADDRESSABILITY` function.

## Task 16: Add the WebSphere MQ entries to the DB2 data-sharing group

- Repeat this task for each WebSphere MQ queue-sharing group and each queue manager.
- You might need to perform this task when migrating from a previous version. For details, see *WebSphere MQ Migration Information*.
- Omit this task if you are not using queue-sharing groups.  
If you later want to use queue-sharing groups, see *WebSphere MQ Migration Information*.

If you are using queue-sharing groups, run the CSQ5PQSG utility to add queue-sharing group and queue manager entries to the WebSphere MQ tables in the DB2 data-sharing group. Run the utility for each queue-sharing group and each queue manager that is to be a member of a queue-sharing group. (CSQ5PQSG is described in the *WebSphere MQ for z/OS System Administration Guide*.)

You should perform the following actions in the specified order:

1. Add a queue-sharing group entry into the WebSphere MQ DB2 tables using the ADD QSG function of the CSQ5PQSG program. A sample is provided in thlqual.SCSQPROC(CSQ45AQS).

Perform this function once for each queue-sharing group that is defined in the DB2 data-sharing group. The queue-sharing group must exist before adding any queue manager entries that reference the queue-sharing group.

2. Add a queue manager entry into the WebSphere MQ DB2 tables using the ADD QMGR function of the CSQ5PQSG program. A sample is provided in thlqual.SCSQPROC(CSQ45AQM).

Perform this function for each queue manager that is to be a member of the queue-sharing group.

**Note:**

- a. A queue manager can only be a member of one queue-sharing group.
- b. You must have RRS running to be able to use queue-sharing groups.

## Task 17: Tailor your system parameter module

- Repeat this task for each WebSphere MQ queue manager, as required.
- You need to perform this task when migrating from a previous version. For details, see *WebSphere MQ Migration Information*.

The WebSphere MQ system parameter module controls the logging, archiving, tracing, and connection environments that WebSphere MQ uses in its operation. The system parameter module has three macros as follows:

Macro name	Purpose
CSQ6SYSP	Specifies the connection and tracing parameters, see topic "Using CSQ6SYSP" on page 28
CSQ6LOGP	Controls log initialization, see topic "Using CSQ6LOGP" on page 35
CSQ6ARVP	Controls archive initialization, see topic "Using CSQ6ARVP" on page 38

WebSphere MQ supplies a default system parameter module, CSQZPARM, which is invoked automatically if you issue the START QMGR command (without a PARM parameter) to start an instance of WebSphere MQ. CSQZPARM is in the APF-authorized library thlqual.SCSQAUTH also supplied with WebSphere MQ. The values of these parameters are displayed as a series of messages when you start WebSphere MQ.

See the WebSphere MQ Script (MQSC) Command Reference manual for more information about the START QMGR command and the WebSphere MQ for z/OS System Administration Guide for more information about how this command is used.

## Creating your own system parameter module

If CSQZPARM does not contain the system parameters you want, you can create your own system parameter module using the sample JCL provided in thlqual.SCSQPROC(CSQ4ZPRM).

To create your own system parameter module:

1. Make a working copy of the JCL sample.
2. Edit the parameters for each macro in the copy as required. If you remove any parameters from the macro calls, the default values are automatically picked up at run time.
3. Replace the placeholder ++NAME++ with the name that the load module is to take (this can be CSQZPARM).
4. If your assembler is not high level assembler, change the JCL as required by your assembler.
5. Run the JCL to assemble and link-edit the tailored versions of the system parameter macros to produce a load module. This is the new system parameter module with the name that you have specified.
6. Put the load module produced in an APF-authorized user library.
7. Include this library in the WebSphere MQ queue manager started task procedure STEPLIB. This library name must come before the library thlqual.SCSQAUTH in STEPLIB.
8. Invoke the new system parameter module when you start the queue manager. For example, if the new module is named NEWMODS, issue the command:

```
START QMGR PARM(NEWMODS)
```

**Note:** If you choose to name your module CSQZPARM, you do not need to specify the PARM parameter on the START QMGR command.

## Fine tuning a system parameter module

WebSphere MQ also supplies a set of three assembler source modules, which can be used to fine tune an existing system parameter module. These modules are in library thlqual.SCSQASMS. Typically, you use these modules in a test environment to change the default parameters in the system parameter macros. Each source module calls a different system parameter macro:

This assembler source module...	Calls this macro...
CSQFSYSP	CSQ6SYSP (connection and tracing parameters)
CSQJLOGP	CSQ6LOGP (log initialization)
CSQJARVP	CSQ6ARVP (archive initialization)

This is how you use these modules:

1. Make working copies of each assembler source module in a user assembler library.
2. Edit your copies by adding or altering the values of any parameters as required.
3. Assemble your copies of any edited modules to create object modules in a user object library.
4. Link-edit these object code modules with an existing system parameter module to produce a load module that is the new system parameter module.
5. Ensure that new system parameter module is a member of a user authorized library.
6. Include this library in the queue manager started task procedure STEPLIB. This library must come before the library thlqual.SCSQAUTH in STEPLIB.
7. Invoke the new system parameter module by issuing a START QMGR command, specifying the new module name in the PARM parameter, as before.

## Altering system parameters

You can alter some system parameters while a queue manager is running. See the SET SYSTEM, SET LOG and SET ARCHIVE commands in the WebSphere MQ Script (MQSC) Command Reference.

Put the SET commands in your initialization input data sets so that they take effect every time you start the queue manager.

## Using CSQ6SYSP

Use CSQ6SYSP to set system parameters.

The default parameters for CSQ6SYSP, and whether you can alter each parameter using the SET SYSTEM command, are shown in Table 8. If you want to change any of these values, see the detailed descriptions of the parameters.

*Table 8. Default values of CSQ6SYSP parameters*

Parameter	Description	Default value	SET command
CLCACHE	Specifies the type of cluster cache to use.	STATIC	–
CMDUSER	The default user ID for command security checks.	CSQOPR	–
EXITLIM	Time (in seconds) for which queue-manager exits can run during each invocation.	30	–
EXITTCB	How many started server tasks to use to run queue manager exits.	8	–
IDBACK	Maximum number of background connections to a single queue manager using batch connections.	20	X

Table 8. Default values of CSQ6SYSP parameters (continued)

Parameter	Description	Default value	SET command
IDFORE	Maximum number of foreground connections to a single queue manager using batch connections.	100	X
LOGLOAD	Number of log records written by WebSphere MQ between the start of one checkpoint and the next.	500 000	X
OTMACON	OTMA connection parameters.	See parameter description	–
QINDXBLD	Determines whether queue manager restart waits until all indexes are rebuilt, or completes before all indexes are rebuilt.	WAIT	–
QMCCSID	Coded character set identifier for the queue manager.	zero	–
QSGDATA	Queue-sharing group parameters.	See parameter description	–
RESAUDIT	RESLEVEL auditing parameter.	YES	–
ROUTCDE	Message routing code assigned to messages not solicited from a specific console.	1	–
SERVICE	Reserved for use by IBM®.	0	X
SMFACCT	Specifies whether SMF accounting data is to be collected when the queue manager is started.	NO	–
SMFSTAT	Specifies whether SMF statistics are to be collected when the queue manager is started.	NO	–
STATIME	Default time, in minutes, between each gathering of statistics.	30	X
TRACSTR	Specifies whether tracing is to be started automatically.	NO	–
TRACTBL	Size of trace table, in 4 KB blocks, to be used by the global trace facility.	99 (396 KB)	X
WLMTIME	Time between scanning the queue index for WLM-managed queues.	30	–
WLMTIMU	Units (minutes or seconds) for WLMTIME.	MINS	–

### CLCACHE

Specifies the type of cluster cache to use. See WebSphere MQ Queue Manager Clusters for more information.

### STATIC

When the cluster cache is static, its size is fixed at queue manager start-up, enough for the current amount of cluster information plus some space for expansion. The size cannot increase while the queue manager is active. This is the default.

### DYNAMIC

When the cluster cache is dynamic, the initial size allocated at queue manager startup can be increased automatically if required while the queue manager is active.

### **CMDUSER**

Specifies the default user ID used for command security checks. This user ID must be defined to the ESM (for example, RACF). Specify a name of 1 through 8 alphanumeric characters. The first character must be alphabetic.

The default is CSQOPR.

### **EXITLIM**

Specifies the time, in seconds, allowed for each invocation of the queue manager exits. (This parameter has no effect on channel exits.)

Specify a value in the range 5 through 9999.

The default is 30. The queue manager polls exits that are running every 30 seconds. On each poll, any that have been running for more than the time specified by EXITLIM are forcibly terminated.

### **EXITTCB**

Specifies the number of started server tasks to use to run exits in the queue manager. (This parameter has no effect on channel exits.) You must specify a number at least as high as the maximum number of exits (other than channel exits) that the queue manager might have to run, else it will fail with a 6c6 abend.

Specify a value in the range zero through 99. A value of zero means that no exits can be run.

The default is 8.

### **IDBACK**

Specifies the maximum number of background batch connections to the queue manager. The value of IDBACK is related to those of IDFORE

Specify a number in the range 1 through 32 767.

The default is 20.

### **IDFORE**

Specifies the maximum number of foreground batch connections to the queue manager.

The value of IDFORE is related to those of IDBACK

The number of TSO connections might be greater than the number of concurrent TSO users if, for example, users split their ISPF screens.

Specify a number in the range zero through 32 767.

The default is 100.

### **LOGLOAD**

Specifies the number of log records that WebSphere MQ writes between the start of one checkpoint and the next. WebSphere MQ starts a new checkpoint after the number of records that you specify has been written.

Specify a value in the range 200 through 16 000 000.

The default is 500 000.

The greater the value, the better the performance of WebSphere MQ; however, restart takes longer if the parameter is set to a large value.

Suggested settings:

<b>Test system</b>	10 000
--------------------	--------



**Production system** 500 000

In a production system, the supplied default value might result in a checkpoint frequency that is too high.

The value of LOGLOAD determines the frequency of queue manager checkpoints. Too large a value means that a large amount of data is written to the log between checkpoints, resulting in an increased queue manager forward recovery restart time following a failure. Too small a value causes checkpoints to occur too frequently during peak load, adversely affecting response times and CPU usage.

An initial value of 500 000 is suggested for LOGLOAD. (For example, this adds approximately 90 seconds to the restart time after a failure if using RAMAC<sup>®</sup> Virtual Array 2 Turbo 82 (RVA2-T82) DASD.) For a 1 KB persistent message rate of 100 messages a second (that is, 100 MQPUTs with commit and 100 MQGETs with commit) the interval between checkpoints is approximately 5 minutes.

**Note:** This is intended as a guideline only and the optimum value for this parameter is dependent on the characteristics of the individual system.

## OTMACON

OTMA parameters. This keyword takes five positional parameters::

**OTMACON = (Group, Member, Druexit, Age, Tpipepfx)**

### Group

This is the name of the XCF group to which this particular instance of WebSphere MQ belongs.

It can be 1 through 8 characters long and must be entered in uppercase characters.

The default is blanks, which indicates that WebSphere MQ must not attempt to join an XCF group.

### Member

This is the member name of this particular instance of WebSphere MQ within the XCF group.

It can be 1 through 16 characters long and must be entered in uppercase characters.

The default is the 4-character queue manager name.

### Druexit

This specifies the name of the OTMA destination resolution user exit to be run by IMS.

It can be 1 through 8 characters long.

The default is DFSYDRU0.

This parameter is optional; it is required if WebSphere MQ is to receive messages from an IMS application that was not started by WebSphere MQ. The name must correspond to the destination resolution user exit coded in the IMS system. For more information see Chapter 9, "Using OTMA exits in IMS," on page 223.

### Age

This represents the length of time, in seconds, that a user ID from WebSphere MQ is considered previously verified by IMS.

It can be in the range zero through 2 147 483 647.

The default is 2 147 483 647.

You are recommended to set this parameter in conjunction with the interval parameter of the ALTER SECURITY command to maintain consistency of security cache settings across the mainframe.

### **Tpipepfx**

This represents the prefix to be used for Tpipe names.

It comprises three characters; the first character is in the range A through Z, subsequent characters are A through Z or 0 through 9. The default is CSQ.

This is used each time WebSphere MQ creates a Tpipe; the rest of the name is assigned by WebSphere MQ. You cannot set the full Tpipe name for any Tpipe created by WebSphere MQ.

### **QINDXBLD**

Determines whether queue manager restart waits until all queue indexes are rebuilt, or completes before all indexes are rebuilt.

#### **WAIT**

Queue manager restart waits for all queue index builds to be completed. This means that no applications are delayed during normal WebSphere MQ API processing while the index is created, as all indexes are created before any applications can connect to the queue manager.

This is the default.

#### **NOWAIT**

The queue manager can restart before all queue index building is completed.

### **QMCCSID**

Specifies the default coded character set identifier that the queue manager (and therefore distributed queuing) is to use.

Specify a value in the range zero through 65 535.

Zero, which is the default value, means use the CCSID currently set or, if none is set, use CCSID 500. This means that if you have explicitly set the CCSID to any non-zero value, you cannot reset it by setting QMCCSID to zero; you must now use the correct non-zero CCSID. If QMCCSID is zero, you can check what CCSID is actually in use by issuing the command DISPLAY QMGR CCSID.

### **QSGDATA**

Queue-sharing group data. This keyword takes five positional parameters:

**QSGDATA=(Qsgname, Dsgname, Db2name, Db2serv, Db2b1ob)**

#### **Qsgname**

This is the name of the queue-sharing group to which the queue manager belongs.

It can be 1 through 4 characters long. Acceptable characters are uppercase A-Z, 0-9, \$, #, and @. It must not start with a numeric. For implementation reasons, names of less than four characters are padded internally with @ symbols, so do not use a name ending in @.

The default is blanks, which indicates that the queue manager is not a member of any queue-sharing group.

**Dsgname**

This is the name of the DB2 data-sharing group to which the queue manager is to connect.

It can be 1 through 8 characters long and must be entered in uppercase characters.

The default is blanks, which indicates that you are not using queue-sharing groups.

**Db2name**

This is the name of the DB2 subsystem or group attachment to which the queue manager is to connect.

It can be 1 through 4 characters long and must be entered in uppercase characters.

The default is blanks, which indicates that you are not using queue-sharing groups.

**Note:** The DB2 subsystem (or group attachment) must be in the DB2 data-sharing group specified in the Dsgname, and all queue managers must specify the same DB2 data-sharing group.

**Db2serv**

This is the number of server tasks used for accessing DB2.

It can be in the range 4 through 10.

The default is 4.

**Db2blob**

This is the number of DB2 tasks used for accessing Binary Large Objects (BLOBs).

It can be in the range 4 through 10.

The default is 4.

If you specify one of the name parameters (that is, not the Db2serv parameter), you must enter values for the other names, otherwise WebSphere MQ fails.

**RESAUDIT**

Specifies whether RACF audit records are written for RESLEVEL security checks performed during connection processing.

Specify one of:

**NO** RESLEVEL auditing is not performed.

**YES** RESLEVEL auditing is performed.

The default is YES.

**ROUTCDE**

Specifies the default z/OS message routing code assigned to messages that are not sent in direct response to an MQSC command.

Specify one of:

1. A value in the range 1 through 16, inclusive.

2. A list of values, separated by a comma and enclosed in parentheses. Each value must be in the range 1 through 16, inclusive.

The default is 1.

For more information about z/OS routing codes, see the *MVS Routing and Descriptor Codes* manual.

#### **SERVICE**

This field is reserved for use by IBM.

#### **SMFACCT**

Specifies whether WebSphere MQ sends accounting data to SMF automatically when the queue manager starts.

Specify one of:

**NO** Do not start gathering accounting data automatically.

**YES** Start gathering accounting data automatically for the default class 1.

The default is NO.

#### **SMFSTAT**

Specifies whether to gather SMF statistics automatically when the queue manager starts.

Specify one of:

**NO** Do not start gathering statistics automatically.

**YES** Start gathering statistics automatically for the default class 1.

The default is NO.

#### **STATIME**

Specifies the default time, in minutes, between consecutive gatherings of statistics.

Specify a number in the range zero through 1440.

If you specify a value of zero, both statistics data and accounting data is collected at the SMF data collection broadcast. See "Using System Management Facility" on page 85 for information about setting this.

The default is 30.

#### **TRACSTR**

Specifies whether global tracing is to start automatically.

Specify one of:

**NO** Do not start global tracing automatically.

**YES** Start global tracing automatically for the default class, class 1.

**integers** A list of classes for which global tracing is to be started automatically in the range 1 through 4.

**\*** Start global trace automatically for all classes.

The default is NO if you do not specify the keyword in the macro.

**Note:** The supplied default system parameter load module (CSQZPARM) has TRACSTR=YES (set in the assembler module CSQFSYSP). If you do not want to start tracing automatically, either create your own system parameter module, or issue the STOP TRACE command after the queue manager has started.

For details about the STOP TRACE command, see the WebSphere MQ Script (MQSC) Command Reference manual.

### TRACTBL

Specifies the default size, in 4 KB blocks, of trace table where the global trace facility stores WebSphere MQ trace records.

Specify a value in the range 1 through 999.

The default is 99. This is equivalent to 396 KB.

**Note:** Storage for the trace table is allocated in the ECSA. Therefore, you must select this value with care.

### WLMTIME

Specifies the time (in minutes or seconds depending on the value of WLMTIMU) between each scan of the indexes for WLM-managed queues.

Specify a value in the range 1 through 9999.

The default is 30.

### WLMTIMU

Time units used with the WLMTIME parameter.

Specify one of :

**MINS** WLMTIME represents a number of minutes.

**SECS** WLMTIME represents a number of seconds.

The default is MINS.

## Using CSQ6LOGP

Use CSQ6LOGP to establish your logging options.

The default parameters for CSQ6LOGP, and whether you can alter each parameter using the SET LOG command, are shown in Table 9. If you need to change any of these values, refer to the detailed descriptions of the parameters.

*Table 9. Default values of CSQ6LOGP parameters*

Parameter	Description	Default value	SET command
DEALLCT	Length of time an archive tape unit remains unused before it is deallocated.	zero	X
INBUFF	Size of input buffer storage for active and archive log data sets.	60 KB	–
MAXARCH	Maximum number of archive log volumes that can be recorded.	500	X
MAXRTU	Maximum number of dedicated tape units allocated to read archive log tape volumes concurrently.	2	X
OFFLOAD	Archiving on or off.	YES (ON)	–
OUTBUFF	Size of output buffer storage for active and archive log data sets.	4 000 KB	–
TWOACTV	Single or dual active logging.	YES (dual)	–
TWOARCH	Single or dual archive logging.	YES (dual)	–

Table 9. Default values of CSQ6LOGP parameters (continued)

Parameter	Description	Default value	SET command
TWOBSDS	Single or dual BSDS.	YES (dual BSDS)	–
WRTHRSR	Number of output buffers to be filled before they are written to the active log data sets.	20	X

### DEALLCT

Specifies the length of time, in minutes, that an archive read tape unit is allowed to remain unused before it is deallocated.

Specify one of the following:

- Time, in minutes, in the range zero through 1440
- NOLIMIT

Specifying 1440 or NOLIMIT means that the tape unit is never deallocated.

The default is zero.

When archive log data is being read from tape, it is recommended that you set this value high enough to allow WebSphere MQ to optimize tape handling for multiple read applications.

### INBUFF

Specifies the size, in kilobytes, of the input buffer for reading the active and archive logs during recovery. Use a decimal number in the range 28 through 60. The value specified is rounded up to a multiple of 4.

The default is 60 KB.

Suggested settings:

<b>Test system</b>	28 KB
<b>Production system</b>	60 KB

Set this to the maximum for best log read performance.

### MAXARCH

Specifies the maximum number of archive log volumes that can be recorded in the BSDS. When this number is exceeded, recording begins again at the start of the BSDS.

Use a decimal number in the range 10 through 1000.

The default is 500.

Suggested settings:

<b>Test system</b>	500 (default)
<b>Production system</b>	1 000

Set this to the maximum so that the BSDS can record as many logs as possible.

For information about the logs and BSDS, see the WebSphere MQ for z/OS Concepts and Planning Guide.

## MAXRTU

Specifies the maximum number of dedicated tape units that can be allocated to read archive log tape volumes concurrently.

This parameter and the DEALLCT parameter allow WebSphere MQ to optimize archive log reading from tape devices.

Specify a value in the range 1 through 99.

The default is 2.

It is recommended that you set the value to be at least one less than the number of tape units available to WebSphere MQ. If you do otherwise, the off-load process could be delayed, which could affect the performance of your system. For maximum throughput during archive log processing, specify the largest value possible for this option, remembering that you need at least one tape unit for off-load processing.

## OFFLOAD

Specifies whether archiving is on or off.

Specify either:

**YES** Archiving is on

**NO** Archiving is off

The default is YES.

**Attention:** Do **not** switch archiving off unless you are working in a test environment. If you do switch it off, you cannot guarantee that data will be recovered in the event of a system or transaction failure.

## OUTBUFF

Specifies the total size, in kilobytes, of the storage to be used by WebSphere MQ for output buffers for writing the active and archive log data sets. Each output buffer is 4 KB.

The parameter must be in the range 40 through 4000. The value specified is rounded up to a multiple of 4.

The default is 4 000 KB.

Suggested settings:

**Test system** 400 KB

**Production system** 4 000 KB

Set this value to the maximum to avoid running out of log output buffers.

## TWOACTV

Specifies single or dual active logging.

Specify either:

**NO** Single active logs

**YES** Dual active logs

The default is YES.

For more information about the use of single and dual logging, refer to the WebSphere MQ for z/OS Concepts and Planning Guide.

## TWOARCH

Specifies the number of archive logs that WebSphere MQ produces when the active log is off-loaded.

Specify either:

**NO** Single archive logs

**YES** Dual archive logs

The default is YES.

Suggested settings:

**Test system** NO

**Production system** YES (default)

For more information about the use of single and dual logging, refer to the WebSphere MQ for z/OS Concepts and Planning Guide.

## TWOBSDS

Specifies the number of bootstrap data sets.

Specify either:

**NO** Single BSIDS

**YES** Dual BSIDS

The default is YES.

For more information about the use of single and dual logging, refer to the WebSphere MQ for z/OS Concepts and Planning Guide.

## WRTHRSH

Specifies the number of 4 KB output buffers to be filled before they are written to the active log data sets.

The larger the number of buffers, the less often the write takes place, and this improves the performance of WebSphere MQ. The buffers might be written before this number is reached if significant events, such as a commit point, occur.

Specify the number of buffers in the range 1 through 256.

The default is 20.

## Using CSQ6ARVP

Use CSQ6ARVP to establish your archiving environment.

The default parameters for CSQ6ARVP, and whether you can alter each parameter using the SET ARCHIVE command, are shown in Table 10. If you need to change any of these values, refer to the detailed descriptions of the parameters. Planning your archive storage is discussed in the WebSphere MQ for z/OS Concepts and Planning Guide.

Table 10. Default values of CSQ6ARVP parameters

Parameter	Description	Default value	SET command
ALCUNIT	Units in which primary and secondary space allocations are made.	BLK (blocks)	X
ARCPFX1	Prefix for first archive log data set name.	CSQARC1	X



Table 10. Default values of CSQ6ARVP parameters (continued)

Parameter	Description	Default value	SET command
ARCPFX2	Prefix for second archive log data set name.	CSQARC2	X
ARCRETN	The retention period of the archive log data set in days.	9999	X
ARCWRTC	List of route codes for messages to the operator about archive log data sets.	1,3,4	X
ARCWTOR	Whether to send message to operator and wait for reply before trying to mount an archive log data set.	YES	X
BLKSIZE	Block size of archive log data set.	28 672	X
CATALOG	Whether archive log data sets are cataloged in the ICF.	NO	X
COMPACT	Whether archive log data sets should be compacted.	NO	X
PRIQTY	Primary space allocation for DASD data sets.	25 715	X
PROTECT	Whether archive log data sets are protected by ESM profiles when the data sets are created.	NO	X
QUIESCE	Maximum time, in seconds, allowed for quiesce when ARCHIVE LOG with MODE(QUIESCE) specified.	5	X
SECQTY	Secondary space allocation for DASD data sets. See the ALCUNIT parameter for the units to be used.	540	X
TSTAMP	Whether the archive data set name should include a time stamp.	NO	X
UNIT	Device type or unit name on which the first copy of archive log data sets is stored.	TAPE	X
UNIT2	Device type or unit name on which the second copy of archive log data sets is stored.	Blank	X

### ALCUNIT

Specifies the unit in which primary and secondary space allocations are made.

Specify one of:

**CYL** Cylinders

**TRK** Tracks

**BLK** Blocks

You are recommended to use BLK because it is independent of the device type.

The default is BLK.

If free space on the archive DASD volumes is likely to be fragmented, you are recommended to specify a smaller primary extent and allow expansion into secondary extents. For more information about space allocation for active logs, refer to the WebSphere MQ for z/OS Concepts and Planning Guide.

### ARCPFX1

Specifies the prefix for the first archive log data set name.

See the TSTAMP parameter for a description of how the data sets are named and for restrictions on the length of ARCPFX1.

This parameter cannot be left blank.

The default is CSQARC1.

You might need to authorize the userid associated with the WebSphere MQ queue manager address space to create archive logs with this prefix.

#### **ARCPFX2**

Specifies the prefix for the second archive log data set name.

See the TSTAMP parameter for a description of how the data sets are named and for restrictions on the length of ARCPFX2.

This parameter cannot be blank even if the TWOARCH parameter is specified as NO.

The default is CSQARC2.

You might need to authorize the userid associated with the WebSphere MQ queue manager address space to create archive logs with this prefix.

#### **ARCRETN**

Specifies the retention period, in days, to be used when the archive log data set is created.

The parameter must be in the range zero through 9999.

The default is 9999.

Suggested settings:

<b>Test system</b>	3	In a test system, archive logs are probably not required over long periods.
<b>Production system</b>	9 999 (default)	Set this value high to effectively switch automatic archive log deletion off.

Discarding archive log data sets is discussed in the WebSphere MQ for z/OS System Administration Guide.

#### **ARCWRTC**

Specifies the list of z/OS routing codes for messages about the archive log data sets to the operator. This field is ignored if ARCWTOR is set to NO.

Specify up to 14 routing codes, each with a value in the range 1 through 16. You must specify at least one code. Separate codes in the list by commas, not by blanks.

The default is the list of values: 1,3,4.

For more information about z/OS routing codes, see the *MVS Routing and Descriptor Codes* manual.

#### **ARCWTOR**

Specifies whether a message is to be sent to the operator and a reply is received before attempting to mount an archive log data set.

Other WebSphere MQ users might be forced to wait until the data set is mounted, but they are not affected while WebSphere MQ is waiting for the reply to the message.

Specify either:

**YES** The device needs a long time to mount archive log data sets. For example, a tape drive.

**NO** The device does not have long delays. For example, DASD.

The default is YES.

Suggested settings:

**Test system** NO

**Production system** YES (default)

This is dependent on operational procedures. If tape robots are used, NO might be more appropriate.

### **BLKSIZE**

Specifies the block size of the archive log data set. The block size you specify must be compatible with the device type you specify in the UNIT parameter.

The parameter must be in the range 4 097 through 28 672. The value you specify is rounded up to a multiple of 4 096.

The default is 28 672.

This parameter is ignored for data sets that are managed by the storage management subsystem (SMS).

If the archive log data set is written to DASD, you are recommended to choose the maximum block size that allows 2 blocks for each track. For example, for a 3390 device, you should use a block size of 24 576.

If the archive log data set is written to tape, specifying the largest possible block size improves the speed of reading the archive log.

Suggested settings:

**Test system** 24 576 (6 log records for each block)

This is the optimum block size for 3390 DASD.

**Production system** 28 672 (the maximum allowed; 7 log records for each block)

Use the highest possible block size for optimum tape I/O efficiency.

### **CATALOG**

Specifies whether archive log data sets are cataloged in the primary integrated catalog facility (ICF) catalog.

Specify either:

**NO** Archive log data sets are not cataloged

**YES** Archive log data sets are cataloged

The default is NO.

All archive log data sets allocated on DASD must be cataloged. If you archive to DASD with the CATALOG parameter set to NO, message CSQJ072E is displayed each time an archive log data set is allocated, and WebSphere MQ catalogs the data set.

Suggested settings:

<b>Test system</b>	YES
<b>Production system</b>	NO (default)

### COMPACT

Specifies whether data written to archive logs is to be compacted. This option applies only to a 3480 or 3490 device that has the improved data recording capability (IDRC) feature. When this feature is turned on, hardware in the tape control unit writes data at a much higher density than normal, allowing for more data on each volume. Specify NO if you do not use a 3480 device with the IDRC feature or a 3490 base model, with the exception of the 3490E. Specify YES if you want the data to be compacted.

Specify either:

**NO** Do not compact the data sets

**YES** Compact the data sets

The default is NO.

Specifying YES adversely affects performance. Also be aware that data compressed to tape can be read only using a device that supports the IDRC feature. This can be a concern if you have to send archive tapes to another site for remote recovery.

Suggested settings:

<b>Test system</b>	Not applicable
<b>Production system</b>	NO (default)

This applies to 3480 and 3490 IDR compression only. Setting this to YES might degrade archive log read performance during recovery and restart; however, it does not affect writing to tape.

### PRIQTY

Specifies the primary space allocation for DASD data sets in ALCUNITs.

The value must be greater than zero.

The default is 25 715.

This value must be sufficient for a copy of either the log data set or its corresponding BSDS, whichever is the larger. To determine the necessary value, follow this procedure:

1. Determine the number of active log records actually allocated (c) as explained in "Task 14: Create the bootstrap and log data sets" on page 24.
2. Determine the number of 4096-byte blocks in each archive log block:

$$d = \text{BLKSIZE} / 4096$$

where BLKSIZE is the rounded up value.

3. If ALCUNIT=BLK:

$$\text{PRIQTY} = \text{INT}(c / d) + 1$$

where INT means round down to an integer.

If ALCUNIT=TRK:

$$\text{PRIQTY} = \text{INT}(c / (d * \text{INT}(e/\text{BLKSIZE}))) + 1$$

where e is the number of bytes for each track (56664 for a 3390 device) and INT means round down to an integer.

If ALCUNIT=CYL:

$$\text{PRIQTY} = \text{INT}(c / (d * \text{INT}(e/\text{BLKSIZE}) * f)) + 1$$

where f is the number of tracks for each cylinder (15 for a 3390 device) and INT means round down to an integer.

For information about how large to make your log and archive data sets, see “Task 14: Create the bootstrap and log data sets” on page 24 and “Task 15: Define your page sets” on page 25.

Suggested settings:

**Test system** 1 680

Sufficient to hold the entire active log, that is:

$$10\ 080 / 6 = 1\ 680 \text{ blocks}$$

**Production system** Not applicable when archiving to tape.

If free space on the archive DASD volumes is likely to be fragmented, you are recommended to specify a smaller primary extent and allow expansion into secondary extents. For more information about space allocation for active logs, refer to the WebSphere MQ for z/OS Concepts and Planning Guide.

**PROTECT**

Specifies whether archive log data sets are to be protected by discrete ESM (external security manager) profiles when the data sets are created.

Specify either:

**NO** Profiles are not created.

**YES** Discrete data set profiles are created when logs are off-loaded. If you specify YES:

- ESM protection must be active for WebSphere MQ.
- The user ID associated with the WebSphere MQ queue manager address space must have authority to create these profiles.
- The TAPEVOL class must be active if you are archiving to tape.

Otherwise, off-loads will fail.

The default is NO.

## QUIESCE

Specifies the maximum time in seconds allowed for the quiesce when an ARCHIVE LOG command is issued with MODE(QUIESCE) specified.

The parameter must be in the range 1 through 999.

The default is 5.

## SECQTY

Specifies the secondary space allocation for DASD data sets in ALCUNITs. The secondary extent can be allocated up to 15 times; see the *z/OS MVS™ JCL Reference* and *z/OS MVS JCL User's Guide* for details.

The parameter must be greater than zero.

The default is 540.

## TSTAMP

Specifies whether the archive log data set name has a time stamp in it.

Specify either:

**NO** Names do not include a time stamp. The archive log data sets are named:

*arcpfxi.Annnnnnn*

Where *arcpfxi* is the data set name prefix specified by ARCPFX1 or ARCPFX2. *arcpfxi* can have up to 35 characters.

**YES** Names include a time stamp. The archive log data sets are named:

*arcpfxi.cyyddd.Thhmsst.Annnnnnn*

where *c* is 'D' for the years up to and including 1999 or 'E' for the year 2000 and later, and *arcpfxi* is the data set name prefix specified by ARCPFX1 or ARCPFX2. *arcpfxi* can have up to 19 characters.

**EXT** Names include a time stamp. The archive log data sets are named:

*arcpfxi.Dyyyyddd.Thhmsst.Annnnnnn*

Where *arcpfxi* is the data set name prefix specified by ARCPFX1 or ARCPFX2. *arcpfxi* can have up to 17 characters.

The default is NO.

## UNIT

Specifies the device type or unit name of the device that is used to store the first copy of the archive log data set.

Specify a device type or unit name of 1 through 8 alphanumeric characters. The first character must be alphabetic.

This parameter cannot be blank.

The default is TAPE.

If you archive to DASD, you can specify a generic device type with a limited volume range.

If you archive to DASD:

- Make sure that the primary space allocation is large enough to contain all the data from the active log data sets.
- Make sure that the archive log data set catalog option (CATALOG) is set to YES.

- The archive log data sets cannot extend to another volume.

If you archive to TAPE, WebSphere MQ can extend to a maximum of 20 volumes.

Suggested settings:

<b>Test system</b>	DASD
<b>Production system</b>	TAPE

For more information about choosing a location for archive logs, refer to the WebSphere MQ for z/OS Concepts and Planning Guide.

## UNIT2

Specifies the device type or unit name of the device that is used to store the second copy of the archive log data sets.

Specify a device type or unit name of 1 through 8 alphanumeric characters. The first character must be alphabetic. If this parameter is blank, the value set for the UNIT parameter is used.

The default is blank.

## Task 18: Tailor the channel initiator parameters

Use ALTER QMGR to customize the channel initiator to suit your requirements.

- *Repeat this task for each WebSphere MQ queue manager, as required.*
- *You need to perform this task when migrating from a previous version. For details, see WebSphere MQ Migration Information.*

A number of queue manager attributes control how distributed queueing operates. Set these attributes using the MQSC command ALTER QMGR. The initialization data set sample thlqual.SCSQPROC(CSQ4INYG) contains some settings that you can customize. For more information, see ALTER QMGR.

The values of these parameters are displayed as a series of messages each time you start the channel initiator.

### The relationship between adapters, dispatchers and maximum number of channels

The ALTER QMGR parameters CHIADAPS and CHIDISPS define the number of task control blocks (TCBs) used by the channel initiator. CHIADAPS (adapter) TCBs are used to make MQI calls to the queue manager. CHIDISPS (dispatcher) TCBs are used to make calls to the communications network.

The ALTER QMGR parameter MAXCHL influences the distribution of channels over the dispatcher TCBs.

#### CHIDISPS

If you have a small number of channels use the default value.

We suggest CHIDISPS(20) for systems with more than 100 channels. There is unlikely to be any significant disadvantage in having CHIDISPS(20) where this is more dispatcher TCBs than necessary.

As a guideline, if you have more than 1000 channels, allow one dispatcher for every 50 current channels. For example, specify CHIDISPS(40) in order to handle up to 2000 active channels.

If you are using TCP/IP, the maximum number of dispatchers used for TCP/IP channels is 100, even if you specify a larger value in CHIDISPS.

### CHIADAPS

Each MQI call to the queue manager is independent of any other and can be made on any adapter TCB. Calls using persistent messages can take much longer than those for nonpersistent messages because of log I/O. Thus a channel initiator processing a large number of persistent messages across many channels may need more than the default 8 adapter TCBs for optimum performance. This is particularly so where achieved batchsize is small, because end of batch processing also requires log I/O, and where thin client channels are used.

The suggested value for a production environment is CHIADAPS(30). Using more than this is unlikely to give any significant extra benefit, and there is unlikely to be any significant disadvantage in having CHIADAPS(30) if this is more adapter TCBs than necessary.

### MAXCHL

Each channel is associated with a particular dispatcher TCB at channel start and remains associated with that TCB until the channel stops. Many channels can share each TCB. MAXCHL is used to spread channels across the available dispatcher TCBs. The first (  $\text{MIN}(\text{MAXCHL} / \text{CHIDISPS})$ , 10 ) channels to start are associated with the first dispatcher TCB and so on until all dispatcher TCBs are in use.

The effect of this for small numbers of channels and a large MAXCHL is that channels are NOT evenly distributed across dispatchers. For example, if you set CHIDISPS(10) and left MAXCHL at its default value of 200 but had only 50 channels, five dispatchers would be associated with 10 channels each and five would be unused. We suggest setting MAXCHL to the number of channels actually to be used where this is a small fixed number.

## Task 19: Set up Batch, TSO, and RRS adapters

- Repeat this task for each WebSphere MQ queue manager as required.
- You might need to perform this task when migrating from a previous version. For details, see *WebSphere MQ Migration Information*.

To make the adapters available to batch and other applications using batch connections, add the following WebSphere MQ libraries to the STEPLIB concatenation for your batch application :

- thlqual.SCSQANLx
- thlqual.SCSQAUTH



where *x* is the language letter for your national language. (You do not need to do this if the libraries are in the LPA or the link list.)

For TSO applications add the libraries to the STEPLIB concatenation in the TSO logon procedure or activate them using the TSO command TSOLIB.

If the adapter detects an unexpected WebSphere MQ error, it issues an z/OS SNAP dump to DDname CSQSNAP, and issues reason code MQRD\_UNEXPECTED\_ERROR to the application. If the CSQSNAP DD statement is not in the application JCL or CSQSNAP is not allocated to a data set under TSO, no dump is taken. If this happens, you could include the CSQSNAP DD statement in the application JCL or allocate CSQSNAP to a data set under TSO and rerun the application. However, because some problems are intermittent, it is recommended that you include a CSQSNAP statement in the application JCL or allocate CSQSNAP to a data set in the TSO logon procedure to capture the reason for failure at the time it occurs.

The supplied program CSQBDEFV improves the portability of your application programs. In CSQBDEFV, you can specify the name of a queue manager, or queue sharing group, to be connected to rather than specifying it in the MQCONN or MQCONNX call in an application program. You can create a new version of CSQBDEFV for each queue manager, or queue sharing group. To do this, follow these steps:

1. Copy the WebSphere MQ assembler program CSQBDEFV from thlqual.SCSQASMS to a user library.
2. The supplied program contains the default subsystem name CSQ1. You can retain this name for testing and installation verification. For production subsystems, you can change the NAME=CSQ1 to your one- to four-character subsystem name, or use CSQ1.

If you are using queue-sharing groups, you can specify a queue-sharing group name instead of CSQ1. If you do this, the program issues a connect request to an active queue manager within that group.

3. Assemble and link-edit the program to produce the CSQBDEFV load module. For the assembly, include the library thlqual.SCSQMACS in your SYSLIB concatenation; use the link-edit parameters RENT,AMODE=31,RMODE=ANY. This is shown in the sample JCL in thlqual.SCSQPROC(CSQ4DEFV). Then include the load library in the z/OS Batch or the TSO STEPLIB, ahead of thlqual.SCSQAUTH.

## Task 20: Set up the operations and control panels

- You need to perform this task once for each z/OS system where you want to run WebSphere MQ.
- You might need to perform this task when migrating from a previous version. For details, see *WebSphere MQ Migration Information*.

To set up the operations and control panels you must first set up the libraries that contain the required panels, EXECs, messages, and tables. To do this, you must take into account which national language feature is to be used for the panels. When you have done this, you can optionally:

- Update the main ISPF menu for WebSphere MQ operations and control panels
- Change the function key settings

## Setting up the libraries

Follow these steps to set up the WebSphere MQ operations and control panels:

1. Ensure that all the libraries contained in your concatenations are either in the same format (F, FB, V, VB) and have the same block size, or are in order of decreasing block sizes. Otherwise, you might have problems trying to use these panels.
2. Include the library `thlqual.SCSQEXEC` in your `SYSEXEC` or `SYSPROC` concatenation or activate it using the TSO `ALTLIB` command. This library, which is allocated with a fixed-block 80 record format during installation, contains the required EXECs.

It is preferable to put the library into your `SYSEXEC` concatenation. However, if you want to put it in `SYSPROC`, the library must have a record length of 80 bytes.

3. Add `thlqual.SCSQAUTH` to the TSO logon procedure `STEPLIB` or activate it using the TSO `TSOLIB` command, if it is not in the link list or the LPA.
4. You can either add the WebSphere MQ panel libraries permanently to your ISPF library setup, or allow them to be set up dynamically when the panels are used. For the former choice, you need to do the following:
  - a. Include the library containing the operations and control panel definitions in your `ISPPLIB` concatenation. The name is `thlqual.SCSQPNLx`, where `x` is the language letter for your national language.
  - b. Include the library containing the required tables in your `ISPTLIB` concatenation. The name is `thlqual.SCSQTBLx`, where `x` is the language letter for your national language.
  - c. Include the library containing the required messages in your `ISPMLIB` concatenation. The name is `thlqual.SCSQMSGx`, where `x` is the language letter for your national language.
  - d. Include the library containing the required load modules in your `ISPLLIB` concatenation. The name of this library is `thlqual.SCSQAUTH`.
5. Test that you can access the WebSphere MQ panels from the TSO Command Processor panel. This is usually option 6 on the ISPF/PDF Primary Options Menu. The name of the EXEC that you run is `CSQOREXX`. There are no parameters to specify if you have put the WebSphere MQ libraries permanently in your ISPF setup as in step 4. If you have not, use the following:

```
CSQOREXX thlqual langletter
```

where `langletter` is a letter identifying the national language to be used:

- |          |                           |
|----------|---------------------------|
| <b>C</b> | Simplified Chinese        |
| <b>E</b> | U.S. English (mixed case) |
| <b>K</b> | Japanese                  |
| <b>U</b> | U.S. English (uppercase)  |

## Updating the ISPF menu

You can update the ISPF main menu to allow access to the WebSphere MQ operations and control panels from ISPF. The required setting for `&ZSEL` is:

```
CMD(%CSQOREXX thlqual langletter)
```

For information about thlqual and langletter, see Step 5 on page 48.

For more details, see the *ISPF Dialog Developer's Guide and Reference* manual.

### Updating the function keys and command settings

You can use the normal ISPF procedures for changing the function keys and command settings used by the panels. The application identifier is CSQO.

However, this is *not* recommended because the help information is not updated to reflect any changes that you have made.

## Task 21: Include the WebSphere MQ dump formatting member

- You need to perform this task once for each z/OS system where you want to run WebSphere MQ.
- You need to perform this task when migrating from a previous version. For details, see *WebSphere MQ Migration Information*.

To be able to format WebSphere MQ dumps using the Interactive Problem Control System (IPCS), copy the data set thlqual.SCSQPROC(CSQ7IPCS) to SYS1.PARMLIB. You should not need to edit this data set.

If you have customized the TSO procedure for IPCS, thlqual.SCSQPROC(CSQ7IPCS) can be copied into any library in the IPCSPARM definition. See the *MVS IPCS Customization* manual for details on IPCSPARM.

You must also include the library thlqual.SCSQPDLA in your ISPLIB concatenation.

To make the dump formatting programs available to your TSO session or IPCS job, you must also include the library thlqual.SCSQAUTH in your STEPLIB concatenation or activate it using the TSO TSOLIB command (even if it is already in the link list or LPA).

## Task 22: Suppress information messages

- You need to perform this task once for each z/OS system where you want to run WebSphere MQ.
- You do not need to perform this task when migrating from a previous version.

If your WebSphere MQ system is heavily used, with many channels stopping and starting, a large number of information messages are sent to the z/OS console and hardcopy log. The WebSphere MQ-IMS bridge and buffer manager might also produce a large number of information messages.

If required, you can suppress some of these console messages by using the z/OS message processing facility list, specified by the MPFLSTxx members of SYS1.PARMLIB. The messages you specify still appear on the hardcopy log, but not on the console.

Sample thlqual.SCSQPROC(CSQ4MPFL) shows suggested settings for MPFLSTxx. See the *MVS Initialization and Tuning Reference* manual for more information about MPFLSTxx.

If you want to suppress selected information messages on the hardcopy log, you can use the z/OS installation exit IEAVMXIT. You can set the following bit switches ON for the required messages:

**CTXTRDTM**

Delete the message.

The message is not displayed on consoles or logged in hardcopy.

**CTXTESJL**

Suppress from job log.

The message does not go into the JES job log.

**CTXTNWTP**

Do not carry out WTP processing.

The message is not sent to a TSO terminal or to the system message data set of a batch job.

**Note:**

1. For full details, refer to the *MVS Installation Exits* book.
2. You are not recommended to suppress messages other than those in the suggested suppression list, CSQ4MPFL.

---

## Chapter 2. Migrating from a previous version

Information about migrating a single queue manager from a previous version of WebSphere MQ for z/OS, coexistence with earlier versions of WebSphere MQ for z/OS, and migrating from an unsupported release of MQSeries® has been moved to *WebSphere MQ Migration Information*.



---

## Chapter 3. Testing your queue manager

When you have customized or migrated your queue manager, you can test it by running some of the sample applications shipped with WebSphere MQ.

You can then compile and link-edit whichever of the other samples are appropriate to your installation using the sample JCL supplied.

This chapter tells you about:

- “Running the basic installation verification program”
- “Testing for queue-sharing groups” on page 57
- “Testing for distributed queuing” on page 58
- “Testing for C, C++, COBOL, PL/I, and CICS” on page 61

---

### Running the basic installation verification program

After you have installed and customized WebSphere MQ, you can use the supplied installation verification program, CSQ4IVP1, to confirm that WebSphere MQ is operational. This is a batch assembler IVP that verifies the base WebSphere MQ without using the C, COBOL, or CICS samples.

The Batch Assembler IVP is link-edited by SMP/E and the load modules are shipped in library thlqual.SCSQLOAD.

After you have completed both the SMP/E APPLY step and the customization steps, run the Batch Assembler IVP.

### Overview of the CSQ4IVP1 application

CSQ4IVP1 is a batch application that connects to your WebSphere MQ subsystem and performs these basic functions:

- Issues WebSphere MQ calls
- Communicates with the command server
- Verifies triggering is active
- Generates and deletes a dynamic queue
- Verifies message expiry processing
- Verifies message commit processing

### Preparing to run CSQ4IVP1

Before you run CSQ4IVP1:

1. Check that the IVP entries are in the CSQINP2 data set concatenation in the queue manager startup program. The IVP entries are supplied in member thlqual.SCSQPROC(CSQ4IVPQ). If not, add the definitions supplied in thlqual.SCSQPROC(CSQ4IVPQ) to your CSQINP2 concatenation. If the queue manager is currently running, you need to restart it so that these definitions can take effect.
2. The sample JCL, CSQ4IVPR, required to run the installation verification program is in library thlqual.SCSQPROC.

Customize the CSQ4IVPR JCL with the high-level qualifier for the WebSphere MQ libraries, the national language you want to use, the four-character WebSphere MQ queue manager name, and the destination for the job output.

- Update RACF to allow CSQ4IVP1 to access its resources if WebSphere MQ security is active.

To run CSQ4IVP1 when WebSphere MQ security is enabled, you need a RACF user ID with authority to access the objects. For details of defining resources to RACF, see Chapter 7, “Setting up security,” on page 123. The user ID that runs the IVP must have the following access authority:

Authority	Profile	Class
READ	ssid.DISPLAY.PROCESS	MQCMDS
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.COMMAND.REPLY.MODEL	MQQUEUE
UPDATE	ssid.CSQ4IVP1.**	MQQUEUE
READ	ssid.BATCH	MQCONN

These requirements assume that all WebSphere MQ security is active. The RACF commands to activate WebSphere MQ security are shown in Figure 4. This example assumes that the queue manager name is CSQ1 and that the user ID of the person running sample CSQ4IVP1 is TS101.

```
RDEFINE MQCMDS CSQ1.DISPLAY.PROCESS
PERMIT CSQ1.DISPLAY.PROCESS CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.REPLY.MODEL
PERMIT CSQ1.SYSTEM.COMMAND.REPLY.MODEL CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.CSQ4IVP1.**
PERMIT CSQ1.CSQ4IVP1.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)
```

Figure 4. RACF commands for CSQ4IVP1

## Running CSQ4IVP1

When you have completed these steps, start your queue manager. If the queue manager is already running and you have made changes to CSQINP2, you must stop the queue manager and restart it.

The IVP runs as a batch job. Customize the job card to meet the submission requirements of your installation.

## Checking the results of CSQ4IVP1

The IVP is split into ten stages; each stage must complete with a zero completion code before the next stage is run. The IVP generates a report, listing:

- The name of queue manager that is being connected to.



- A one-line message showing the completion code and the reason code returned from each stage.
- A one-line informational message where appropriate.

A sample report is provided in Figure 5 on page 57

For an explanation of the completion and reason codes, see the WebSphere MQ for z/OS Messages and Codes manual.

Some stages have more than one WebSphere MQ call and, in the event of failure, a message is issued indicating the specific WebSphere MQ call that returned the failure. Also, for some stages the IVP puts explanatory and diagnostic information into a comment field.

The IVP job requests exclusive control of certain queue manager objects and therefore should be single threaded through the system. However, there is no limit to the number of times the IVP can be run against your queue manager.

The functions performed by each stage are:

#### **Stage 1**

Connect to the queue manager by issuing **MQCONN**.

#### **Stage 2**

Determine the name of the system-command input queue used by the command server to retrieve request messages. This queue receives display requests from Stage 5.

To do this, the sequence of calls is:

1. Issue an **MQOPEN**, specifying the queue manager name, to open the queue manager object.
2. Issue an **MQINQ** to find out the name of the system-command input queue.
3. Issue an **MQINQ** to find out about various queue manager event switches.
4. Issue an **MQCLOSE** to close the queue manager object.

On successful completion of this stage, the name of the system-command input queue is displayed in the comment field.

#### **Stage 3**

Open an initiation queue using **MQOPEN**.

This queue is opened at this stage in anticipation of a trigger message, which arrives as a result of the command server replying to the request from Stage 5. The queue must be opened for input to meet the triggering criteria.

#### **Stage 4**

Create a permanent dynamic queue using the CSQ4IVP1.MODEL queue as a model. The dynamic queue has the same attributes as the model from which it was created. This means that when the replies from the command server request in Stage 5 are written to this queue, a trigger message is written to the initiation queue opened in Stage 3.

Upon successful completion of this stage, the name of the permanent dynamic queue is indicated in the comment field.

**Stage 5**

Issue an **MQPUT1** request to the command server command queue.

A message of type **MQMT\_REQUEST** is written to the system-command input queue requesting a display of process **CSQ4IVP1**. The message descriptor for the message specifies the permanent dynamic queue created in Stage 4 as the reply-to queue for the command server's response.

**Stage 6**

Issue an **MQGET** request from the initiation queue. At this stage, a **GET WAIT** with an interval of one minute is issued against the initiation queue opened in Stage 3. The message returned is expected to be the trigger message generated by the command server's response messages being written to the reply-to queue.

**Stage 7**

Delete the permanent dynamic queue created in Stage 4. As the queue still has messages on it, the **MQCO\_PURGE\_DELETE** option is used.

**Stage 8**

1. Open a dynamic queue.
2. **MQPUT** a message with an expiry interval set.
3. Wait for the message to expire.
4. Attempt to **MQGET** the expired message.
5. **MQCLOSE** the queue.

**Stage 9**

1. Open a dynamic queue.
2. **MQPUT** a message.
3. Issue **MQCMIT** to commit the current unit of work.
4. **MQGET** the message.
5. Issue **MQBACK** to backout the message.
6. **MQGET** the same message and ensure the backout count is set to 1.
7. Issue **MQCLOSE** to close the queue.
- 8.

**Stage 10**

Disconnect from the queue manager using **MQDISC**.

After running the IVP, you can delete any objects that you no longer require.

If the IVP does not run successfully, try each step manually to find out which function is failing.



RACF, see Chapter 7, “Setting up security,” on page 123. The user ID that runs the IVP must have the following access authority in addition to that required to run the basic IVP:

Authority	Profile	Class
UPDATE	ssid.CSQ4IVPG.**	MQQUEUE

These requirements assume that all WebSphere MQ security is active. The RACF commands to activate WebSphere MQ security are shown in Figure 6. This example assumes that the queue manager name is CSQ1 and that the user ID of the person running sample CSQ4IVP1 is TS101.

```
RDEFINE MQQUEUE CSQ1.CSQ4IVPG.**
PERMIT CSQ1.CSQ4IVPG.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)
```

Figure 6. RACF commands for CSQ4IVP1 for a queue-sharing group

## Running CSQ4IVP1 for a queue-sharing group

When you have completed these steps, start your queue manager. If the queue manager is already running and you have made changes to CSQINP2, you must stop the queue manager and restart it.

The IVP runs as a batch job. Customize the job card to meet the submission requirements of your installation.

## Checking the results of CSQ4IVP1 for a queue-sharing group

The IVP for queue-sharing groups works in the same way as the basic IVP, except that the queues that are created are called CSQIVPG.xx. Follow the instructions given in “Checking the results of CSQ4IVP1” on page 54 to check the results of the IVP for queue-sharing groups.

---

## Testing for distributed queuing

You can use the supplied installation verification program, CSQ4IVPX, to confirm that distributed queuing is operational.

## Overview of CSQ4IVPX job

CSQ4IVPX is a batch job that starts the channel initiator and issues the WebSphere MQ DISPLAY CHINIT command. This verifies that all major aspects of distributed queuing are operational, while avoiding the need to set up channel and network definitions.

## Preparing to run CSQ4IVPX

Before you run CSQ4IVPX:

1. The sample JCL, CSQ4IVPX, required to run the installation verification program is in library thlqual.SCSQPROC.

Customize the CSQ4IVPX JCL with the high-level qualifier for the WebSphere MQ libraries, the national language you want to use, the four-character queue manager name, and the destination for the job output.

- Update RACF to allow CSQ4IVPX to access its resources if WebSphere MQ security is active. To run CSQ4IVPX when WebSphere MQ security is enabled, you need a RACF user ID with authority to access the objects. For details of defining resources to RACF, see Chapter 7, “Setting up security,” on page 123. The user ID that runs the IVP must have the following access authority:

Authority	Profile	Class
CONTROL	ssid.START.CHINIT and ssid.STOP.CHINIT	MQCMDS
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.CSQUUTIL.*	MQQUEUE
READ	ssid.BATCH	MQCONN
READ	ssid.DISPLAY.CHINIT	MQCMDS

These requirements assume that the connection security profile ssid.CHIN has been defined (as shown in “Connection security profiles for the channel initiator” on page 135), and that all WebSphere MQ security is active. The RACF commands to do this are shown in Figure 7 on page 60. This example assumes that:

- The queue manager name is CSQ1
  - The user ID of the person running sample CSQ4IVPX is TS101
  - The channel initiator address space is running under the user ID CSQ1MSTR
- Update RACF to allow the channel initiator address space the following access authority:

Authority	Profile	Class
READ	ssid.CHIN	MQCONN
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.CHANNEL.INITQ	MQQUEUE
UPDATE	ssid.SYSTEM.CHANNEL.SYNCQ	MQQUEUE
ALTER	ssid.SYSTEM.CLUSTER.COMMAND.QUEUE	MQQUEUE
UPDATE	ssid.SYSTEM.CLUSTER.TRANSMIT.QUEUE	MQQUEUE
ALTER	ssid.SYSTEM.CLUSTER.REPOSITORY.QUEUE	MQQUEUE
CONTROL	ssid.CONTEXT.**	MQADMIN

The RACF commands to do this are also shown in Figure 7 on page 60.

```

RDEFINE MQCMDS CSQ1.DISPLAY.DQM
PERMIT CSQ1.DISPLAY.DQM CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQCMDS CSQ1.START.CHINIT
PERMIT CSQ1.START.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQCMDS CSQ1.STOP.CHINIT
PERMIT CSQ1.STOP.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101,CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CSQUTIL.*
PERMIT CSQ1.SYSTEM.CSQUTIL.* CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)

RDEFINE MQCONN CSQ1.CHIN
PERMIT CSQ1.CHIN CLASS(MQCONN) ID(CSQ1MSTR) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.SYNCQ
PERMIT CSQ1.SYSTEM.CHANNEL.SYNCQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.INITQ
PERMIT CSQ1.SYSTEM.CHANNEL.INITQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQADMIN CSQ1.CONTEXT.**
PERMIT CSQ1.CONTEXT.** CLASS(MQADMIN) ID(CSQ1MSTR) ACCESS(CONTROL)

```

Figure 7. RACF commands for CSQ4IVPX

## Running CSQ4IVPX

When you have completed these steps, start your queue manager.

The IVP runs as a batch job. Customize the job card to meet the submission requirements of your installation.

## Checking the results of CSQ4IVPX

CSQ4IVPX runs the CSQUTIL WebSphere MQ utility to issue three MQSC commands. The SYSPRINT output data set should look like Figure 8 on page 61, although details might differ depending on your queue manager attributes .

- You should see the commands **(1)** each followed by several messages.
- The last message from each command should be “CSQ9022I ... NORMAL COMPLETION” **(2)**.
- The job as a whole should complete with return code zero **(3)**.

```

CSQU000I CSQUTIL IBM WebSphere MQ for z/OS - V6
CSQU001I CSQUTIL Queue Manager Utility - 2005-05-09 09:06:48
COMMAND
CSQU127I CSQUTIL Executing COMMAND using input from CSQUCMD data set
CSQU120I CSQUTIL Connecting to queue manager CSQ1
CSQU121I CSQUTIL Connected to queue manager CSQ1
CSQU055I CSQUTIL Target queue manager is CSQ1
START CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM138I +CSQ1 CSQMSCHI CHANNEL INITIATOR STARTING
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' START CHINIT' NORMAL COMPLETION
(2)
DISPLAY CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMDDQM DISPLAY CHINIT COMMAND ACCEPTED
CSQN205I COUNT= 12, RETURN=00000000, REASON=00000000
CSQX830I +CSQ1 CSQXRDQM Channel initiator active
CSQX002I +CSQ1 CSQXRDQM Queue-sharing group is QSG1
CSQX831I +CSQ1 CSQXRDQM 8 adapter subtasks started, 8 requested
CSQX832I +CSQ1 CSQXRDQM 5 dispatchers started, 5 requested
CSQX833I +CSQ1 CSQXRDQM 0 SSL server subtasks started, 0 requested
CSQX840I +CSQ1 CSQXRDQM 0 channel connections current, maximum 200
CSQX841I +CSQ1 CSQXRDQM 0 channel connections active, maximum 200,
including 0 paused
CSQX842I +CSQ1 CSQXRDQM 0 channel connections starting,
0 stopped, 0 retrying
CSQX836I +CSQ1 Maximum channels - TCP/IP 200, LU 6.2 200
CSQX845I +CSQ1 CSQXRDQM TCP/IP system name is TCP/IP
CSQX848I +CSQ1 CSQXRDQM TCP/IP listener INDISP=QMGR not started
CSQX848I +CSQ1 CSQXRDQM TCP/IP listener INDISP=GROUP not started
CSQX849I +CSQ1 CSQXRDQM LU 6.2 listener INDISP=QMGR not started
CSQX849I +CSQ1 CSQXRDQM LU 6.2 listener INDISP=GROUP not started
CSQ9022I +CSQ1 CSQXCRPS ' DISPLAY CHINIT' NORMAL COMPLETION
(2)
STOP CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMTCHI STOP CHINIT COMMAND ACCEPTED
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' STOP CHINIT' NORMAL COMPLETION
(2)
CSQU057I CSQUCMDS 3 commands read
CSQU058I CSQUCMDS 3 commands issued and responses received, 0 failed
CSQU143I CSQUTIL 1 COMMAND statements attempted
CSQU144I CSQUTIL 1 COMMAND statements executed successfully
CSQU148I CSQUTIL Utility completed, return code=0
(3)

```

Figure 8. Example output from CSQ4IVPX

## Testing for C, C++, COBOL, PL/I, and CICS

You can test for C, C++, COBOL, PL/I, or CICS, using the sample applications supplied with WebSphere MQ. Although the IVP (CSQ4IVP1) is supplied as a load module, the samples are supplied as source modules.

For more information about sample applications, see the WebSphere MQ Application Programming Reference and WebSphere MQ Using C++ manuals.





---

## Chapter 4. Customizing for CICS

This section applies to CICS Transaction Server for z/OS V3.1 and earlier. If you are using CICS Transaction Server for z/OS V3.2, see the 'CICS integration with WebSphere MQ' section in the CICS Transaction Server for z/OS Version 3.2 Information Center at: <http://publib.boulder.ibm.com/infocenter/cicsts/v3r2/index.jsp>. for further information.

---

### Setting up the CICS adapter

The WebSphere MQ-CICS adapter (generally referred to in this book as the CICS adapter) is required to use WebSphere MQ within CICS.

This chapter tells you how to make the CICS adapter available to your CICS subsystem. If you are not familiar with defining resources to CICS, refer to:

- The *CICS System Definition Guide* for general information on setting up a CICS subsystem.
- The *CICS Resource Definition Guide*, for background information on defining resources to CICS, details of and the command syntax of the CEDA transaction, and the MIGRATE command.
- The *CICS Operations and Utilities Guide* and the *CICS Resource Definition Guide* for details of the CSD utility program (DFHCSDUP).

### Resource definition

This section takes you through the steps you must perform to define the resources for the CICS adapter.

#### Updating the CSD

This section describes the updates required for the CICS system definition (CSD) data set for the CICS adapter. If you intend to use the CICS sample application programs, see the WebSphere MQ Application Programming Guide .

You must use resource definition online (RDO) to add new groups to the CSD data set. The new groups must contain definitions of:

- The supplied adapter programs
- The supplied adapter management transactions
- The supplied sets of BMS maps, required for the adapter panels

To update the CSD, run the CICS offline utility program, DFHCSDUP, with the supplied sample input data sets:

This data set...	Provides the definitions required for...	
thlqual.SCSQPROC(CSQ4B100)	CICS adapter	Required
thlqual.SCSQPROC(CSQ4S100)	Supplied samples	Optional

Each of these data sets contains sample CICS definitions that must be tailored. To preserve the originals, copy these data sets into a user JCL library whose name contains the WebSphere MQ subsystem name, for example, MQS.CSQ1.USERJCL, and tailor them there.

**Note:** With some versions of CICS, you might receive warning messages about obsolete keywords; you can ignore these.

Ensure that any user-written CICS applications that issue MQI calls, and the resources they use, are also defined to the CSD. You can edit the input data set, to include definitions of user-programs and their resources.

You can add this fragment of JCL to your CSD upgrade (DFHCSDUP) job to define the WebSphere MQ supplied groups to the CICS CSD:

```
//SYSIN DD DSN=thlqual.SCSQPROC(CSQ4B100),DISP=SHR
//      DD DSN=thlqual.SCSQPROC(CSQ4S100),DISP=SHR
//      DD *
//      ADD GROUP(CSQCAT1) LIST(yourlist)
//      ADD GROUP(CSQ4SAMP) LIST(yourlist)
/*
```

Figure 9. JCL fragment for upgrading the CICS CSD

Here, `yourlist` is the name of a CICS list that contains a list of groups to be installed by CICS during a cold start of the system. This is specified in the `GRPLIST` parameter of your CICS system initialization table (SIT). For details of CICS SIT parameters, see the *CICS System Definition Guide*.

Include the new resource groups in the CICS startup group list. For information about resource groups, installing them in CICS, the CICS CSD, and DFHCSDUP, see the *CICS Resource Definition Guide*.

**Note:** If you use the CEDA transaction to install redefined adapter resources in an active CICS system, you must first shut down the adapter and wait until the alert monitor has finished its work.

If you want to use CICS program autoinstall rather than define the programs to the CICS CSD, you must ensure that the required programs are available to WebSphere MQ. To do this, ensure that the autoinstalled definitions map to those supplied in member `thlqual.SCSQPROC(CSQ4B100)`.

## Starting a connection automatically during CICS initialization

If you want the adapter to connect to WebSphere MQ automatically during CICS initialization, the `CSQCCODF` program should be included in a CICS PLTPI program. `CSQCCODF` must execute during the third stage of CICS initialization and must therefore be added after the entry for `DFHDELIM`. If there is no entry for `DFHDELIM` in your current PLTPI, you must add one.

Alternatively, if your version of CICS supports it, you can use the `MQCONN` SIT parameter to connect to WebSphere MQ automatically. See the *CICS System Definition Guide* for information about this parameter.

Instead of using `CSQCCODF`, you can write your own program; see “Writing a PLTPI program to start the connection” on page 67.

1. Use the CICS DFHPLT macro to add your program to the list of programs executed by CICS during the third stage initialization. Figure 10 shows how to code the entry for CSQCCODF in a CICS PLT program called DFHPLT41. For information about coding PLT entries, see the *CICS Resource Definition Guide*.

```
DFHPLT41 DFHPLT TYPE=INITIAL,SUFFIX=41
         DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
         DFHPLT TYPE=ENTRY,PROGRAM=CSQCCODF
         DFHPLT TYPE=FINAL
         END
```

Figure 10. Sample PLT for use with the CICS adapter. This sample assumes that you are using the supplied PLTPI program, CSQCCODF, to start the adapter.

2. Specify the particular list of programs to be run at initialization by naming the suffix of your PLT on the PLTPI system initialization parameter. In Figure 10, the PLT suffix is 41.

**Note:** You can use the CICS adapter in a CICS system that has interregion communication (IRC) to remote CICS systems. If you are using IRC, you should ensure that the IRC facility is OPEN before you start the adapter. This is essential if the IRC access method is defined as cross memory, that is, ACCESSMETHOD(XM).

## System definition

Use the INITPARM parameter in the CICS system initialization table (SIT), or the SYSIN override, to set the default connection parameters. Figure 11 shows you how to do this.

```
INITPARM=(CSQCPARM='SN=CSQ1,TN=001,IQ=CICS01.INITQ')
```

Figure 11. Sample INITPARM statement to set the default connection values for CICS

Where:

- SN** The subsystem name. This must be the name of a queue manager, not a queue-sharing group.
- TN** The trace number to identify the adapter in CICS trace entries. This must be in the range zero through 199.
- IQ** The name of the default initiation queue. If this is blank, and you do not specify an initiation queue name by any other method, an instance of CKTI is not started when the CICS adapter connects to the queue manager.

The INITPARM statement does not accept a parameter string longer than 60 characters. If you specify a 4-character subsystem name and a 3-character trace number, the maximum allowable length of the initiation queue name is 42 characters. If you need a queue name longer than 42 characters, you cannot use the INITPARM statement to specify the default initiation queue.

At connect time, you must override the INITPARM setting, either by using the CKQC transaction, or in a PLTPI program.

1. If you are using a PLTPI program to start the adapter, code the suffix of your PLT on the PLTPI system initialization parameter. See Figure 10 on page 65 for an example of this.
2. Add the following WebSphere MQ libraries to the STEPLIB concatenation in your CICS procedure in the following order:
  - thlqual.SCSQANLx
  - thlqual.SCSQAUTH

Where x is the language letter for your national language.
3. Add the following WebSphere MQ libraries to the DFHRPL concatenation in your CICS procedure in the following order, even if they are in the LPA or link list:
  - thlqual.SCSQANLx
  - thlqual.SCSQCICS
  - thlqual.SCSQAUTH

Where x is the language letter for your national language.

If you are using any CICS programs that dynamically call the WebSphere MQ CICS stub, CSQCSTUB, also add thlqual.SCSQLOAD to the DFHRPL concatenation.

If you are using the API-crossing exit (CSQCAPX), also add the name of the library that contains the load module for the program.
4. Update CSQINP2. You can use the sample CSQ4INYG, but you might need to change the initiation queue name to match your system definition.

For more information about:

- The CICS initiation queue, see the *WebSphere MQ for z/OS Concepts and Planning Guide*.
- The CKQC transaction, see the *WebSphere MQ for z/OS System Administration Guide*
- PLTPI programs, see “Writing a PLTPI program to start the connection” on page 67.
- Coding CICS system initialization parameters, see the *CICS System Definition Guide*.

## SNAP dumps

If the CICS adapter detects an unexpected WebSphere MQ error, it issues a z/OS SNAP dump to DDname CSQSNAP and issues reason code MQRC\_UNEXPECTED\_ERROR to the application. If the CSQSNAP DD statement was not in the CICS startup JCL, no dump is taken. If this happens, you could include the CSQSNAP DD statement in the startup JCL and rerun the application. However, because some problems might be intermittent, it is recommended that you include the CSQSNAP DD statement to capture the reason for failure at the time it occurs.

## Completing the connection from CICS

The connection is completed when the CICS adapter completes these steps:

1. Enable the CICS adapter and initialize the control blocks.
2. Attach the z/OS subtasks and identify CICS generic *applId* (as specified in the CICS system initialization parameters as the connection ID) to WebSphere MQ. This is described in the *CICS System Definition Guide*.

These two steps are done for you automatically if you use the INITPARM parameter or the CKQC transaction (this is described in the WebSphere MQ for z/OS System Administration Guide). You can also use a PLTPI program to do this; see “Writing a PLTPI program to start the connection.”

When the connection is complete, a pending event called a *termination notification* is activated. This pending event remains active until the queue manager terminates in either an orderly or a forced way. When the pending event expires (or matures), it causes a FORCE shutdown request to be issued to the CICS adapter, and the pending event is canceled.

## Controlling CICS application connections

Every CICS transaction that issues calls to WebSphere MQ is assigned a unique thread ID to service the requests and keep track of changes made to WebSphere MQ resources. The thread ID is created the first time a transaction issues a WebSphere MQ request, and accompanies all subsequent WebSphere MQ requests made by that transaction.

While executing work under the CICS main task TCB, the CICS adapter queues WebSphere MQ requests for processing by any of the eight subtask TCBs. These subtask TCBs are attached by the adapter when the connection to WebSphere MQ is established.

## Customizing the CICS adapter

You can customize the CICS adapter by:

- Writing a user version of CSQCCODF that can be included in a CICS PLTPI program. See “Writing a PLTPI program to start the connection” for more information.
- Writing an API-crossing exit program. See “The API-crossing exit” on page 68 for more information.

### Writing a PLTPI program to start the connection

You can write your own PLTPI program, based on the supplied assembler sample `thlqual.SCSQASMS(CSQCSPLT)`.

Although this sample is written in assembler, you can write your own program in any language supported by CICS. A typical use of PLTPI programs is for overriding the INITPARM settings if your CICS adapter initiation queue name is too long. (You cannot use more than 42 characters for an initiation queue name in an INITPARM statement.) If your PLTPI program gets its input parameters from a data set, you do not need an INITPARM statement.

Your PLTPI program must link to the adapter connect program, `thlqual.SCSQCICS(CSQCQCON)`, and pass a parameter list that specifies the connection values to be used. The parameter list is described in the WebSphere MQ for z/OS System Administration Guide. Figure 12 on page 68 shows the LINK command that your PLTPI program must issue. In this example, the parameter list is named CONNPL. Because no terminals are available at this stage of CICS start up, you must use the COMMAREA option to pass the parameter list.

```
EXEC CICS LINK PROGRAM('CSQCQCON')
      COMMAREA(CONNPL) LENGTH(length of CONNPL)
```

Figure 12. Linking to the adapter connect program, CSQCQCON, from a PLT program. The COMMAREA option is used, because no terminals are currently available.

For more information about writing CICS PLTPI programs, see the *CICS Customization Guide*.

## The API-crossing exit

WebSphere MQ provides an API-crossing exit for use with the CICS adapter; it runs in the CICS address space. You can use this exit to intercept WebSphere MQ calls as they are being run, for monitoring, testing, maintenance, or security purposes. If you are using CICS Transaction Server V3.2, you must write your exit program to be threadsafe and declare your exit program as threadsafe. If you are using earlier CICS releases, you are also recommended to write and declare your exit programs as threadsafe to be ready for migrating to CICS Transaction Server V3.2..

The sample API-crossing exit is supplied in source form only. For more information about writing API-crossing exit programs, see the WebSphere MQ Application Programming Guide

**Note:** Using the API-crossing exit degrades WebSphere MQ performance. You should plan your use of it carefully.

### Defining the exit program:

Before the API-crossing exit can be used, an exit program load module must be available when the CICS adapter connects to WebSphere MQ. The exit program is a CICS program that must be named CSQCAPX and reside in a library in the DFHRPL concatenation. CSQCAPX must be defined in the CICS system definition file (CSD) and must be enabled.

When CSQCAPX is loaded a confirmation message is written to the CICS adapter control panel, CKQC, or the console. If it cannot be loaded, a diagnostic message is displayed, but otherwise the application program runs normally.

---

## Customizing the CICS bridge

The WebSphere MQ-CICS bridge is an optional component that allows WebSphere MQ to input and output to and from existing programs and transactions that are not WebSphere MQ-enabled.

This chapter describes what you have to do to customize the WebSphere MQ-CICS bridge. The bridge is described in the WebSphere MQ for z/OS Concepts and Planning Guide and WebSphere MQ Application Programming Reference.

### Prerequisite APARs

To run 3270 transactions, you must be using CICS Transaction Server for OS/390® Version 1 Release 3 or later. Version 1 Release 3 requires APAR PQ23961.

Before you can run the bridge you must ensure that your z/OS system has both the CICS and WebSphere MQ components in place. To run transactions using the

bridge, you must have the LE runtime environment installed, and have a link to the LE runtime library SCEERUN included in the z/OS link list. For further information about the link list, see “Task 3: Update the z/OS link list and LPA” on page 8.

## Setting up CICS

1. Run the resource definition utility DFHCSDUP, using the sample thlqual.SCSQPROC(CSQ4CKBC) as input, to define the bridge transactions and programs:

<b>CKBR</b>	Bridge monitor transaction
<b>CSQBCDI</b>	Data conversion exit
<b>CSQCBR00</b>	Bridge monitor program
<b>CKBP</b>	Bridge ProgramLink transaction
<b>CSQCBP00</b>	Bridge ProgramLink program
<b>CSQCBP10</b>	Bridge ProgramLink abend handler program
<b>CSQCBE30</b>	3270 bridge exit for WebSphere MQ (CICS Transaction Server, Version 1.3)

2. Add the load library to the DFHRPL concatenation of your CICS startup JCL.
3. Add the group, CSQCKB, to your startup group list.

### Note:

1. The bridge uses CICS temporary storage IDs with the prefix CKB. You should make sure these are not recoverable.
2. By default, your CICS DPL programs are run under transaction code CKBP. The transaction to be run can be specified in the MQCIH CICS-bridge header in the message. For more information, see the WebSphere MQ Application Programming Reference manual. You need to change the TASKDATALOC attribute to 'BELOW' if you are going to run 24-bit programs, otherwise you will get a CICS abend AEZC.

If you want to run your programs under different transaction codes you need to install copies of the definition of CKBP, changing the transaction name to the ones of your choice. DPL bridge transactions must not be routed to a remote system.

3. To ensure that the bridge monitor can route requests to the correct region, each CICS region that runs a bridge monitor must have CICS links defined to all of the other regions that run bridge monitors.

## Setting up WebSphere MQ

1. Define a local queue for the request messages.

You can use the sample thlqual.SCSQPROC(CSQ4CKBM) to define a queue named SYSTEM.CICS.BRIDGE.QUEUE, or define your own. If you define your own, set the following attribute:

### SHARE

So that both the monitor and the bridge tasks can read it.

If recovery is required, set the following attributes:

### DEFPSIST(YES)

Set messages as persistent on the queue by default.

## **HARDENBO**

Set **HARDENBO** to ensure that the count of the number of times that a message has been backed out is accurate.

## **BACKOUTQ**

If the bridge encounters an error while processing a message, it will attempt to write the request messages to the backout requeue queue for the queue. If no backout queue is specified, or it cannot be used, the messages will be put to the dead-letter queue.

If you want to process messages in FIFO sequence, set the following attribute:

## **MSGDLVSQ(FIFO)**

Otherwise, messages will be processed in priority sequence.

If the request queue is defined with **QSGDISP(SHARED)**, you must also define it with **INDXTYPE(CORRELID)**. This setting is also recommended for non-shared queues.

If you want to start the bridge by triggering, set the following attributes:

## **TRIGGER TRIGTYPE(FIRST) PROCESS(procid)**

where **procid** is a process specifying **APPLICID(CKBR)**. For more information on running the **CKBR** transaction, see *WebSphere MQ for z/OS System Administration Guide*.

2. Define one or more queues to hold the responses, as required. If your response queue is remote, you must define a transmission queue to hold the responses before they are forwarded to the response queue.
3. Ensure that a dead-letter queue is defined and a procedure is defined for processing messages on this queue.
4. Ensure that the LE libraries are included in the CICS library concatenation.
5. Ensure that the WebSphere MQ-CICS adapter is enabled.

If the bridge is to be accessed remotely from WebSphere MQ, you need channel and transmission queue definitions, and a remote queue definition for the request queue. For more information about using remote queues see *WebSphere MQ Intercommunication*.

Consider specifying **BOQNAME** and **BOTHRESH** on the bridge requests queue to ensure that messages are put to the specified backout queue when a message has been processed and backed out **BOTHRESH** times, rather than being placed on the dead-letter queue. If you specify a backout queue, put a process in place to process messages on this queue.

If 3270 bridge reply messages are sent to a non-z/OS system, the sender channel should specify the **CONVERT(YES)** option.

### **Note:**

1. The WebSphere MQ queue defined to hold requests for the CICS bridge must not be used by any other application.
2. Do not attempt to run multiple bridge monitors on a shared queue with bridge monitors attached to a system earlier than WebSphere MQ Version 5.3.1. This will lead to unpredictable results.



## Security

You might need to add RACF definitions, depending on the authentication option you choose to use. See “Security considerations for the CICS bridge” on page 197 for more information about this.

## Controlling CICS bridge throughput

### Single bridge monitor

You can control the throughput of the bridge by putting the bridge transaction CKBP in a class of its own, and setting the CLASSMAXTASK to suit your requirements.

Request messages browsed by the bridge monitor CKBR are marked and hidden for a period of time specified as the Mark Browse Interval (MarkBInt). This allows time for CKBP (when it has been started by CKBR) to retrieve the message. If the message is not retrieved within time MarkBInt, it will become visible again for reprocessing.

If, for whatever reason, CKBP is not retrieving messages from the queue, reprocessing will continue indefinitely. This is the default action of the CICS bridge in this circumstance.

You can change the default action by specifying the ROUTEMEM=Y parameter on the bridge start data. This will cause messages to be routed to the Dead Letter Queue (DLQ) when their mark expires and they become visible for reprocessing. For information on specifying CICS bridge start parameters see *WebSphere MQ for z/OS System Administration Guide, Operating the CICS bridge*.

### Multiple bridge monitors

If a high volume of requests is expected, you could consider starting a second or subsequent monitor task in another CICS region. Such monitors can each have a separate request queue for their sole use, which you must create. In this case you could give each monitor different service characteristics, but this has the disadvantage that applications have to know the names of the various queues.

You can avoid this problem by having several bridge monitors sharing the same request queue. In this case you must ensure that:

- All transactions in a 3270 pseudoconversation specify the remote SYSID returned by the first transaction in all subsequent messages in the pseudoconversation. This is required even if you use CICS transaction routing facilities to direct the transactions to other CICS regions.
- If you use passtickets for user validation, all bridge monitors for a queue specify the same PASSTKTA applid.
- Each CICS region running a bridge monitor has a unique SYSID and there are CICS ISC links between the CICS regions.

When doing problem determination with multiple CICS bridge monitors you might have to look at all of the logs of all of the CICS regions to find any error messages produced by the bridge. You can use the command `DISPLAY QSTATUS(queue name) TYPE(HANDLE)` on each queue manager to show which CICS regions have the queue open.



---

## Chapter 5. Customizing for IMS

---

### Setting up the IMS adapter

The WebSphere MQ-IMS adapter (generally referred to in this book as the IMS adapter) is required to use WebSphere MQ within IMS.

This section tells you how to make the IMS adapter available to your IMS subsystem. If you are not familiar with tailoring an IMS subsystem, see the *IMS Customization Guide*.

To make the IMS adapter available to IMS applications, follow these steps:

1. Define WebSphere MQ to IMS as an external subsystem using the IMS external subsystem attach facility (ESAF). See “Defining WebSphere MQ to IMS” on page 74.
2. Include the WebSphere MQ load library `thlqual.SCSQAUTH` in the JOBLIB or STEPLIB concatenation in the JCL for your IMS control region and for any dependent region that connects to WebSphere MQ (if it is not in the LPA or link list). If your JOBLIB or STEPLIB is not authorized, also include it in the DFSESL concatenation after the library containing the IMS modules (usually IMS RESLIB).

Also include `thlqual.SCSQANLx` (where x is the language letter).

3. Copy the WebSphere MQ assembler program `CSQQDEFV` from `thlqual.SCSQASMS` to a user library.
4. The supplied program, `CSQQDEFV`, contains one subsystem name `CSQ1` identified as default with an IMS language interface token (LIT) of `MQM1`. You can retain this name for testing and installation verification. For production subsystems, you can change the `NAME=CSQ1` to your own subsystem name or use `CSQ1`. You can add further subsystem definitions as required. See “Defining WebSphere MQ queue managers to the IMS adapter” on page 77.
5. Assemble and link-edit the program to produce the `CSQQDEFV` load module. For the assembly, include the library `thlqual.SCSQMACS` in your SYSLIB concatenation; use the link-edit parameters `RENT,AMODE=31,RMODE=ANY`. This is shown in the sample JCL in `thlqual.SCSQPROC(CSQ4DEFV)`.
6. Include the user library containing the module `CSQQDEFV` that you created in the JOBLIB or STEPLIB concatenation in the JCL for any dependent region that connects to WebSphere MQ. If you do not do this, you will receive a user 3041 abend from IMS.
7. If the IMS adapter detects an unexpected WebSphere MQ error, it issues a z/OS SNAP dump to DDname `CSQSNAP` and issues reason code `MQRC_UNEXPECTED_ERROR` to the application. If the `CSQSNAP DD` statement was not in the IMS dependent region JCL, no dump is taken. If this happens, you could include the `CSQSNAP DD` statement in the JCL and rerun the application. However, because some problems might be intermittent, it is recommended that you include the `CSQSNAP DD` statement to capture the reason for failure at the time it occurs.
8. If you want to use dynamic WebSphere MQ calls (described in the WebSphere MQ Application Programming Guide), build the dynamic stub, as shown in Figure 13 on page 74.

9. If you want to use the IMS trigger monitor, define the IMS trigger monitor application CSQQTRMN, and perform PSBGEN and ACBGEN. See “Setting up the IMS trigger monitor” on page 78.
10. If you are using RACF to protect resources in the OPERCMDS class, ensure that the userid associated with your WebSphere MQ queue manager address space has authority to issue the MODIFY command to any IMS system to which it might connect.

```
//DYNSTUB EXEC PGM=IEWL,PARM='RENT,REUS,MAP,XREF'
//SYSPRINT DD SYSOUT=*
//ACSQMOD DD DISP=SHR,DSN=thlqual.SCSQLOAD
//IMSLIB DD DISP=SHR,DSN=ims.reslib
//SYSLMOD DD DISP=SHR,DSN=private.load1
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,1)
//SYSLIN DD *
  INCLUDE ACSQMOD(CSQQSTUB)
  INCLUDE IMSLIB (DFSLI000)
  ALIAS MQCONN,MQCONN,MQDISC      MQI entry points
  ALIAS MQGET,MQPUT,MQPUT1       MQI entry points
  ALIAS MQOPEN,MQCLOSE          MQI entry points
  ALIAS MQBACK,MQCMIT           MQI entry points
  ALIAS CSQBBAK,CSQBCMT         MQI entry points
  ALIAS MQINQ,MQSET             MQI entry points
  ALIAS DFSPLI,PLITDLI          IMS entry points
  ALIAS DFSCOBOL,CBLTDLI        IMS entry points
  ALIAS DFSFOR,FORTDLI          IMS entry points
  ALIAS DFSASM,ASMTDLI          IMS entry points
  ALIAS DFSPASCL,PASTDLI        IMS entry points
  ALIAS DFHEI01,DFHEI1          IMS entry points
  ALIAS DFSAIBLI,AIBTDLI        IMS entry points
  ALIAS DFSESS,DSNWLI,DSNHLI    IMS entry points
  MODE AMODE(31),RMODE(ANY)     Note RMODE
  NAME CSQQDYS(R)
/*
```

<sup>1</sup>Specify the name of a library accessible to IMS applications that want to make dynamic calls to WebSphere MQ.

Figure 13. Sample JCL to link-edit the dynamic call stub. This includes the IMS language interface module and the WebSphere MQ IMS stub CSQQSTUB.

## Defining WebSphere MQ to IMS

WebSphere MQ must be defined to the control region, and to each dependent region accessing that WebSphere MQ queue manager. To do this, you must create a subsystem member (SSM) in the IMS.PROCLIB library, and identify the SSM to the applicable IMS regions.

### Placing the subsystem member entry in IMS.PROCLIB

Each SSM entry in IMS.PROCLIB defines a connection from an IMS region to a different queue manager.

To name an SSM member, concatenate the value (one to four alphanumeric characters) of the IMSID field of the IMS IMSCTRL macro with any name (one to four alphanumeric characters) defined by your site.

One SSM member can be shared by all the IMS regions, or a specific member can be defined for each region. This member contains as many entries as there are connections to external subsystems. Each entry is an 80-character record.

### Positional parameters:

The fields in this entry are:

SSN,LIT,ESMT,RTT,REO,CRC
--------------------------

where:

#### SSN

Specifies the WebSphere MQ queue manager name. It is required, and must contain one through four characters.

#### LIT

Specifies the language interface token (LIT) supplied to IMS. This field is required, its value must match one in the CSQQDEFV module.

#### ESMT

Specifies the external subsystem module table (ESMT). This table specifies which attachment modules must be loaded by IMS. CSQQESMT is the required value for this field.

#### RTT

This option is not supported by WebSphere MQ.

#### REO

Specifies the region error option (REO) to be used if an IMS application references a non-operational external subsystem or if resources are unavailable at create thread time. This field is optional and contains a single character, which can be:

**R** Passes a return code to the application, indicating that the request for WebSphere MQ services failed.

**Q** Ends the application with an abend code U3051, backs out activity to the last commit point, does a PSTOP of the transaction, and requeues the input message. This option only applies when an IMS application tries to reference a non-operational external subsystem or if the resources are unavailable at create thread time.

WebSphere MQ completion and reason codes are returned to the application if the WebSphere MQ problem occurs while WebSphere MQ is processing the request; that is, after the adapter has passed the request on to WebSphere MQ.

**A** Ends the application with an abend code of U3047 and discards the input message. This option only applies when an IMS application references a non-operational external subsystem or if the resources are unavailable at create thread time.

WebSphere MQ completion and reason codes are returned to the application if the WebSphere MQ problem occurs while WebSphere MQ is processing the request; that is, after the adapter has passed the request on to WebSphere MQ.

#### CRC

This option can be specified but is not used by WebSphere MQ.

An example SSM entry is:

```
CSQ1, MQM1, CSQQESMT, , R,
```

where:

**CSQ1** The default subsystem name as supplied with WebSphere MQ. You can change this to suit your installation.

**MQM1** The default LIT as supplied in CSQQDEFV.

**CSQQESMT** The external subsystem module name. You must use this value.

**R** REO option.

### Keyword parameters:

WebSphere MQ parameters can be specified in keyword format; to do this you must specify SST=DB2. Other parameters are as described in “Positional parameters” on page 75, and shown in the following example:

```
SST=DB2, SSN=SYS3, LIT=MQM3, ESMT=CSQQESMT
```

where:

**SYS3** The subsystem name

**MQM3** The LIT as supplied in CSQQDEFV

**CSQQESMT** The external subsystem module name

## Specifying the SSM EXEC parameter

Specify the SSM EXEC parameter in the start up procedure of the IMS control region. This parameter specifies the one-character to four-character subsystem member name (SSM).

If you specify the SSM for the IMS control region, any dependent region running under the control region can attach to the WebSphere MQ queue manager named in the IMS.PROCLIB member specified by the SSM parameter. The IMS.PROCLIB member name is the IMS ID (IMSID=xxxx) concatenated with the one to four characters specified in the SSM EXEC parameter. The IMS ID is the IMSID parameter of the IMSCTRL generation macro.

IMS lets you define as many external subsystem connections as are required. More than one connection can be defined for different WebSphere MQ queue managers. All WebSphere MQ connections must be within the same z/OS system. For a dependent region, you can specify a dependent region SSM or use the one specified for the control region. You can specify different region error options (REOs) in the dependent region SSM member and the control region SSM member. Table 11 shows the different possibilities of SSM specifications.

Table 11. SSM specifications options

SSM for control region	SSM for dependent region	Action	Comments
No	No	None	No external subsystem can be connected.

Table 11. SSM specifications options (continued)

SSM for control region	SSM for dependent region	Action	Comments
No	Yes	None	No external subsystem can be connected.
Yes	No	Use the control region SSM	Applications scheduled in the region can access external subsystems identified in the control region SSM. Exits and control blocks for each attachment are loaded into the control region and the dependent region address spaces.
Yes	Yes (empty)	No SSM is used for the dependent region	Applications scheduled in this region can access DL/I databases only. Exits and control blocks for each attachment are loaded into the control region address space.
Yes	Yes (not empty)	Check the dependent region SSM with the control region SSM	Applications scheduled in this region can access only external subsystems identified in both SSMs. Exits and control blocks for each attachment are loaded into the control region and the dependent region address spaces.

There is no specific parameter to control the maximum number of SSM specification possibilities.

### Preloading the IMS adapter

The performance of the IMS adapter can be improved if it is preloaded by IMS. Preloading is controlled by the DMSMPLxx member of IMS.PROCLIB: see "IMS Administration Guide: System" for more information. The WebSphere MQ module names to specify are:

CSQACLST	CSQAMLST	CSQAPRH	CSQAVICM	CSQFSALM	CSQQDEFV
CSQQCONN	CSQQDISC	CSQQTERM	CSQQINIT	CSQQBACK	CSQQCMMT
CSQQESMT	CSQQPREP	CSQQTHD	CSQQWAIT	CSQQNORM	CSQQSSOF
CSQQSSON	CSQQMTXT	CSQQRESV	CSQQSNOP	CSQQCMND	CSQQCVER
CSQQT MID	CSQQTRGI				

### Defining WebSphere MQ queue managers to the IMS adapter

The IMS adapter cannot access the IMS PROCLIB so the names of the WebSphere MQ queue managers and their corresponding LITs must be defined in the queue manager definition table, CSQQDEFV. Use the supplied CSQQDEFX macro to create the CSQQDEFV load module. Figure 14 shows the syntax of this assembler macro.

```
CSQQDEFX TYPE=ENTRY | DEFAULT, NAME=qmgr-name, LIT=token
or
CSQQDEFX TYPE=END
```

Figure 14. CSQQDEFX macro syntax

## Parameters

### TYPE=ENTRY|DEFAULT

Specify either TYPE=ENTRY or TYPE=DEFAULT as follows:

#### TYPE=ENTRY

Specifies that a table entry describing a WebSphere MQ queue manager available to an IMS application is to be generated. If this is the first entry, the table header is also generated, including a CSQQDEFV CSECT statement.

#### TYPE=DEFAULT

As for TYPE=ENTRY. The queue manager specified is the default queue manager to be used when MQCONN or MQCONNX specifies a name that is all blanks. There must be only one such entry in the table.

### NAME=qmgr-name

Specifies the name of the queue manager, as specified with MQCONN or MQCONNX.

### LIT=token

Specifies the name of the language interface token (LIT) that IMS uses to identify the queue manager.

An MQCONN or MQCONNX call associates the *name* input parameter and the *hconn* output parameter with the name label and, therefore, the LIT in the CSQQDEFV entry. Further WebSphere MQ calls passing the *hconn* parameter use the LIT from the CSQQDEFV entry identified in the MQCONN or MQCONNX call to direct calls to the WebSphere MQ queue manager defined in the IMS SSM PROCLIB member with that same LIT.

In summary, the *name* parameter on the MQCONN or MQCONNX call identifies a LIT in CSQQDEFV and the same LIT in the SSM member identifies a WebSphere MQ queue manager. (For information about the MQCONN and MQCONNX calls, see the WebSphere MQ Application Programming Reference manual.)

### TYPE=END

Specifies that the table is complete. If this parameter is omitted, TYPE=ENTRY is assumed.

## Using the CSQQDEFX macro

Figure 15 shows the general layout of a queue manager definition table.

```
CSQQDEFX NAME=subsystem1,LIT=token1
CSQQDEFX NAME=subsystem2,LIT=token2,TYPE=DEFAULT
CSQQDEFX NAME=subsystem3,LIT=token3
...
CSQQDEFX NAME=subsystemN,LIT=tokenN
CSQQDEFX TYPE=END
END
```

Figure 15. Layout of a queue manager definition table

## Setting up the IMS trigger monitor

Define the application to IMS using the model CSQQTAPL in the thlqual.SCSQPROC library (see Figure 16 on page 79).



Generate the PSB and ACB using the model CSQQTSPB in the thlqual.SCSQPROC library (see Figure 17).

```
* This is the application definition *
* for the IMS Trigger Monitor BMP *

      APPLCTN PSB=CSQQTRMN,
          PGMTYPE=BATCH,
          SCHDTYP=PARALLEL
```

Figure 16. Example transaction definition for CSQQTRMN

```
PCB  TYPE=TP,           ALTPCB for transaction messages
      MODIFY=YES,       To "triggered" IMS transaction
      PCBNAME=CSQQTRMN
PCB  TYPE=TP,           ALTPCB for diagnostic messages
      MODIFY=YES,       To LTERM specified or "MASTER"
      PCBNAME=CSQQTRMG,
      EXPRESS=YES
PSBGEN LANG=ASSEM,
        PSBNAME=CSQQTRMN,  Runs program CSQQTRMN
        CMPAT=YES
```

Figure 17. Example PSB definition for CSQQTRMN

---

## Customizing the IMS bridge

The WebSphere MQ-IMS bridge is an optional component that allows WebSphere MQ to input and output to and from existing programs and transactions that are not WebSphere MQ-enabled.

This chapter describes what you have to do to customize the WebSphere MQ-IMS bridge. The bridge is described in the WebSphere MQ for z/OS Concepts and Planning Guide.

### Define the XCF and OTMA parameters for WebSphere MQ.

This step defines the XCF group and member names for your WebSphere MQ system, and other OTMA parameters. WebSphere MQ and IMS must belong to the same XCF group. Use the OTMACON keyword of the CSQ6SYSP macro to tailor these parameters in the system parameter load module.

See "Using CSQ6SYSP" on page 28 for information about this.

### Define the XCF and OTMA parameters to IMS.

This step defines the XCF group and member names for the IMS system. IMS and WebSphere MQ must belong to the same XCF group.

Add the following parameters to your IMS parameter list, either in your JCL or in member DFSPBxxx in the IMS PROCLIB:

#### OTMA=Y

This starts OTMA automatically when IMS is started. (This is optional, if you specify OTMA=N you can also start OTMA by issuing the IMS command /START OTMA.)

#### GRNAME=

This gives the XCF group name.

This is the same as the group name specified in the storage class definition (see the next step), and in the Group parameter of the OTMACON keyword of the CSQ6SYSP macro.

**USERVAR=**

This gives the XCF member name of the IMS system.

This is the same as the member name specified in the storage class definition (see the next step).

If you do not specify a name for USERVAR, the value of APPLID1 is used.

**Tell WebSphere MQ the XCF group and member name of the IMS system.**

This is specified by the storage class of a queue. If you want to send messages across the WebSphere MQ-IMS bridge you need to specify this when you define the storage class for the queue. In the storage class, you need to define the XCF group and the member name of the target IMS system. To do this, either use the WebSphere MQ operations and control panels, or use the WebSphere MQ commands as described in the WebSphere MQ Script (MQSC) Command Reference manual.

**Set up the security that you require.**

The /SECURE OTMA IMS command determines the level of security to be applied to **every** WebSphere MQ queue manager that connects to IMS through OTMA. See “Security considerations for the IMS bridge” on page 199 for information about what this should be set to.

---

## Chapter 6. Monitoring performance and resource usage

---

### Introduction to monitoring

This section describes how to monitor the performance and resource usage of WebSphere MQ.

- It outlines some of the information that you can retrieve and briefly describes a general approach to investigating performance problems. (You can find information about dealing with performance problems in the WebSphere MQ for z/OS Problem Determination Guide.)
- It describes how you can collect statistics about the performance of WebSphere MQ by using SMF records.
- It describes how to gather accounting data to enable you to charge your customers for their use of your WebSphere MQ systems.
- It describes how to use WebSphere MQ events (alerts) to monitor your systems.

These are some of the tools you might use to monitor WebSphere MQ; they are described in the sections that follow:

- Tools provided by WebSphere MQ:
  - “Using DISPLAY commands”
  - “Using CICS adapter statistics” on page 82
  - “Using WebSphere MQ events” on page 84
- z/OS service aids:
  - “Using System Management Facility” on page 85
- Other IBM licensed programs:
  - “Using Resource Measurement Facility” on page 87
  - “Using Tivoli Decision Support for z/OS” on page 87
  - “Using the CICS monitoring facility” on page 87

Information about interpreting the data gathered by the performance statistics trace is given in “Interpreting WebSphere MQ performance statistics” on page 88.

Information about interpreting the data gathered by the accounting trace is given in “Interpreting WebSphere MQ accounting data” on page 104.

### Getting snapshots of WebSphere MQ

You can get an idea of the current state of WebSphere MQ by using the DISPLAY commands and, for the CICS adapter, the CICS adapter panels.

#### Using DISPLAY commands

You can use the WebSphere MQ MQSC DISPLAY or PCF Inquire commands to obtain information about the current state of WebSphere MQ. They provide information on the status of the command server, process definitions, queues, the queue manager, and so on. These commands are:

MQSC command	PCF command
DISPLAY ARCHIVE	Inquire Archive
DISPLAY AUTHINFO	Inquire Authentication Information Object
DISPLAY CFSTATUS	Inquire CF Structure Status
DISPLAY CFSTRUCT	Inquire CF Structure
DISPLAY CHANNEL	Inquire Channel
DISPLAY CHINIT	Inquire Channel Initiator
DISPLAY CHSTATUS	Inquire Channel Status
DISPLAY CMDSERV	
DISPLAY CLUSQMGR	Inquire Cluster Queue Manager
DISPLAY CONN	Inquire Connection
DISPLAY GROUP	Inquire Group
DISPLAY LOG	Inquire Log
DISPLAY PROCESS	Inquire Process
DISPLAY QMGR	Inquire Queue Manager
DISPLAY QSTATUS	Inquire Queue Status
DISPLAY QUEUE	Inquire Queue
DISPLAY SECURITY	Inquire Security
DISPLAY STGCLASS	Inquire Storage Class
DISPLAY SYSTEM	Inquire System
DISPLAY TRACE	
DISPLAY USAGE	Inquire Usage

For the detailed syntax of each command, see WebSphere MQ Script (MQSC) Command Reference or WebSphere MQ Programmable Command Formats and Administration Interface manual. All of the functions of these commands (except DISPLAY CMDSERV and DISPLAY TRACE) are also available through the operations and control panels.

These commands provide a snapshot of the system *only* at the moment the command was processed. If you want to examine trends in the system, you must start a WebSphere MQ trace and analyze the results over a period of time.

## Using CICS adapter statistics

If you are an authorized CICS user, you can use the CICS adapter control panels to display CICS adapter statistics dynamically. These statistics provide a snapshot of information related to CICS thread usage and situations when all threads are busy. The display connection panel can be refreshed by pressing the Enter key. For more information, see the WebSphere MQ for z/OS System Administration Guide.

## Using WebSphere MQ trace

You can record performance statistics and accounting data for WebSphere MQ by using the WebSphere MQ trace facility. The data generated by WebSphere MQ is sent to:

- The System Management Facility (SMF), specifically as SMF record type 115, subtypes 1 and 2 for the performance statistics trace

- The SMF, specifically as SMF record type 116, subtypes zero, 1, and 2 for the accounting trace.

If you prefer, the data generated by the WebSphere MQ accounting trace can also be sent to the generalized trace facility (GTF).

## Starting WebSphere MQ trace

You can start the WebSphere MQ trace facility at any time by issuing the WebSphere MQ START TRACE command.

Accounting data can be lost if the accounting trace is started or stopped while applications are running. To collect accounting data successfully, the following conditions must apply:

- The accounting trace must be active when an application starts, and it must still be active when the application finishes.
- If the accounting trace is stopped, any accounting data collection that was active stops.

You can also start collecting some trace information automatically if you specify YES on the SMFSTAT (SMF STATISTICS) and SMFACCT (SMF ACCOUNTING) parameters of the CSQ6SYSP macro (described in “Using CSQ6SYSP” on page 28).

You cannot use this method to start collecting class 3 accounting information (thread-level and queue-level accounting). You must use the START TRACE command to do this (however, you can include the command in your CSQINP2 input data set so that the trace is started automatically when you start your queue manager).

Before starting a WebSphere MQ trace, read “Using System Management Facility” on page 85.

## Controlling WebSphere MQ trace

To control the WebSphere MQ trace data collection at start up, specify values for the parameters in the CSQ6SYSP macro when you customize WebSphere MQ, see “Using CSQ6SYSP” on page 28 for details.

You can control WebSphere MQ tracing when the queue manager is running with these commands:

- START TRACE
- ALTER TRACE
- STOP TRACE

You can chose the destination to which trace data is sent. Possible destinations are:

**SMF** System Management Facility

**GTF** Generalized Trace Facility (accounting trace only)

**SRV** Serviceability routine for diagnostic use by IBM service personnel

For daily monitoring, information is sent to SMF (the default destination). SMF data sets usually contain information from other systems; this information is not available for reporting until the SMF data set is dumped.

You can also send accounting trace information to the GTF. This information has an event identifier of 5EE. The WebSphere MQ for z/OS Problem Determination Guide describes how to deal with WebSphere MQ trace information sent to the GTF.

For information about WebSphere MQ commands, see the WebSphere MQ Script (MQSC) Command Reference manual.

### Effect of trace on WebSphere MQ performance

Using the WebSphere MQ trace facility can have a significant effect on WebSphere MQ and transaction performance. For example, if you start a global trace for class 1 or for all classes, it is likely to increase CPU usage and transaction response times by approximately 50%. However, if you start a global trace for classes 2 to 4 alone, or a statistics or accounting trace, the increase in CPU usage and transaction response times is likely to be less than 1% additional CPU cost to the cost of WebSphere MQ calls.

## Using WebSphere MQ online monitoring

You can collect monitoring data for queues and channels (including automatically defined cluster-server channels) by setting the MONQ, MONCHL and MONACLS attributes. Table 12 summarizes the commands to set these attributes at different levels and to display the monitoring information.

Table 12. Setting and displaying attributes to control online monitoring

Attribute	Applicable at this level	Set using command	Display monitoring information using command
MONQ	Queue	DEFINE QLOCAL DEFINE QMODEL ALTER QLOCAL ALTER QMODEL	DISPLAY QSTATUS
	Queue manager	ALTER QMGR	
MONCHL	Channel	DEFINE CHANNEL ALTER CHANNEL	DISPLAY CHSTATUS
	Queue manager	ALTER QMGR	
MONACLS	Queue manager	ALTER QMGR	

For full details of these commands, see WebSphere MQ Script (MQSC) Command Reference. For more information about online monitoring, see Monitoring WebSphere MQ.

## Using WebSphere MQ events

WebSphere MQ *instrumentation events* provide information about errors, warnings, and other significant occurrences in a queue manager. You can monitor the operation of all your queue managers by incorporating these events into your own system management application.

WebSphere MQ instrumentation events fall into the following categories:

**Queue manager events**

These events are related to the definitions of resources within queue managers. For example, an application attempts to put a message to a queue that does not exist.

**Performance events**

These events are notifications that a threshold condition has been reached by a resource. For example, a queue depth limit has been reached, or the queue was not serviced within a predefined time limit.

**Channel events**

These events are reported by channels as a result of conditions detected during their operation. For example, a channel instance is stopped.

**Configuration events**

These events are notifications that an object has been created, changed or deleted.

When an event occurs, the queue manager puts an *event message* on the appropriate *event queue*, if defined. The event message contains information about the event that can be retrieved by a suitable WebSphere MQ application.

WebSphere MQ events can be enabled using the WebSphere MQ commands or the operations and control panels.

See Monitoring WebSphere MQ for information about the WebSphere MQ events that generate messages, and for information about the format of these messages. See WebSphere MQ Script (MQSC) Command Reference for information about enabling the events.

## Using System Management Facility

You can use SMF to collect statistics and accounting information. To use SMF, certain parameters must be set in z/OS and in WebSphere MQ.

System management facility (SMF) is a z/OS service aid used to collect information from various z/OS subsystems. This information is dumped and reported periodically, for example, hourly. You can use SMF with the WebSphere MQ trace facility to collect data from WebSphere MQ. In this way you can monitor *trends*, for example, in system utilization and performance, and collect accounting information about each user ID using WebSphere MQ.

To record performance statistics (record type 115) to SMF specify the following in the SMFPRMxx member of SYS1.PARMLIB or with the SETSMF z/OS operator command.

```
SYS(TYPE(115))
```

To record accounting information (record type 116) to SMF specify the following in the SMFPRMxx member of SYS1.PARMLIB or with the SETSMF z/OS operator command.

```
SYS(TYPE(116))
```

You can turn on or off the recording of accounting information at the queue or queue manager level using the ACCTQ parameter of the DEFINE QLOCAL,

DEFINE QMODEL, ALTER QLOCAL, ALTER QMODEL, DEFINE QMGR, or ALTER QMGR commands. See WebSphere MQ Script (MQSC) Command Reference for details of these commands.

To use the z/OS command SETSMF, either PROMPT(ALL) or PROMPT(LIST) must be specified in the SMFPRMxx member. See the *OS/390 MVS Initialization and Tuning Reference* and the *OS/390 MVS System Commands* manuals for more information.

You must also set the SMFSTAT and SMFACCT parameters to YES; this is described in “Using CSQ6SYSP” on page 28.

You can specify the interval at which WebSphere MQ collects statistics and accounting data in one of two ways:

- You can specify a value for STATIME in your system parameters (described in “Using CSQ6SYSP” on page 28).
- You can specify zero for STATIME and use the SMF global accounting interval (described in the *OS/390 MVS Initialization and Tuning Reference*).

SMF must be running before you can send data to it. For more information about SMF, see the *MVS System Management Facilities (SMF)* manual.

For the statistics and accounting data to be reset, at least one MQI call must be issued during the accounting interval.

## Allocating additional SMF buffers

When you invoke a trace, you must ensure that you allocate adequate SMF buffers. Specify SMF buffering on the VSAM BUFSP parameter of the access method services DEFINE CLUSTER statement. Specify CISZ(4096) and BUFSP(81920) on the DEFINE CLUSTER statement for each SMF VSAM data set.

If an SMF buffer shortage occurs, SMF rejects any trace records sent to it. WebSphere MQ sends a CSQW133I message to the z/OS console when this occurs. WebSphere MQ treats the error as temporary and remains active even though SMF data could be lost. When the shortage has been alleviated and trace recording has resumed, WebSphere MQ sends a CSQW123I message to the z/OS console.

## Reporting data in SMF

You can use the SMF program IFASMFDP to dump SMF records to a sequential data set so that they can be processed.

There are several ways to report on this data, for example:

- Write an application program to read and report information from the SMF data set. You can then tailor the report to fit your exact needs.
- Use Performance Reporter to process the records (see “Using Tivoli Decision Support for z/OS” on page 87).

## Using other products with WebSphere MQ

You can use other products to help you to improve the presentation of, or to augment statistics related to, performance and accounting.



## Using Resource Measurement Facility

Resource Management Facility (RMF™) is an IBM licensed program (program number 5685-029) that provides system-wide information on processor utilization, I/O activity, storage, and paging. You can use RMF to monitor the utilization of physical resources across the whole system dynamically. For more information, see the *MVS Resource Measurement Facility User's Guide*.

## Using Tivoli Decision Support for z/OS

You can use Tivoli® Decision Support for z/OS to interpret RMF and SMF records.

Tivoli Decision Support for z/OS is an IBM licensed program (program number 5698-A07) that enables you to manage the performance of your system by collecting performance data in a DB2 database and presenting the data in a variety of formats for use in systems management. Tivoli Decision Support for can generate graphic and tabular reports using systems management data it stores in its DB2 database. It includes an administration dialog, a reporting dialog, and a log collector, all of which interact with a standard DB2 database.

This is described in the *Tivoli Decision Support Administrator's Guide*.

## Using the CICS monitoring facility

The CICS monitoring facility provides performance information about each CICS transaction running. It can be used to investigate the resources used and the time spent processing transactions. For background information, see the *CICS Performance Guide* and the *CICS Customization Guide*.

## Investigating performance problems

Performance can be adversely affected by:

- Buffer pools that are an incorrect size
- Lack of real storage
- I/O contention for page sets or logs
- Log buffer thresholds that are set incorrectly
- Incorrect setting of the number of log buffers
- Large messages
- Units of recovery that last a long time, incorporating many messages for each syncpoint
- Messages that remain on a queue for a long time
- RACF auditing
- Unnecessary security checks
- Inefficient program design

When you analyze performance data, always start by looking at the overall system before you decide that you have a specific WebSphere MQ problem. Remember that almost all symptoms of reduced performance are magnified when there is contention. For example, if there is contention for DASD, transaction response times can increase. Also, the more transactions there are in the system, the greater the processor overhead and greater the demand for both virtual and real storage.

In such situations, the system shows heavy use of *all* its resources. However, the system is actually experiencing normal system stress, and this might be hiding the cause of a performance reduction. To find the cause of such a loss of performance, you must consider all items that might be affecting your active tasks.

## Investigating the overall system

Within WebSphere MQ, the performance problem is either increased response time or an unexpected and unexplained heavy use of resources. You should first check factors such as total processor usage, DASD activity, and paging. An IBM tool for this is resource management facility (RMF). In general, you need to look at the system in some detail to see why tasks are progressing slowly, or why a given resource is being heavily used.

Start by looking at general task activity, then focus on particular activities, such as specific tasks or a specific time interval.

Another possibility is that the system has limited real storage; therefore, because of paging interrupts, the tasks progress more slowly than expected.

## Investigating individual tasks

You can use the accounting trace to gather information about WebSphere MQ tasks. These trace records tell you a great deal about the activity that the task has performed, and about how much time the task spent suspended, waiting for latches. The trace record also includes information about how much DB2 and Coupling Facility activity was performed by the task.

This is described in “Interpreting WebSphere MQ accounting data” on page 104.

Long running units of work can be identified by the presence of message CSQR026I in the job log. This message indicates that a task has existed for more than three queue manager checkpoints and its log records have been shunted. For a description of log record shunting, see WebSphere MQ for z/OS Concepts and Planning Guide.

---

## Interpreting WebSphere MQ performance statistics

WebSphere MQ performance statistics are written as SMF type 115 records. Statistics records are produced periodically at a time interval specified by the STATIME parameter of the CSQ6SYSP system parameter module, or at the SMF global accounting interval if you specify zero for STATIME. The information provided in the SMF records comes from the following components of WebSphere MQ:

<b>Buffer manager</b>	Manages the buffer pools in virtual storage and the writing of pages to page sets as the buffer pools become full. Also manages the reading of pages from page sets.
<b>Coupling Facility manager</b>	Manages the interface with the Coupling Facility.
<b>Data manager</b>	Manages the links between messages and queues. It calls the buffer manager to process the pages with messages on them.
<b>DB2 manager</b>	Manages the interface with the DB2 database that is used as the shared repository.
<b>Lock manager</b>	Manages locks for WebSphere MQ for z/OS.

<b>Log manager</b>	Manages the writing of log records, which are essential for maintaining the integrity of the system if there is a back out request, or for recovery, if there is a system or media failure.
<b>Message manager</b>	Processes all WebSphere MQ API requests.
<b>Storage manager</b>	Manages storage for WebSphere MQ for z/OS, for example, storage pool allocation, expansion, and deallocation.

WebSphere MQ statistics can be collected for two subtypes:

- 1 System information, for example, related to the logs and storage.
- 2 Information about number of messages, buffer and paging information. Queue-sharing group information related to the Coupling Facility and DB2.

The subtype is specified in the SM115STF field (shown in Table 13).

## Layout of an SMF type 115 record

The standard layout for SMF records involves three parts:

### SMF header

Provides format, identification, and time and date information about the record itself.

### Self-defining section

Defines the location and size of the individual data records within the SMF record.

### Data records

The actual data from WebSphere MQ that you want to analyze.

For more information about SMF record formats, see the *MVS System Management Facilities (SMF)* manual.

## The SMF header

Table 13 shows the format of SMF record header (SM115).

Table 13. SMF record 115 header description

Offset: Dec	Offset: Hex	Type	Len	Name	Description	Example
0	0	Structure	28	SM115	SMF record header.	
0	0	Integer	2	SM115LEN	SMF record length.	14A0
2	2		2		Reserved.	
4	4	Integer	1	SM115FLG	System indicator.	5E
5	5	Integer	1	SM115RTY	Record type. The SMF record type, for WebSphere MQ statistics records this is always 115 (X'73').	73
6	6	Integer	4	SM115TME	Time when SMF moved record.	00355575
10	A	Integer	4	SM115DTE	Date when SMF moved record.	0100223F
14	E	Character	4	SM115SID	z/OS subsystem ID. Defines the z/OS subsystem on which the records were collected.	D4E5F4F1 (MV41)

Table 13. SMF record 115 header description (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description	Example
18	12	Character	4	SM115SSI	WebSphere MQ subsystem ID.	D4D8F0F7 (MQ07)
22	16	Integer	2	SM115STF	Record subtype.	0002
24	18	Character	3	SM115REL	WebSphere MQ version.	F6F0F0 (600)
27	1B		1		Reserved	
28	1C	Character	0	SM115END	End of SMF header and start of self-defining section.	

**Note:** The (hexadecimal) values in the right-hand column relate to Figure 18 on page 91.

## Self-defining sections

A self-defining section of a type 115 SMF record tells you where to find a statistics record, how long it is, and how many times that type of record is repeated (with different values). The self-defining sections follow after the header, at fixed offsets from the start of the SMF record. Each statistics record can be identified by an eye-catcher string.

Eight types of self-defining section are available to users for type 115 records. Each self-defining section points to statistics data related to one of the WebSphere MQ components. Table 14 summarizes the sources of the statistics, the eye-catcher strings, and the offsets of the self-defining sections from the start of the SMF record header.

Table 14. Offsets to self-defining sections. Offsets are from the start of the SMF record and are fixed for each type of statistics source.

Source of statistics	Record subtype (SM115STF)	Offset of self-defining section		Length of data	Eye-catcher of data
		Dec	Hex		
Storage manager	1	100	X'64'	X'48'	QSST
Log manager	1	116	X'74'	X'78'	QJST
Message manager	2	36	X'24'	X'30'	QMST
Data manager	2	44	X'2C'	X'50'	QIST
Buffer manager - one for each buffer pool	2	52	X'34'	X'68'	QPST
Lock manager	2	60	X'3C'	X'20'	QLST
DB2 manager	2	68	X'44'	X'1E0'	Q5ST
Coupling Facility manager	2	76	X'4C'	X'1008'	QEST

**Note:** Other self-defining sections refer to data for IBM use only.

Each self-defining section is two fullwords long and has this format:

```
sssssss1111nnnn
```

where:

sssssss Fullword containing the offset from the start of the SMF record.

**llll**            Halfword giving the length of this data record.  
**nnnn**            Halfword giving the number of data records in this SMF record.

Figure 18 shows an example of part of an SMF type 115 subtype 2 record. The numbers in the left-hand column represent the offset, in hexadecimal, from the start of the record. Each line corresponds to sixteen bytes of data, where each byte is two hexadecimal characters, for example 0C. The characters in the right-hand column represent the printable characters for each byte. Non-printable characters are shown by a period (.) character.

In this example, alternate fields in the SMF header are underlined to help you to see them; refer to Table 13 on page 89 to identify them. The self-defining sections for the message manager statistics data records (at the offset given in Table 14 on page 90) and buffer manager statistics are shown in **bold**.

000000	14A00000	5E730035	55750100	223FD4E5	*....;.....MV*
000010	<u>F4F1D4D8</u>	<u>F0F70002</u>	<u>F6F0F000</u>	<u>0000147C</u>	*41MQ07..600...0*
000020	00240001	<b>00000054</b>	<b>00300001</b>	00000084	*.....*
000030	00500001	<b>000000D4</b>	<b>00680004</b>	00000274	*.&....M.....*
000040	00200001	00000294	01E00001	00000474	*.....*
000050	10080001	D40F0030	D8D4E2E3	00000000	*...M...QMST...*

Figure 18. Part of an SMF record 115 showing the header and self-defining sections

The self-defining section for message manager statistics is located at offset X'20' from the start of the SMF record and contains this information:

- The offset of the message manager statistics is located X'00000054' bytes from the start of the SMF record.
- The message manager record is X'0030' bytes long.
- There is one record (X'0001').

Similarly, the buffer manager self-defining section at X'30' specifies that the offset to the buffer manager statistics is X'000000D4', is of length X'0068', and occurs X'0004' times.

**Note:** Always use offsets in the self-defining sections to locate the statistics records.

## Examples of SMF statistics records

Figure 19 on page 92 shows an example of part of the SMF record for subtype 1. Subtype 1 includes the storage manager and log manager statistics records. The SMF record header is shown underlined.

The self-defining section at offset X'64' refers to storage manager statistics and the self-defining section at offset X'74' refers to log manager statistics, both shown in **bold**.

The storage manager statistics record is located at offset X'0000011C' from the start of the header and is X'48' bytes long. There is one set of storage manager statistics, identified by the eye-catcher string QSST. The start of this statistics record is also shown in the example.

The log manager statistics record is located at offset X'00000164' from the start of the header and is X'78' bytes long. There is one set of log manager statistics, identified by the eye-catcher string QJST.

000000	02000000	5E730035	55750100	223FD4E5	*....;.....MV*
000010	<u>F4F1D4D8</u>	<u>F0F70001</u>	<u>F6F0F000</u>	000001DC	*41MQ07..600.....*
000020	00240001	00000000	00000000	00000000	*.....*
000030	00000000	00000000	00000000	0000007C	*.....@*
000040	00400001	000000BC	00600001	00000000	*.....-.....*
000050	00000000	00000000	00000000	00000000	*.....*
000060	00000000	<b>0000011C</b>	<b>00480001</b>	00000000	*.....*
000070	00000000	<b>00000164</b>	<b>00780001</b>	00000000	*.....*
000080	00000000	00000000	00000000	00000000	*.....*
.					
.					
000110	00000000	00000000	00000000	003C0048	*.....*
000120	D8E2E2E3	0000004F	00000003	00000002	*QSST... .....*

Figure 19. SMF record 115, subtype 1

Figure 20 on page 93 shows an example of part of the SMF record for subtype 2. Subtype 2 includes the statistics records for the message, data, buffer, lock, Coupling Facility, and DB2 managers. The SMF record header is shown underlined; the self-defining sections are shown alternately **bold** and *italic*.

- The self-defining section at offset X'24' refers to message manager statistics. The message manager statistics record is located at offset X'00000054' from the start of the header and is X'30' bytes long. There is one set of these statistics, identified by the eye-catcher string QMST.
- The self-defining section at offset X'2C' refers to data manager statistics. The data manager statistics record is located at offset X'00000084' from the start of the header and is X'50' bytes long. There is one set of these statistics, identified by the eye-catcher string QIST.
- The self-defining section at offset X'34' refers to buffer manager statistics. The buffer manager statistics record is located at offset X'000000D4' from the start of the header and is X'68' bytes long. There are four sets of these statistics, identified by the eye-catcher string QPST.
- The self-defining section at offset X'3C' refers to lock manager statistics. The lock manager statistics record is located at offset X'00000274' from the start of the header and is X'20' bytes long. There is one set of these statistics, identified by the eye-catcher string QLST.
- The self-defining section at offset X'44' refers to DB2 manager statistics. The DB2 manager statistics record is located at offset X'00000294' from the start of the header and is X'1E0' bytes long. There is one set of these statistics, identified by the eye-catcher string Q5ST.
- The self-defining section at offset X'4C' refers to Coupling Facility manager statistics. The Coupling Facility manager statistics record is located at offset X'00000474' from the start of the header and is X'1008' bytes long. There is one set of these statistics, identified by the eye-catcher string QEST.

```

000000 14A00000 5E730035 55750100 223FD4E5 *....;.....MV*
000010 F4F1D4D8 F0F70002 F6F0F000 0000147C *41MQ07..600...@*
000020 00240001 00000054 00300001 00000084 *.....*
000030 00500001 000000D4 00680004 00000274 *.&....M.....*
000040 00200001 00000294 01E00001 00000474 *.....*
000050 10080001 D40F0030 D8D4E2E3 00000000 *...M...QMST...*
000060 00000000 00000000 00000000 00000000 *.....*
000070 00000000 00000000 00000000 00000000 *.....*
000080 00000000 C90F0050 D8C9E2E3 00000000 *...I..&QIST...*
000090 00000001 00000000 00000025 00000003 *.....*
0000A0 00000000 0000002C 00000007 00000000 *.....*
0000B0 00000000 00000000 00000000 00000012 *.....*
0000C0 00000000 00000000 00000000 00000000 *.....*
0000D0 00000000 D70F0068 D8D7E2E3 00000000 *...P...QPST...*
0000E0 000007D0 000007BD 000007BD 00000037 *...}.....*
0000F0 00000000 0000001B 0000003B 00000000 *.....*
000100 00000000 00000000 00000000 00000000 *.....*
000110 0000001B 00000000 00000000 00000000 *.....*
000120 00000000 00000000 00000000 00000000 *.....*
000130 00000000 00000000 00000000 D70F0068 *...P...*
000140 D8D7E2E3 00000001 000007D0 000007CD *QPST.....}....*
.
.

```

Figure 20. SMF record 115, subtype 2

## Processing type 115 SMF records

You must process any data you collect from SMF to extract useful information. When you process the data, verify that the records are from WebSphere MQ and that they are the records you are expecting.

Validate the values of the following fields:

- SM115RTY, the SMF record number, must be X'73' (115)
- SM115STF, the record subtype, must be 0001 or 0002

There is a C sample program called CSQ4SMFD which prints the contents of SMF type 115 and 116 records. The program is provided as source in thlqual.SCSQC37S and in executable format in thlqual.SCSQLOAD. Sample JCL is provided in thlqual.SCSQPROC(CSQ4SMFJ).

## Storage manager data records

The format of the storage manager statistics record is described in assembler macro thlqual.SCSQMACS(CSQDQSST).

The data contains information about the number of fixed and variable storage pools that the queue manager has allocated, expanded, contracted, and deleted during the statistics interval, plus the number of GETMAIN, FREEMAIN, and STORAGE requests to z/OS, including a count of those that were unsuccessful. Additional information includes a count of the number of times the short-on-storage condition was detected and a count of the number of abends that occurred as a result of that condition.

## Log manager data records

The format of the log manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQJST)`.

In the statistics, these counts are important:

1. The total number of log write requests:

$$N_{\text{logwrite}} = \text{QJSTWRW} + \text{QJSTWRNW} + \text{QJSTWRF}$$

2. The total number of log read requests:

$$N_{\text{logread}} = \text{QJSTRBUF} + \text{QJSTRACT} + \text{QJSTRARH}$$

The problem symptoms that can be examined using log manager statistics are described in Table 15.

*Table 15. Problem symptoms that can be examined using log manager statistics*

<p><b>Symptom 1</b> QJSTWTB is nonzero.</p> <p><b>Reason</b> Tasks are being suspended while the in-storage buffer is being written to the active log.  There might be problems writing to the active log.  The OUTBUFF parameter within CSQ6LOGP is too small.</p> <p><b>Action</b> Investigate the problems writing to the active log.  Increase the value of the OUTBUFF parameter within CSQ6LOGP.</p>
<p><b>Symptom 2</b> The ratio: <math>\text{QJSTWTL}/N_{\text{logread}}</math> is greater than 1%.</p> <p><b>Reason</b> Log reads were initiated that had to read from an archive log, but WebSphere MQ could not allocate a data set because MAXRTU data sets were already allocated.</p> <p><b>Action</b> Increase MAXRTU.</p>
<p><b>Symptom 3</b> The ratio: <math>\text{QJSTRARH}/N_{\text{logread}}</math> is larger than normal.</p> <p><b>Reason</b> Most log read requests should come from the output buffer or the active log. To satisfy requests for back out, unit-of-recovery records are read from the in-storage buffer, the active log, and the archived logs.  A long-running unit of recovery, extending over a period of many minutes, might have log records spread across many different logs. This degrades performance because extra work has to be done to recover the log records.</p> <p><b>Action</b> Change the application to reduce the length of a unit of recovery. Also, consider increasing the size of the active log to reduce the possibility of a single unit of recovery being spread out over more than one log.</p> <p><b>Other pointers</b> The ratio <math>N_{\text{logread}}/N_{\text{logwrite}}</math> gives an indication of how much work has to be backed out.</p>



Table 15. Problem symptoms that can be examined using log manager statistics (continued)

**Symptom 4**

QJSTLLCP is more than 10 an hour.

**Reason**

On a busy system you would expect to see typically 10 checkpoints an hour. If the QJSTLLCP value is larger than this, it indicates a problem in the setup of the queue manager.

The most likely reason for this is that the LOGLOAD parameter in CSQ6SYSP is too small. The other event that causes a checkpoint is when an active log fills up and switches to the next active log data set. If your logs are too small, this can cause frequent checkpoints.

**Action**

Increase the LOGLOAD parameter, or increase the size of your log data sets as required.

**Note:** In the first set of statistics produced after system startup, there might be significant log activity due to the resolution of in-flight units of recovery.

## Message manager data records

The format of the message manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQMST)`.

The data gives you counts of different WebSphere MQ API requests.

## Data manager data records

The format of the data manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQIST)`.

The data gives you counts of different object requests.

## Buffer manager data records

The format of the buffer manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQPST)`.

**Note:** If you have defined a buffer pool, but not used it, no values are set so the buffer manager statistics record does not contain any data.

When interpreting the statistics, you are recommended to consider the following factors because the values of these fields can be used to improve the performance of your system:

1. If QPSTSOS, QPSTSTLA, or QPSTDMC is greater than zero, you should either increase the size of the buffer pool or reallocate the page sets to different buffer pools.
  - QPSTSOS is the number of times that there were no buffers available for page get requests. If QPSTSOS ever becomes nonzero, it shows that WebSphere MQ is under severe stress. The buffer pool size should be

significantly increased. If increasing the buffer pool size does not make the value of QPSTSOS zero, there might be I/O contention on the DASD page sets.

- QPSTDMC is the number of updates that were performed synchronously because there was either more than 95% of the pages in the buffer pool waiting for write I/O, or there was less than 5% of the buffer pool available for read requests. If this number is not zero, the buffer pool might be too small and should be enlarged. If increasing the buffer pool size does not reduce QPSTDMC to zero, there might be I/O contention on the DASD page sets.
- QPSTIMW is a count of the number of times pages were written out synchronously. If QPSTDMC is zero, QPSTIMW is the number of times pages were found on the queue waiting for write I/O that had been there for at least two checkpoints.

2. For buffer pool zero and buffer pools that contain short-lived messages:

- QPSTDWT should be zero, and the percentage  $QPSTCBSL/QPSTNBUF$  should be greater than 15%.

QPSTDWT is the number of times the asynchronous write processor was started because there was either more than 85% of the pages in the buffer pool waiting for write I/O, or there was less than 15% of the buffer pool available for read requests. Increasing the buffer pool size should reduce this value. If it does not, the pattern of access is one of long delays between puts and gets.

- QPSTTPW might be greater than zero due to checkpointing activity.
- QPSTRIO should be zero unless messages are being read from a page set after the queue manager is restarted.

The ratio of QPSTRIO to QPSTGETP shows the efficiency of page retrieval within the buffer pool. Increasing the buffer pool size should decrease this ratio and, therefore, increase the page retrieval efficiency. If this does not happen, it indicates that pages are not being frequently reaccessed. This implies a transaction pattern where there is a long delay between messages being put and then subsequently retrieved.

The ratio of QPSTGETN to QPSTGETP indicates the number of times an empty page, as opposed to a non-empty page, has been requested. This ratio is more an indication of transaction pattern, than a value that can be used to tune the system.

- If QPSTSTL has a value greater than zero, this indicates that pages that have not been used before are now being used. This might be caused by an increased message rate, messages not being processed as fast as they were previously (leading to a buildup of messages), or larger messages being used.

QPSTSTL is a count of the number of times a page access request did not find the page already in the buffer pool. Again, the lower the ratio of QPSTSTL to  $(QPSTGETP + QPSTGETN)$  is, the higher the page retrieval efficiency. Increasing the buffer pool size should decrease this ratio but, if it does not, it is an indication that there are long delays between puts and gets.

- You are recommended to have sufficient buffers to handle your peak message rate.

3. For buffer pools with long-lived messages, where there are more messages than can fit into the buffer pool:

- $(QPSTRIO+QPSTWIO)/\text{Statistics interval}$  is the I/O rate to page sets. If this value is high, you should consider using multiple page sets on different volumes to allow I/O to be carried out in parallel.

The higher the ratio of QPSTSTW to QPSTWIO, the better the efficiency of the asynchronous write processor. You can increase this ratio, and therefore the efficiency of the asynchronous write processor, by increasing the buffer pool size.

- Over the period of time that the messages are processed (for example, if messages are written to a queue during the day and processed overnight) the number of read I/Os (QPSTRIO) should be approximately the total number of pages written (QPSTTPW). This shows that one page is read for every page written.

If QPSTRIO is much larger than QPSTTPW, this shows that pages are being read in multiple times. This might be a result of the application using **MQGET** by *MsgId* or *CorrelId* when the queue is not indexed, or browsing messages on the queue using `get next`.

The following actions might relieve this problem:

- a. Increase the size of the buffer pool so that there are enough pages to hold the queue, in addition to any changed pages.
- b. Use the `INDXTYPE` queue attribute, which allows a queue to be indexed by *MsgId* or *CorrelId* and eliminates the need for a sequential scan of the queue.
- c. Change the design of the application to eliminate the use of **MQGET** with *MsgId* or *CorrelId*, or the `get next` with `browse` option.

**Note:** Applications using long-lived messages typically process the first available message and do not use **MQGET** with *MsgId* or *CorrelId*, and they might browse only the first available message.

- d. Move page sets to a different buffer pool to reduce contention between messages from different applications.

## Managing your buffer pools

To manage your buffer pools efficiently, you must consider the factors that affect the buffer pool I/O operations and also the statistics associated with the buffer pools.

The following factors affect buffer pool I/O operations.

- If a page containing the required data is not found in the buffer pool, it is read in synchronously to an available buffer from its DASD page set.
- Whenever a page is updated, it is put on an internal queue of pages to be (potentially) written out to DASD. This means that the buffer used by that page is unavailable for use by any other page until the buffer has been written to DASD.
- If the number of pages queued to be written to DASD exceeds 85% of the total number of buffers in the pool, an asynchronous write processor is started to put the buffers to DASD.

Similarly, should the number of buffers available for page get requests become less than 15% of the total number of buffers in the pool, the asynchronous write processor is started to perform the write I/O operations.

The write processor stops when the number of pages queued to be written to DASD has fallen to 75% of the total number of buffer in the pool.

- If the number of pages queued for writing to DASD exceeds 95% of the total number of buffers in the pool, all updates result in a synchronous write of the page to DASD.

Similarly, if the number of buffers available for page get requests becomes less than 5% of the total number of buffers in the pool, all updates result in a synchronous write of the page to DASD.

- If the number of buffers available for page get requests ever reaches zero, a transaction that encounters this condition is suspended until the asynchronous write processor has finished.
- If a page is frequently updated, the page spends most of its time on the queue of pages waiting to be written to DASD. Because this queue is in least recently used order, it is possible that a frequently updated page placed on this least recently used queue is never written out to DASD. For this reason, at the time of update, if the page is found to have been waiting on the write to DASD queue for at least 2 checkpoints, it is synchronously written to DASD. Updating occurs at checkpoint time.

The aim of this algorithm is to maximize the time pages spend in buffer pool memory while allowing the system to function if the system load puts the buffer pool usage under stress.

## Lock manager data records

The format of the lock manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQLST)`.

The data contains information about the following:

- The number of lock get requests and lock release requests.
- The number of times a lock get request determined that the requested lock was already held.

## DB2 manager data records

The format of the DB2 manager statistics record is described in the following table and in assembler macro `thlqual.SCSQMACS(CSQDQ5ST)` and C header file `thlqual.SCSQC370(CSQDSMFC)`. The field names in C are all in lower case, for example `q5st`, `q5stid`.

If the queue manager was not started as a member of a queue-sharing group, no data is recorded in this record.

*Table 16. DB2 statistics record (Q5ST)*

Offset: Dec	Offset: Hex	Type	Len	Name	Description
0	0	Structure	668	Q5ST	DB2 manager statistics
0	0	Bitstring	2	Q5STID	Control block identifier
2	2	Integer	2	Q5STLL	Control block length
4	4	Character	4	Q5STEYEC	Control block eye catcher
8	8	Character	660	Q5STZERO	QMST part cleared on occasion
8	8	Integer	4	NUMTASK	Number of server tasks
12	C	Integer	4	ACTTASK	Number of active server tasks
16	10	Integer	4	CONNCNT	Number of connect requests
20	14	Integer	4	DISCCNT	Number of disconnect requests

Table 16. DB2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
24	18	Integer	4	DHIGMAX	Max. request queue depth
28	1C	Integer	4	ABNDCNT	Number of DB2SRV task abends
32	20	Integer	4	REQUCNT	Number of requests queued
36	24	Integer	4	DEADCNT	Number of deadlock timeouts
40	28	Integer	4	DELECNT	Number of delete requests
44	2C	Integer	4	LISTCNT	Number of list requests
48	30	Integer	4	READCNT	Number of read requests
52	34	Integer	4	UPDTCNT	Number of update requests
56	38	Integer	4	WRITCNT	Number of write requests
60	3C	Integer	4	SCSSEL	SCST (shared-channel-status) selects
64	40	Integer	4	SCSINS	SCST inserts
68	44	Integer	4	SCSUPD	SCST updates
72	48	Integer	4	SCSDEL	SCST deletes
76	4C	Integer	4	SSKSEL	SSKT (shared-sync-key) selects
80	50	Integer	4	SSKINS	SSKT inserts
84	54	Integer	4	SSKDEL	SSKT deletes
88	58	Integer	4	SCSBFTS	SCST number of times buffer too small
92	5C	Integer	4	SCSMAXR	SCST maximum rows on query
96	60	Integer	4	* (2)	Reserved
104	68	Character	8	DELETCUW	Cumulative STCK difference - Thread delete
112	70	Character	8	DELETMXW	Maximum STCK difference - Thread delete
120	78	Character	8	DELESCUW	Cumulative STCK difference - SQL delete
128	80	Character	8	DELESMXW	Maximum STCK difference - SQL delete
136	88	Character	8	LISTTCUW	Cumulative STCK difference - Thread list
144	90	Character	8	LISTTMXW	Maximum STCK difference - Thread list
152	98	Character	8	LISTSCUW	Cumulative STCK difference - SQL list
160	A0	Character	8	LISTSMXW	Maximum STCK difference - SQL list
168	A8	Character	8	READTCUW	Cumulative STCK difference - Thread read
176	B0	Character	8	READTMXW	Maximum STCK difference - Thread read
184	B8	Character	8	READSCUW	Cumulative STCK difference - SQL read
192	C0	Character	8	READSMXW	Maximum STCK difference - SQL read

Table 16. DB2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
200	C8	Character	8	UPDTTCUW	Cumulative STCK difference - Thread update
208	D0	Character	8	UPDTTMXW	Maximum STCK difference - Thread update
216	D8	Character	8	UPDTSCUW	Cumulative STCK difference - SQL update
224	E0	Character	8	UPDTSMXW	Maximum STCK difference - SQL update
232	E8	Character	8	WRITTCUW	Cumulative STCK difference - Thread write
240	F0	Character	8	WRITTMXW	Maximum STCK difference - Thread write
248	F8	Character	8	WRITSCUW	Cumulative STCK difference - SQL write
256	100	Character	8	WRITSMXW	Maximum STCK difference - SQL write
264	108	Character	8	SCSSTCUW	Cumulative STCK difference - Thread select
272	110	Character	8	SCSSTMXW	Maximum STCK difference - Thread select
280	118	Character	8	SCSSCUW	Cumulative STCK difference - SQL select
288	120	Character	8	SCSSMXW	Maximum STCK difference - SQL select
296	128	Character	8	SCSITCUW	Cumulative STCK difference - Thread insert
304	130	Character	8	SCSITMXW	Maximum STCK difference - Thread insert
312	138	Character	8	SCSISCUW	Cumulative STCK difference - SQL insert
320	140	Character	8	SCSISMXW	Maximum STCK difference - SQL insert
328	148	Character	8	SCSUTCUW	Cumulative STCK difference - Thread update
336	150	Character	8	SCSUTMXW	Maximum STCK difference - Thread update
344	158	Character	8	SCSUSCUW	Cumulative STCK difference - SQL update
352	160	Character	8	SCSUSMXW	Maximum STCK difference - SQL update
360	168	Character	8	SCSDTCUW	Cumulative STCK difference - Thread delete
368	170	Character	8	SCSDTMXW	Maximum STCK difference - Thread delete
376	178	Character	8	SCSDSCUW	Cumulative STCK difference - SQL delete

Table 16. DB2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
384	180	Character	8	SCSDSMXW	Maximum STCK difference - SQL delete
392	188	Character	8	SSKSTCUW	Cumulative STCK difference - Thread select
400	190	Character	8	SSKSTMXW	Maximum STCK difference - Thread select
408	198	Character	8	SSKSSCUW	Cumulative STCK difference - SQL select
416	1A0	Character	8	SSKSSMXW	Maximum STCK difference - SQL select
424	1A8	Character	8	SSKITCUW	Cumulative STCK difference - Thread insert
432	1B0	Character	8	SSKITMXW	Maximum STCK difference - Thread insert
440	1B8	Character	8	SSKISCUW	Cumulative STCK difference - SQL insert
448	1C0	Character	8	SSKISMXW	Maximum STCK difference - SQL insert
456	1C8	Character	8	SSKDTCUW	Cumulative STCK difference - Thread delete
464	1D0	Character	8	SSKDTMXW	Maximum STCK difference - Thread delete
472	1D8	Character	8	SSKDSCUW	Cumulative STCK difference - SQL delete
480	1E0	Character	8	SSKDSMXW	Maximum STCK difference - SQL delete
488	1E8	Integer	4	LMSSEL	Number of DB2 BLOB read requests
492	1EC	Integer	4	LMSINS	Number of DB2 BLOB insert requests
496	1F0	Integer	4	LMSUPD	Number of DB2 BLOB update requests
500	1F4	Integer	4	LMSDEL	Number of DB2 BLOB delete requests
504	1F8	Integer	4	LMSLIS	Number of DB2 BLOB list requests
508	IFC	64-bit integer	8	LMSSTCUW	Total elapsed time for all thread read BLOB requests
516	204	64-bit integer	8	LMSSTMXW	Maximum elapsed time for a thread read BLOB request
524	20C	64-bit integer	8	LMSSSCUW	Total elapsed time for all SQL read BLOB requests
532	214	64-bit integer	8	LMSSSMXW	Maximum elapsed time for an SQL read BLOB request
540	21C	64-bit integer	8	LMSITCUW	Total elapsed time for all thread insert BLOB requests
548	224	64-bit integer	8	LMSITMXW	Maximum elapsed time for a thread insert BLOB request
556	22C	64-bit integer	8	LMSISCUW	Total elapsed time for all SQL insert BLOB requests

Table 16. DB2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
564	234	64-bit integer	8	LMSISMXW	Maximum elapsed time for an SQL insert BLOB request
572	23C	64-bit integer	8	LMSUTCW	Total elapsed time for all thread update BLOB requests
580	244	64-bit integer	8	LMSUTMXW	Maximum elapsed time for a thread update BLOB request
588	24C	64-bit integer	8	LMSUSCUW	Total elapsed time for all SQL update BLOB requests
596	254	64-bit integer	8	LMSUSMXW	Maximum elapsed time for an SQL update BLOB request
604	25C	64-bit integer	8	LMSDTCW	Total elapsed time for all thread delete BLOB requests
612	264	64-bit integer	8	LMSDTMXW	Maximum elapsed time for a thread delete BLOB request
620	26C	64-bit integer	8	LMSDSCW	Total elapsed time for all SQL delete BLOB requests
628	274	64-bit integer	8	LMSDSMXW	Maximum elapsed time for an SQL delete BLOB request
636	27C	64-bit integer	8	LMSLTCW	Total elapsed time for all thread list BLOB requests
644	284	64-bit integer	8	LMSLTMXW	Maximum elapsed time for a thread list BLOB request
652	28C	64-bit integer	8	LMSLSCW	Total elapsed time for all SQL list BLOB requests
660	294	64-bit integer	8	LMSLSMXW	Maximum elapsed time for an SQL list BLOB request

The data contains counts for each request type that the DB2 resource manager supports. For these request types, maximum and cumulative elapsed times are kept for the following:

- The time spent in the DB2 resource manager as a whole (called the thread time).
- The time that was spent performing the RRSF and SQL parts of the request (a subset of the thread time called the SQL time).

Information is also provided for:

- The number of server tasks attached.
- The maximum overall request depth against any of the server tasks.
- The number of times any of the server task requests terminated abnormally.

If the abnormal termination count is not zero, a requeue count is provided indicating the number of queued requests that were requeued to other server tasks as a result of the abnormal termination.

If the average thread time is significantly greater than the average SQL time, this might indicate that thread requests are spending an excessive amount of time waiting for a server task to process the SQL part of the request. If this is the case,



examine the DHIGMAX field and, if the value is greater than one, consider increasing the number of DB2 server tasks specified in the QSGDATA parameter of the CSQ6SYSP system parameter macro.

## Coupling Facility manager data records

The format of the Coupling Facility manager statistics record is described in the following table and in assembler macro thlqual.SCSQMACS(CSQDQEST) and C header file thlqual.SCSQC370(CSQDSMFC). The field names in C are all in lower case, for example qest, qestid.

If the queue manager was not started as a member of a queue-sharing group, no data is recorded in this record.

Table 17. Coupling facility statistics record (QEST)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
0	0	Structure	4104	QEST	CF manager statistics
0	0	Bitstring	2	QESTID	Control block identifier
2	2	Integer	2	QESTLL	Control block length
4	4	Character	4	QESTYEC	Control block eye catcher
8	8	Character	4096	QESTZERO	QEST part cleared on occasion
8	8	Character	64	QESTSTUC (0:63)	Array (one entry per structure)
8	8	Character	12	QESTSTR	Structure name
20	14	Integer	4	QESTSTRN	Structure number
24	18	Integer	4	QESTCSEC	Number of IXLLSTE calls
28	1C	Integer	4	QESTCMEC	Number of IXLLSTM calls
32	20	Character	8	QESTSSTC	Time spent doing IXLLSTE calls
40	28	Character	8	QESTMSTC	Time spent doing IXLLSTM calls
48	30	Integer	4	QESTRSEC	Number of IXLLSTE redrives
52	34	Integer	4	QESTRMEC	Number of IXLLSTM redrives
56	38	Integer	4	QESTSFUL	Number of structure fulls
60	3C	Integer	4	QESTMNUS	Maximum number of entries in use
64	40	Integer	4	QESTMLUS	Maximum number of elements in use
68	44	Character	4	*	Reserved
4104	1008	Character	0	*	End of control block

The data contains information for each Coupling Facility list structure, including the CSQ\_ADMIN structure, that the queue manager could connect to during the statistics interval. The information for each structure includes the following:

- The number of and cumulative elapsed times for IXLLSTE and IXLLSTM requests.
- The number of times a request had to be retried because of a timeout.
- The number of times a 'structure full' condition occurred.

---

## Interpreting WebSphere MQ accounting data

WebSphere MQ accounting data is written as SMF type 116 records.

WebSphere MQ accounting information can be collected for three subtypes:

- 0 Message manager accounting records (how much CPU was spent processing WebSphere MQ API calls and the number of **MQPUT** and **MQGET** calls). This information is produced when a named task disconnects from WebSphere MQ, and so the information contained within the record might cover many hours.
- 1 Accounting data for each task, at thread and queue level.
- 2 Additional queue-level accounting data (if the task used more queues than could fit in the subtype 1 record).

Subtype 0 is produced with trace class 1; subtypes 1 and 2 are produced with trace class 3.

## Layout of an SMF type 116 record

The standard layout for SMF records involves three parts:

### SMF header

Provides format, identification, and time and date information about the record itself.

### Self-defining section

Defines the location and size of the individual data records within the SMF record.

### Data records

The actual data from WebSphere MQ that you want to analyze.

For more information about SMF record formats, see the *MVS System Management Facilities (SMF)* manual.

## The SMF header

Table 18 shows the format of SMF record header (SM116).

Table 18. SMF record header description

Offset: Dec	Offset: Hex	Type	Len	Name	Description	Example
0	0	Structure	28	SM116	SMF record header.	
0	0	Integer	2	SM116LEN	SMF record length.	01A4
2	2		2		Reserved.	
4	4	Integer	1	SM116FLG	System indicator.	5E

Table 18. SMF record header description (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description	Example
5	5	Integer	1	SM116RTY	Record type. The SMF record type, for WebSphere MQ accounting records this is always 116 (X'74').	74
6	6	Integer	4	SM116TME	Time when SMF moved record.	00356124
10	A	Integer	4	SM116DTE	Date when SMF moved record.	0100223F
14	E	Character	4	SM116SID	z/OS subsystem ID. Defines the z/OS subsystem on which the records were collected.	D4E5F4F1 (MV41)
18	12	Character	4	SM116SSI	WebSphere MQ subsystem ID.	D4D8F0F7 (MQ07)
22	16	Integer	2	SM116STF	Record subtype.	0000
24	18	Character	3	SM116REL	WebSphere MQ version.	F6F0F0 (600)
27	1B		1		Reserved.	
28	1C	Character	0	SM116END	End of SMF header and start of self-defining section.	

**Note:** The (hexadecimal) values in the right-hand column relate to Figure 21 on page 107.

## Self-defining sections

A self-defining section of an SMF record tells you where to find an accounting record, how long it is, and how many times that type of record is repeated (with different values). The self-defining sections follow the header, at a fixed offset from the start of the SMF record.

Each self-defining section points to accounting related data. Table 19 summarizes the offsets from the start of the SMF record header.

Table 19. Offsets to self-defining sections. Offsets are from the start of the SMF record and are fixed for each type of accounting source.

Record subtype (SMF116STF)	Source of accounting data	Offset of self-defining section		See...
		Dec	Hex	
All	Common header	28	X'1C'	"Common WebSphere MQ SMF header" on page 107
0	Message manager	44	X'2C'	"Message manager data records" on page 109
1	Thread identification record	36	X'24'	"Thread-level and queue-level data records" on page 110

1	Thread-level accounting	44	X'2C'	"Thread-level and queue-level data records" on page 110
1	Queue-level accounting	52	X'34'	"Thread-level and queue-level data records" on page 110. This section is present only if the WTASWQCT field in the task-related information (WTAS) structure (Table 23 on page 111) is non-zero.

2	Thread identification record	36	X'24'	"Thread-level and queue-level data records" on page 110
2	Queue-level accounting	44	X'2C'	"Thread-level and queue-level data records" on page 110

**Note:** Other self-defining sections refer to data for IBM use only.

Each self-defining section is two fullwords long and has this format:

sssssss1111nnnn

where:

**sssssss**

Fullword containing the offset from start of the SMF record.

**1111** Halfword giving the length of this data record.

**nnnn** Halfword giving the number of data records in this SMF record.

Figure 21 on page 107 shows an example of part of an SMF type 116 record. The numbers in the left-hand column represent the offset, in hexadecimal, from the start of the record. Each line corresponds to sixteen bytes of data, where each byte is two hexadecimal characters, for example 0C. The characters in the right-hand column represent the printable characters for each byte. Non-printable characters are shown by a period (.) character.

In this example, alternate fields in the SMF header are underlined to help you to see them; refer to Table 18 on page 104 to identify them. The self defining section for one of the message manager accounting data records (at the offset given in Table 19 on page 105) is shown in **bold**.

```

000000 01A40000 5E740035 61240100 223FD4E5 *....;.../.....MV*
000000 F4F1D4D8 F0F70000 F6F0F000 00000134 *41MQ07..600....*
000000 00700001 00000054 00B00001 00000104 *.....*
000000 00300001 00000000 00000000 00000000 *.....*
000000 00000000 00000000 00000000 00000000 *.....*

```

Figure 21. Part of an SMF record 116 showing the header and self-defining sections

The self-defining section for the type of message manager accounting data is located at offset X'2C' from the start of the SMF record and contains this information:

- The offset of the message manager accounting data is located X'00000104' bytes from the start of the SMF record.
- This message manager record is X'0030' bytes long.
- There is one record (X'0001').

**Note:** Always use offsets in the self-defining sections to locate the accounting records.

### Processing type 116 SMF records

Any accounting data you collect from SMF must be processed to extract useful information. When you process the data, verify that the records are from WebSphere MQ and that they are the records you are expecting.

Validate the value of the following fields:

- SM116RTY, the SMF record number = X'74' (116)
- SM116STF, the record subtype, must be 0000, 0001, or 0002

There is a C sample program called CSQ4SMFD which prints the contents of SMF type 115 and 116 records. The program is provided as source in thlqual.SCSQC37S and in executable format in thlqual.SCSQLOAD. Sample JCL is provided in thlqual.SCSQPROC.

### Common WebSphere MQ SMF header

The format of this record is described in Table 20 and in assembler macros thlqual.SCSQMACS(CSQDQWHS) and thlqual.SCSQMACS(CSQDQWHC), and C header file thlqual.SCSQC370(CSQDSMFC). The field names in C are all in lower case, for example qwhs, qwhsnsda..

The QWHS data includes the subsystem name. For subtype 1 records, it also shows whether there are queue-level accounting records present. If the QWHSNSDA field is 3 or less, there are not, and the corresponding self-defining section (at offset X'34') is not set.

The QWHC data gives you information about the user (for example, the user ID (QWHCAID) and the type of application (QWHCATYP)).

Table 20. Structure of the common WebSphere MQ SMF header record QWHS

Offset: Dec	Offset: Hex	Type	Length	Name	Description
0	0	Structure	128	QWHS	

Table 20. Structure of the common WebSphere MQ SMF header record QWHS (continued)

Offset: Dec	Offset: Hex	Type	Length	Name	Description
0	0		6		Reserved
6	6	Character	1	QWHSNSDA	Number of self defining sections in the SMF records
7	7		5		Reserved
12	C	Character	4	QWHSSSID	Subsystem name
16	10		24		Reserved
40	28	Character	8	QWHCAID	User ID associated with the OS/390 job
48	30	Character	12	QWHCCV	Thread cross reference
60	3C	Character	8	QWHCCN	Connection name
68	44		8		Reserved
76	4C	Character	8	QWHCOPID	User ID associated with the transaction
84	54	Integer	4	QWHCATYP	Type of connecting system (1=CICS, 2=Batch or TSO, 3=IMS control region, 4=IMS MPP or BMP, 5=Command server, 6=Channel initiator, 7=RRS Batch)
88	58	Character	22	QWHCTOKN	Accounting token set to the OS/390 accounting information for the user
110	6E	Character	16	QWHCNID	Network identifier
126	7E		2		Reserved

## Thread cross reference data

The interpretation of the data in the thread cross reference (QWHCCV) field varies. This depends on what the data relates to:

- CICS connections (QWHCATYP=1) – see Table 21
- IMS connections (QWHCATYP=3 or 4) – see Table 22 on page 109
- Batch connections (QWHCATYP=2 or 7) – this field consists of binary zeros
- Others – no meaningful data

Table 21. Structure of the thread cross reference for a CICS system

Offset: Dec	Offset: Hex	Type	Length	Description
48	30	Character	4	CICS thread number.
52	34	Character	4	CICS transaction name.
56	38	Integer	4	CICS task number.

Some entries contain blank characters. These apply to the task, rather than to a specific transaction.

Table 22. Structure of the thread cross reference for an IMS system

Offset: Dec	Offset: Hex	Type	Length	Description
48	30	Character	4	IMS partition specification table (PST) region identifier.
52	34	Character	8	IMS program specification block (PSB) name.

## Message manager data records

The message manager is the component of WebSphere MQ that processes all API requests. The format of the message manager accounting records is described in assembler macro `thlqual.SCSQMACS(CSQDQMAC)`.

The QMAC data gives you information about the CPU time spent processing WebSphere MQ calls, and counts of the number of **MQPUT** and **MQGET** requests for messages of different sizes.

**Note:** A single IMS application might write two SMF records. In this case, the figures from both records should be added to provide the correct totals for the IMS application.

### Records containing zero CPU time

Records are sometimes produced that contain zero CPU time in the QMACCPUT field. These records occur when long running tasks identified to WebSphere MQ either terminate or are prompted to output accounting records by accounting trace being stopped. Such tasks exist in the CICS adapter and in the channel initiator (for distributed queuing). The number of these tasks with zero CPU time depends upon how much activity there has been in the system:

- For the CICS adapter, this can result in up to nine records with zero CPU time.
- For the channel initiator, the number of records with zero CPU time can be up to the sum of Adapters + Dispatchers + 6, as defined in the queue manager attributes.

These records reflect the amount of work done under the task, and can be ignored.

## Sample subtype zero accounting record

Figure 22 on page 110 shows a type 116, subtype zero SMF record. In this figure, the SMF record header and the QMAC accounting data record are underlined. The self-defining sections are in bold.

000000	01A40000	5E740035	61240100	223FD4E5	*....;.../.....MV*
000010	F4F1D4D8	F0F70000	F6F0F000	00000134	*41MQ07..600....*
000020	00700001	00000054	00B00001	00000104	*.....*
000030	00300001	00000000	00000000	00000000	*.....*
000040	00000000	00000000	00000000	00000000	*.....*
000050	00000000	B478AB43	9C6C2280	B478AB47	*.....%.....*
000060	9DB47E02	00000000	04C0F631	00000001	*..=.....{6.....*
000070	9880E72D	00000000	014D9540	00000000	*..X.....(. ....*
000080	08480C80	00000010	40404040	40404040	*.....*
000090	00000000	00000000	00000051	00000000	*.....*
0000A0	00000000	00000000	00000000	00000000	*.....*
0000B0	00000000	00000000	00000000	00000000	*.....*
0000C0	00000000	00000000	00000000	00000000	*.....*
0000D0	00000000	00000000	00000000	00000000	*.....*
0000E0	00000000	00000000	00000000	00000000	*.....*
0000F0	00000000	00000000	00000000	00000000	*.....*
000100	00000000	D4140030	D8D4C1C3	00000000	*...M...QMAC...*
000110	689C738D	00000050	00000000	00000050	*.....&.....&*
000120	0000000A	00000000	00000000	00000000	*.....*
000130	00000000	0024011A	00030710	02DAACF0	*.....0*

Figure 22. Example SMF type 116, subtype zero record

## Thread-level and queue-level data records

Thread level accounting records are collected for each task using WebSphere MQ. For each task, a thread-level accounting data record is written to the SMF when the task finishes. For a long running task, data is also written at the statistics interval set by the STATIME parameter of the CSQ6SYSP system parameter macro (or by the system SMF statistics broadcast), provided that the task was running the previous time statistics were gathered. In addition, accounting information is gathered about each queue that the task opens. A queue-level accounting record is written for each queue that the task has used since the thread-level accounting record was last written.

Thread-level and queue-level accounting records are produced if you specify class 3 when you start the accounting trace.

The thread level accounting information is written to an SMF type 116, subtype 1 record, and is followed by queue-level records. If the task opened many queues, further queue information is written to one or more SMF type 116 subtype 2 records. A thread identification control block is included in each subtype 1 and 2 record to enable you to relate each record to the correct task. Typically, the maximum number of queue-level records in each SMF record is about 45.

The format of the thread-level accounting record is described in assembler macro `thlqual.SCSQMACS(CSQDWTAS)` and in Table 23 on page 111. The format of the queue-level accounting record is described in assembler macro `thlqual.SCSQMACS(CSQDWQ)` and in Table 24 on page 113. The format of the thread identification record is described in assembler macro `thlqual.SCSQMACS(CSQDWTID)` and in Table 25 on page 118. All these records are also described in C header file `thlqual.SCSQC370(CSQDSMFC)`. The field names in C are all in lower case, for example `wtas`, `wtasshex`.



Table 23. Layout of the task-related information (WTAS) structure

Offset: Dec	Offset: Hex	Type	Length	Name	Description
0	0	Structure	712	WTAS	
0	0	Integer	2	WTASSEX	Hex ID of block
2	2	Integer	2	WTASLEN	Length of block
4	4	Character	4	WTASEYEC	Eye catcher
8	8	Character	16	WTASCORR	Correlator identifier
8	8	Character	8	WTASSTRT	Time WTAS allocated
16	10	Character	8	WTASHASH	Reserved
16	10	Integer	4	WTASMTHR	Reserved
20	14	Integer	4	WTASWTAS	Reserved
24	18	Character	8	WTASLATC	Reserved
32	20	Integer	4	WTASHSHI	Reserved
36	24	Address	4	*	Reserved
40	28	Bitstring	4	*	Reserved
44	2C	Character	4	*	Reserved
48	30	Character	384	WTASTHST	Thread statistics
48	30	Character	8	*	Reserved
304	130	Integer	4	*	Reserved
<b>Non-queue 'other' statistics</b>					
432	1B0	Character	8	WTASOTET	Other MQI calls elapsed time
440	1B8	Character	8	WTASOTCT	Other MQI calls CPU time
448	1C0	Integer	4	WTASOTN	Number of other calls
452	1C4	Character	8	WTASMLW	Maximum latch wait time
460	1CC	Integer	4	WTASMLWN	Maximum wait latch number
464	1D0	Character	4	*	Reserved
468	1D4	Integer	4	*	Reserved
<b>Commit statistics</b>					
472	1D8	Character	8	WTASCMET	Commit elapsed time
480	1E0	Character	8	WTASCMCT	Commit CPU time
488	1E8	Integer	4	WTASCMN	Commit number of calls
<b>Backout statistics</b>					
492	1EC	Character	8	WTASBAET	Backout elapsed time
500	1F4	Character	8	WTASBACT	Backout CPU time
508	1FC	Integer	4	WTASBAN	Backout number of calls
512	200	Character	4	*	Reserved
<b>Journal and logging information</b>					
516	204	Character	8	WTASJWET	Log write elapsed time in STCK format
524	20C	Integer	4	WTASJWN	Number of log writes

Table 23. Layout of the task-related information (WTAS) structure (continued)

Offset: Dec	Offset: Hex	Type	Length	Name	Description
528	210	Integer	4	WTASJWB	Number of bytes written to the log
532	214	Character	8	WTASJCET	Elapsed time waiting for log data to be forced to DASD
540	21C	Integer	4	WTASJCN	Number of times the log was forced
544	220	Integer	4	WTASSUSN	Number of times the task was suspended
548	224	Character	8	WTASSUSE	Total suspend time
<b>Page set 0 logging activity</b>					
556	22C	Character	8	WTASPSE0	Elapsed time logging for page set 0
564	234	Integer	4	WTASPSN0	Logging requests for page set 0
<b>DB2 manager</b>					
568	238	Character	8	WTASDBET	DB2 elapsed time for thread
576	240	Character	8	WTASDBES	DB2 elapsed time for server
584	248	Character	8	WTASDBMT	DB2 maximum elapsed time for thread
592	250	Character	8	WTASDBMS	DB2 maximum elapsed time for server
600	258	Integer	4	WTASDBCT	Number of DB2 requests
604	25C	64-bit integer	8	WTASDBGT	Bytes written to DB2
612	264	64-bit integer	8	WTASDBPT	Bytes read from DB2
<b>CF manager</b>					
620	26C	Integer	4	WTASCSEC	Number of IXLLSTE calls
624	270	Integer	4	WTASCMEC	Number of IXLLSTM calls
628	274	Integer	4	WTASRSEC	Number of IXLLSTE redrives
632	278	Integer	4	WTASRMEC	Number of IXLLSTM redrives
636	27C	Character	8	WTASSSTC	Time spent in IXLLSTE calls
644	284	Character	8	WTASMSTC	Time spent in IXLLSTM calls
652	28C	Character	8	* (3)	Reserved
<b>Interval data and page counts</b>					
676	2A4	Character	8	WTASINTS	Interval start - for post processing
684	2AC	Character	8	WTASINTE	Interval end - for post processing

Table 23. Layout of the task-related information (WTAS) structure (continued)

Offset: Dec	Offset: Hex	Type	Length	Name	Description
692	2B4	Integer	4	WTASGPO	Get pages old
696	2B8	Integer	4	WTASGPN	Get pages new

Table 24. Layout of the Queue (WQ) structure

Offset: Dec	Offset: Hex	Type	Length	Name	Description
0	0	Structure	576	WQSTAT	
0	0	Integer	2	WQID	Control block hex ID
2	2	Integer	2	WQLL	Length of the block
4	4	Character	4	WQEYE	Eye catcher (WQST)
8	8	Integer	4	WQVER	Version number
12	C	Address	4	WQNEXT	Reserved
16	10	Character	16	CORREL	Correlator to tie block to owning WTAS
32	20	Character	48	OBJNAME	Object name as opened
80	50	Character	48	BASENAME	Base name or generate name if applicable
128	80	Character	8	OPENTIME	Time queue opened (this is the first time if data is accumulated)
136	88	Character	8	CLOSTIME	Time the queue was closed (this is the last time if data is accumulated)
<b>Object information</b>					
144	90	Integer	4	QTYPE	Queue type (for example, local)
148	94	Integer	4	INDXTYPE	Index type of queue
152	98	Integer	4	QSGDISP	QSGDISP (for example, SHARED or GROUP)
<b>MQOPEN</b>					
156	9C	Character	4	OPENEYE	Eye catcher (OPEN)
160	A0	Character	8	OPENET	Total elapsed time for MQOPEN processing
168	A8	Character	8	OPENCT	Total amount of CPU time processing MQOPEN calls
176	B0	Integer	4	OPENN	Number of MQOPEN calls
<b>MQCLOSE</b>					
180	B4	Character	4	CLOSEEYE	Eye catcher (CLOS)
184	B8	Character	8	CLOSEET	Total elapsed time for MQCLOSE processing
192	C0	Character	8	CLOSECT	Total CPU times used for MQCLOSE processing
200	C8	Integer	4	CLOSEN	Number of MQCLOSE calls

Table 24. Layout of the Queue (WQ) structure (continued)

Offset: Dec	Offset: Hex	Type	Length	Name	Description
<b>MQGET</b>					
204	CC	Character	4	GETEYE	Eye catcher (GET)
208	D0	Character	8	GETET	Elapsed time processing MQGET calls
216	D8	Character	8	GETCT	CPU times used processing MQGET calls
224	E0	Integer	4	GETN	Total number of MQGET calls
228	E4	Integer	4	GETBRWA	Number of MQGET browses (any)
232	E8	Integer	4	GETBRWS	Number of MQGET browses (specific)
236	EC	Integer	4	GETA	Number of MQGET calls (any)
240	F0	Integer	4	GETS	Number of MQGET calls (specific)
244	F4	Integer	4	GETERR	Number of unaccountable MQGETs
248	F8	Character	8	GETJWET	Elapsed time waiting for a journal write to complete. This is for getting persistent messages out of syncpoint.
256	100	Integer	4	GETJWN	Number of journal write requests. This is for getting persistent messages out of syncpoint.
260	104	Character	8	GETPSET	Elapsed time waiting for a read from a page set
268	10C	Integer	4	GETPSN	Number of reads from a page set
272	110	Character	8	GETSUSET	Total suspend time for MQGET calls
280	118	Integer	4	GETSUSN	Number of times suspended
284	11C	Integer	4	GETEPAGE	Number of empty pages skipped over when doing an MQGET
288	120	Integer	4	GETSMMSG	Number of messages skipped when doing an MQGET, either by MsgId or CorrelId
292	124	Integer	4	GETEXMSG	Number of expired messages processed (this causes an increase in time because the event messages need to be produced)
<b>MQPUT</b>					
296	128	Character	4	PUTEYE	Eye catcher (PUT)

Table 24. Layout of the Queue (WQ) structure (continued)

Offset: Dec	Offset: Hex	Type	Length	Name	Description
300	12C	Character	8	PUTET	Total elapsed time for the MQPUT calls
308	134	Character	8	PUTCT	CPU time used during MQPUT processing
316	13C	Integer	4	PUTN	Number of MQPUT requests
320	140	Character	8	PUTJWET	Elapsed time waiting for a journal write request. This is for putting persistent messages out of syncpoint.
328	148	Integer	4	PUTJWN	Number of journal write requests. This is for putting persistent messages out of syncpoint.
332	14C	Character	8	PUTSUSET	Elapsed time the task was suspended for
340	154	Integer	4	PUTSUSN	Number of times suspended
344	158	Character	8	PUTPSET	Time taken to read from a page set for MQPUT
352	160	Integer	4	PUTPSN	Number of page set put requests
<b>MQPUT1</b>					
356	164	Character	4	PUT1EYE	Eye catcher (PUT1)
360	168	Character	8	PUT1ET	Total elapsed time for the MQPUT1 calls
368	170	Character	8	PUT1CT	CPU time used during MQPUT1 processing
376	178	Integer	4	PUT1N	Number of MQPUT1 requests
380	17C	Character	8	PUT1JWET	Elapsed time waiting for a journal write request. This is for putting persistent messages out of syncpoint.
388	184	Integer	4	PUT1JWN	Number of journal write requests. This is for putting persistent messages out of syncpoint.
392	188	Character	8	PUT1SUSET	Elapsed time the task was suspended
400	190	Integer	4	PUT1SUSN	Number of times suspended
404	194	Character	8	PUT1PSET	Time taken to read from a page set for MQPUT1
412	19C	Integer	4	PUT1PSN	Number of page set MQPUT1 requests
<b>MQINQ</b>					
416	1A0	Character	4	INQEYE	Eye catcher (INQ)

Table 24. Layout of the Queue (WQ) structure (continued)

Offset: Dec	Offset: Hex	Type	Length	Name	Description
420	1A4	Character	8	INQET	Total elapsed time for the MQINQ calls
420	1A4	Character	8	INQET	Total elapsed time for the MQINQ calls
436	1B4	Integer	4	INQN	Number of MQINQ requests
<b>MQSET</b>					
440	1B8	Character	4	SETEYE	Eye catcher (SET)
444	1BC	Character	8	SETET	Total elapsed time for the MQSET calls
452	1C4	Character	8	SETCT	CPU time used during MQSET processing
460	1CC	Integer	4	SETN	Number of MQSET requests
464	1D0	Character	8	SETJWET	Elapsed time waiting for journal write requests
472	1D8	Integer	4	SETJWN	Number of journal write requests
<b>Other statistics</b>					
476	1DC	Integer	4	NPS	Page set number.  This field is not used for the 'fast put for waiting getter' function. See the PUTPWG field.
480	1E0	Character	12	CFSTRUCNAME	Name of CF structure
492	1EC	Integer	4	NBUFFPOOL	Buffer pool number.  This field is not used for the 'fast put for waiting getter' function. See the PUTPWG field.
496	1F0	Character	8	PUTBYTES	Total number of bytes put successfully
504	1F8	Character	8	GETBYTES	Total number of bytes got successfully
512	200	Integer	4	VALIDPUT	Number of MQPUT calls writing data
516	204	Integer	4	VALIDGET	Number of MQGET calls with data
520	208	Integer	4	NGEN	Number of messages generated (including COA, COD, event, and expiry messages)
524	20C	Integer	4	GETMAXMS	Get maximum messages size
528	210	Integer	4	GETMINMS	Get minimum messages size

Table 24. Layout of the Queue (WQ) structure (continued)

Offset: Dec	Offset: Hex	Type	Length	Name	Description
532	214	Integer	4	PUTMAXMS	Put maximum messages size
536	218	Integer	4	PUTMINMS	Put minimum messages size
540	21C	Character	8	MAXLATNT	Maximum latency of message
548	224	Character	8	MINLATNT	Minimum latency of message
556	22C	Character	8	TOTLATNT	Total latency of messages
564	234	Integer	4	*	Reserved
568	238	Integer	4	USE_COUNT	Use count (plus one for MQOPEN, minus one for MQCLOSE)
572	23C	Integer	4	TOTAL_USE	Total number of calls using this queue
576	240	Integer	4	GETPMSG	Number of persistent messages created using MQPUT
580	244	Integer	4	PUTPMSG	Number of persistent messages retrieved using MQGET
584	248	Integer	4	PUT1PMSG	Number of persistent messages created using MQPUT1
588	24C	Integer	4	MAXQDPTH	Maximum queue depth encountered
592	250	Integer	2	FLAGS	
594	252	Integer	2	*	
The following fields are included only if WQ_VER is greater than WQ_VER_4:					
596	254	Integer	4	GETDVAL	Number of successful destructive MQGET calls
600	258	Character	8	GETJCET	Elapsed time waiting for force journal writes to complete
608	260	Integer	4	GETJCN	Number of force journal writes during MQGET calls
612	264	Integer	4	PUTPWG	Number of MQPUT calls where the message is passed directly to the waiting getter.
616	268	Character	8	PUTJCET	Elapsed time waiting for force journal writes to complete
624	270	Integer	4	PUTJCN	Number of force journal writes during MQPUT calls

Table 24. Layout of the Queue (WQ) structure (continued)

Offset: Dec	Offset: Hex	Type	Length	Name	Description
628	274	Integer	4	PUT1PWG	Number of MQPUT1 calls where the message is passed directly to the waiting getter.
632	278	Character	8	PUT1JCET	Elapsed time waiting for force journal writes to complete
640	280	Integer	4	PUT1JCN	Number of force journal writes during MQPUT1 calls
644	284	Character	8	SETJCET	Elapsed time waiting for force
652	28C	Integer	4	SETJCN	Number of force journal writes during MQPUT calls
656	290	Integer	4	*	
672	2A0	Integer	4	*	
<p><b>Note:</b></p> <ul style="list-style-type: none"> <li>• The average size of message put to a queue or got from a queue is not specifically recorded. However, it can be obtained by dividing PUTBYTES by VALIDPUT or GETBYTES by VALIDGET, as appropriate.</li> <li>• The NPS and NBUFFPOOL fields are always zero if no MQGET, MQPUT, or MQPUT1 calls are performed on the queue. If MQGET, MQPUT, or MQPUT1 calls are issued, the NPS and NBUFFPOOL fields refer to the values associated with the page set on which the messages reside.</li> <li>• The NPS and NBUFFPOOL fields are not updated for the 'fast put for waiting getter' function. This function means that a message is not stored in a buffer pool or page set if the message is nonpersistent, and PUT is out of syncpoint, and there is a waiting getter also out of syncpoint. The fields 'Put to waiting getter', PUTPWG, and PUT1PWG, are incremented when the message is passed directly to a waiting getter during a MQPUT or MQPUT1 call respectively.</li> </ul>					

Table 25. Layout of the Task Id structure (WTID)

Offset: Dec	Offset: Hex	Type	Length	Name	Description
0	0	Structure	208	WTID	
0	0	Integer	2	WTIDSHEX	Hex ID of block
2	2	Integer	2	WTIDLEN	Length of block
4	4	Character	4	WTIDEYEC	Eye catcher
8	8	Character	186	WTASID	
8	8	Integer	4	WTIDATYP	CCBCTCOD 1=CICS etc
12	C	Character	8	WTIDCCN	CCBNAME connection name. See Table 28 on page 119.
20	14	Character	8	WTIDOPID	CCBOPID operator ID
28	1C	Character	16	WTIDNID	NID



Table 25. Layout of the Task Id structure (WTID) (continued)

Offset: Dec	Offset: Hex	Type	Length	Name	Description
44	2C	Character	12	WTIDCORI	Correlator. See Table 26 or Table 27 for the internal structure of this field.
56	38	Character	24	WTIDUOWI	LUWID
80	50	Character	22	WTIDACCT	Accounting token
102	66	Character	20	WTIDCHL	Channel name. See Table 28.
122	7A	Character	48	WTIDCHLC	Channel connection name. See Table 28.
170	AA	Character	16	WTIDCTXT	Current context token
186	BA	Character	8	WTIDTRAN	CCBUSER MVS user ID
194	C2	Character	2	*	Reserved
196	C4	Address	4	WTIDCFWD	Reserved
200	C8	Address	4	WTIDCBWD	Reserved
204	CC	Address	4	WTIDWTAS	Reserved
208	D0	Character	0	*	Reserved

Table 26. Structure of the WTIDCORI for a CICS system

Offset: Dec	Offset: Hex	Type	Length	Name	Description
44	2C	Hexadecimal	4	WTICTNO	CICS thread number
48	30	Character	4	WTIDCTRN	CICS transaction name
52	34	Packed decimal	4	WTIDCTSK	CICS task number

Table 27. Structure of the WTIDCORI for an IMS system

Offset: Dec	Offset: Hex	Type	Length	Name	Description
44	2C	Hexadecimal	4	WTIDPST	IMS partition specification table (PST) region identifier
48	30	Character	8	WTIDPSB	IMS program specification block (PSB) name

## Meaning of the channel names

The channel name in the WTID is constructed as shown in the following example. In this example a sender channel exists from queue manager QM1 to queue manager QM2.

Table 28. Meaning of channel names

Field name	Meaning	Example
For queue manager QM1 the sender channel has the following fields set:		
WTIDCCN	The job name	QM1CHIN
WTIDCHL	The channel name	QM1.QM2

Table 28. Meaning of channel names (continued)

WTIDCHLC	This is defined in the CONNAME of the channel	WINMVS2B(2162)
For queue manager QM2 the receiver channel has the following fields set:		
WTIDCCN	The job name	QM2CHIN
WTIDCHL	The channel name	QM1.QM2
WTIDCHLC	Where the channel came from	9.20.101.14

## Sample subtype 1 and subtype 2 records

Figure 23 and Figure 24 on page 121 show examples of SMF type 116, subtype 1 and subtype 2 records. These two accounting records were created for a batch job that opened 80 queues. Because many queues were opened, a subtype 2 record was required to contain all the information produced.

```

000000 703C0000 5E74002D 983B0100 229FD4E5 *....;.....MV*
000010 F4F1D4D8 F0F70001 F6F0F000 00006FCC *41MQ07..600...?.*
000020 00700001 0000003C 00D00001 0000010C *......}.....*
000030 02C00001 000003CC 02400030 F70000D0 *.{..... ..7..}*
000040 E6E3C9C4 00000000 00000000 00000040 *WTID..... *
.
.
.
000100 00000000 00000000 7F4A4BB8 F70102C0 *....."...7..{*
000110 E6E3C1E2 B4802373 0BF07885 7F4AE718 *WTAS....0..".X.*

```

Figure 23. Example SMF type 116, subtype 1 record. This record contains a CSQDWTID control block, the CSQDWTAS control block, and the first set of CSQDWQST control blocks.

The first self-defining section starts at X'24' and is **bold** in the example; X'0000003C' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is *italic*; X'0000010C' is the offset to the WTAS data record, X'02C0' is the length of the WTAS record, and X'0001' is the number of WTAS records.

The third self-defining section starts at X'34' and is **bold** in the example; X'000003CC' is the offset to the first WQST data record, X'0240' is the length of the WQST record, and X'0030' is the number of WQST records.

Figure 24 on page 121 shows an example of an SMF type 116, subtype 2 record.

```

000000 49740000 5E74002D 983B0100 229FD4E5 *....;.....MV*
000010 F4F1D4D8 F0F70002 F6F0F000 00004904 *41MQ07..600....*
000020 00700001 00000034 00D00001 00000104 *.....}.....*
000030 02400020 F70000D0 E6E3C9C4 00000002 *..7..}WTID....*
.
.
.
000100 7F4A4BB8 F7020240 E6D8E2E3 00000001 *"...7.. WQST....*

```

Figure 24. Example SMF type 116, subtype 2 record. This record contains a CSQDWTID control block and the remaining CSQDWQST control blocks.

The first self-defining section starts at X'24' and is **bold** in the example; X'00000034' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is *italic*; X'00000104' is the offset to the first WQST data record, X'0240' is the length of the WQST record, and X'0020' is the number of WQST records.

Figure 25 shows an example of an SMF type 116, subtype 1 record where no queues have been opened and there are consequently no self-defining sections for WQST records..

```

000000          5E740039 4E9B0104 344FD4E5 * .....|MV*
000010 F4F1D4D8 F0F70001 F6F0F000 000003DC *41MQ07..600....*
000020 00800001 00000034 00D00001 00000104 *.....*
000030 02D80001 F70000D0 E6E3C9C4 00000002 *.Q..7...WTID....*
000040 C1F8C5C1 C4C5D740 C1F8C5C1 C4C54040 *A8EADEP A8EADE *
000050 40404040 40404040 00000000 00000000 * .....*
000060 40404040 40404040 4040          *      *

```

Figure 25. Example SMF type 116, subtype 1 record with no WQST data records

The first self-defining section starts at X'24' and is **bold** in the example; X'00000034' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is *italic*; X'0000010C' is the offset to the WTAS data record, X'02D8' is the length of the WTAS record, and X'0001' is the number of WTAS records.

There is no self-defining section describing a WQST data record, equivalent to the third self-defining section in Figure 23 on page 120.



---

## Chapter 7. Setting up security

---

### Using RACF classes and profiles

This chapter discusses the following subjects:

- “Using RACF security classes”
- “RACF profiles” on page 124
- “Switch profiles” on page 126

### Using RACF security classes

RACF classes are used to hold the profiles required for WebSphere MQ security checking. Each RACF class holds one or more profiles used at some point in the checking sequence, as shown in Table 29.

*Table 29. RACF classes used by WebSphere MQ*

Member class	Group class	Contents
MQADMIN	GMQADMIN	Profiles:  Used mainly for holding profiles for administration-type functions. For example: <ul style="list-style-type: none"><li>• Profiles for WebSphere MQ security switches</li><li>• The RESLEVEL security profile</li><li>• Profiles for alternate user security</li><li>• The context security profile</li><li>• Profiles for command resource security</li></ul>
MXADMIN	GMXADMIN	Profiles:  Used mainly for holding profiles for administration-type functions. For example: <ul style="list-style-type: none"><li>• Profiles for WebSphere MQ security switches</li><li>• The RESLEVEL security profile</li><li>• Profiles for alternate user security</li><li>• The context security profile</li><li>• Profiles for command resource security</li></ul> This class can hold both uppercase and mixed case RACF profiles.
MQCONN		Profiles used for connection security
MQCMDS		Profiles used for command security
MQQUEUE	GMQQUEUE	Profiles used in queue resource security
MXQUEUE	GMXQUEUE	Mixed case and uppercase profiles used in queue resource security
MQPROC	GMQPROC	Profiles used in process resource security
MXPROC	GMXPROC	Mixed case and uppercase profiles used in process resource security
MQNLIST	GMQNLIST	Profiles used in namelist resource security

Table 29. RACF classes used by WebSphere MQ (continued)

Member class	Group class	Contents
MXNLIST	GMXNLIST	Mixed case and uppercase profiles used in namelist resource security
MXTOPIC	GMXTOPIC	Mixed case and uppercase profiles used in topic security

Some classes have a related *group class* that enables you to put together groups of resources that have similar access requirements. For details about the difference between the member and group classes and when to use a member or group class, see the *z/OS SecureWay® Security Server RACF Security Administrator's Guide*.

The classes must be activated before security checks can be made. To activate all the WebSphere MQ classes, you use can use this RACF command:

```
SETROPTS CLASSACT(MQADMIN,MXADMIN,MQQUEUE,MXQUEUE,MQPROC,MXPROC,
MQNLIST,MXNLIST,MXTOPIC,MQCONN,MQCMDS)
```

You should also ensure that you set up the classes so that they can accept generic profiles. You also do this with the RACF command SETROPTS, for example:

```
SETROPTS GENERIC(MQADMIN,MXADMIN,MQQUEUE,MXQUEUE,MQPROC,MXPROC,
MQNLIST,MXNLIST,MXTOPIC,MQCONN,MQCMDS)
```

## RACF profiles

All RACF profiles used by WebSphere MQ contain a prefix. For queue-sharing group level security, this is the queue-sharing group name. For queue manager level security, the prefix is the queue manager name. If you are using a mixture of queue manager and queue-sharing group level security, you will use profiles with both types of prefix. (Queue-sharing group and queue manager level security are described in the WebSphere MQ for z/OS Concepts and Planning Guide.)

For example, if you want to protect a queue called `QUEUE_FOR_SUBSCRIBER_LIST` in queue-sharing group `QSG1` at queue-sharing group level, the appropriate profile would be defined to RACF as:

```
RDEFINE MQQUEUE QSG1.QUEUE_FOR_SUBSCRIBER_LIST
```

If you want to protect a queue called `QUEUE_FOR_LOST_CARD_LIST`, that belongs to queue manager `STCD` at queue manager level, the appropriate profile would be defined to RACF as:

```
RDEFINE MQQUEUE STCD.QUEUE_FOR_LOST_CARD_LIST
```

This means that different queue managers and queue-sharing groups can share the same RACF database and yet have different security options.

Do not use generic queue manager names in profiles to avoid unanticipated user access.

WebSphere MQ allows the use of the percent character (%) in object names. However, RACF uses the % character as a single-character wild card. This means that when you define an object name with a % character in its name, you must consider this when you define the corresponding profile.

For example, for the queue CREDIT\_CARD\_%\_RATE\_INQUIRY, on queue manager CRDP, the profile would be defined to RACF as follows:

```
RDEFINE MQQUEUE CRDP.CREDIT_CARD_%_RATE_INQUIRY
```

This queue cannot be protected by a generic profile, such as, CRDP:\*\*.

WebSphere MQ allows the use of mixed case characters in object names. You can protect these objects by defining:

1. Mixed case profiles in the appropriate mixed case RACF classes, or
2. Generic profiles in the appropriate uppercase RACF classes.

In order to use mixed case profiles and mixed case RACF classes you must follow the steps described in *WebSphere MQ Migration Information*.

There are some profiles, or parts of profiles, that remain uppercase only as the values are provided by WebSphere MQ. These are:

- Switch profiles.
- All high-level qualifiers (HLQ) including subsystem and Queue-Sharing Group identifiers.
- Profiles for SYSTEM objects.
- Profiles for Default objects.
- The MQCMD class, so all command profiles are uppercase only.
- The MQCONN class, so all connection profiles are uppercase only.
- RESLEVEL profiles.
- The 'object' qualification in command resource profiles; for example, hlq.QUEUE.queueName. The resource name only is mixed case.
- Dynamic queue profiles hlq.CSQOREXX.\* , hlq.CSQUTIL.\* , and CSQXCMD.\*.
- The 'CONTEXT' part of the hlq.CONTEXT.resourcename.

For example, if you have a queue called PAYROLL.Dept1 on Queue Manager QM01 and you are using:

- Mixed case classes; you can define a profile in the WebSphere MQ RACF class MXQUEUE  
RDEFINE MXQUEUE QM01.PAYROLL.Dept1
- Uppercase classes; you can define a profile in the WebSphere MQ RACF class MQQUEUE  
RDEFINE MQQUEUE QM01.PAYROLL.\*

The first example, using mixed case classes, gives you more granular control over granting authority to access the resource.

## Switch profiles

To control the security checking performed by WebSphere MQ, you must define *switch profiles*. A switch profile is a normal RACF profile that has a special meaning to WebSphere MQ. The access list in switch profiles is not used by WebSphere MQ.

WebSphere MQ maintains an internal switch for each switch type shown in tables Table 30 on page 127 through Table 36 on page 131. Switch profiles can be maintained at queue-sharing group level or at queue manager level or at a combination of both. Using a single set of queue-sharing group security switch profiles, you can control security on all the queue managers within a queue-sharing group.

When a security switch is set on, the security checks associated with the switch are performed. When a security switch is set off, the security checks associated with the switch are bypassed. The default is that all security switches are set on.

### Switches and classes

When a queue manager is started (or when the MQADMIN or MXADMIN class is refreshed by the WebSphere MQ REFRESH SECURITY command), WebSphere MQ first checks the status of RACF and the appropriate class:

- The MQADMIN class if you are using uppercase profiles
- The MXADMIN class if you are using mixed case profile.

It sets the subsystem security switch off if any of these conditions is true:

- RACF is inactive or not installed.
- The MQADMIN or MXADMIN class is not defined (these classes are always defined for RACF because they are included in the class descriptor table (CDT)).
- The MQADMIN or MXADMIN class has not been activated.

If both RACF and the MQADMIN or MXADMIN class are active, WebSphere MQ checks the MQADMIN or MXADMIN class to see whether any of the switch profiles have been defined. It first checks the profiles described in “Profiles to control subsystem security” on page 127. If subsystem security is not required, WebSphere MQ sets the internal subsystem security switch off, and performs no further checks.

The profiles determine whether the corresponding WebSphere MQ switch is set on or off.

- If the switch is off, that type of security is deactivated.
- If any WebSphere MQ switch is set on, WebSphere MQ checks the status of the RACF class associated with the type of security corresponding to the WebSphere MQ switch. If the class is not installed or not active, the WebSphere MQ switch is set off. For example, process security checks are not carried out if the MQPROC or MXPROC class has not been activated. The class not being active is equivalent to defining NO.PROCESS.CHECKS profile for every queue manager and queue-sharing group that uses this RACF database.

### How switches work

To set a security switch off, you need to define a NO.\* switch profile for it. The existence of a NO.\* profile means that security checks are **not** performed for that



type of resource, unless you choose to override a queue-sharing group level setting on a particular queue manager. This is described in “Overriding queue-sharing group level settings.”

If your queue manager is not a member of a queue-sharing group, you do not need to define any queue-sharing group level profiles or any override profiles. However, you must remember to define these profiles if the queue manager joins a queue-sharing group at a later date.

Each NO.\* switch profile that WebSphere MQ detects turns off the checking for that type of resource. Switch profiles are activated during startup of the queue manager. If you change the switch profiles while any affected queue managers are running, you can get WebSphere MQ to recognize the changes by issuing the WebSphere MQ REFRESH SECURITY command.

The switch profiles must always be defined in the MQADMIN or MXADMIN class. Do not define them in the GMQADMIN or GMXADMIN class. Tables Table 30 and Table 36 on page 131 show the valid switch profiles and the security type they control.

### Overriding queue-sharing group level settings:

You can override queue-sharing group level security settings for a particular queue manager that is a member of that group. If you want to perform queue manager checks on an individual queue manager that are not performed on other queue managers in the group, use the (qmgr-name.YES.\*) switch profiles.

Conversely, if you do not want to perform a certain check on one particular queue manager within a queue-sharing group, define a (qmgr-name.NO.\*) profile for that particular resource type on the queue manager, and do not define a profile for the queue-sharing group. (WebSphere MQ only checks for a queue-sharing group level profile if it does not find a queue manager level profile.)

### Profiles to control subsystem security

The first security check made by WebSphere MQ is used to determine whether security checks are required for the whole WebSphere MQ subsystem. If you specify that you do not want subsystem security, no further checks are made.

The following switch profiles are checked to determine whether subsystem security is required. Figure 26 on page 128 shows the order in which they are checked.

Table 30. Switch profiles for subsystem level security

Switch profile name	Type of resource or checking that is controlled
qmgr-name.NO.SUBSYS.SECURITY	Subsystem security for this queue manager
qsg-name.NO.SUBSYS.SECURITY	Subsystem security for this queue-sharing group
qmgr-name.YES.SUBSYS.SECURITY	Subsystem security override for this queue manager

If your queue manager is not a member of a queue-sharing group, WebSphere MQ checks for the qmgr-name.NO.SUBSYS.SECURITY switch profile only.

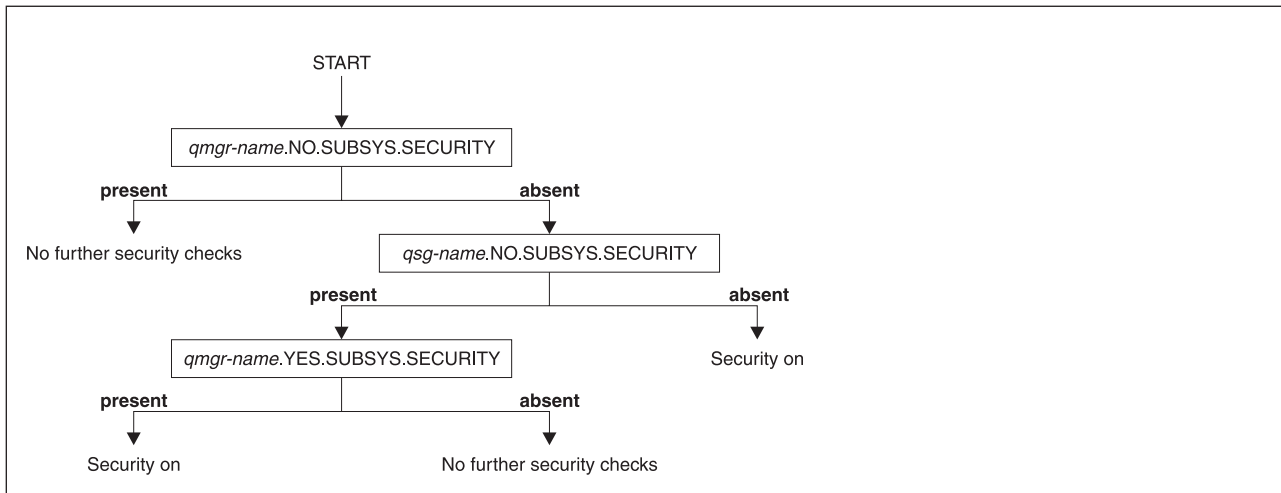


Figure 26. Checking for subsystem security

## Profiles to control queue-sharing group or queue manager level security

When WebSphere MQ has determined that security checking is required, it then determines whether checking is required at queue-sharing group or queue manager level, or both. These checks are not performed if your queue manager is not a member of a queue sharing group.

The following switch profiles are checked to determine the level required. Figure 27 on page 129 and Figure 28 on page 129 show the order in which they are checked.

Table 31. Switch profiles for queue-sharing group or queue manager level security

Switch profile name	Type of resource or checking that is controlled
qmgr-name.NO.QMGR.CHECKS	No queue manager level checks for this queue manager
qsg-name.NO.QMGR.CHECKS	No queue manager level checks for this queue-sharing group
qmgr-name.YES.QMGR.CHECKS	Queue manager level checks override for this queue manager
qmgr-name.NO.QSG.CHECKS	No queue-sharing group level checks for this queue manager
qsg-name.NO.QSG.CHECKS	No queue-sharing group level checks for this queue-sharing group
qmgr-name.YES.QSG.CHECKS	Queue-sharing group level checks override for this queue manager

If subsystem security is active, you cannot switch off both queue-sharing group and queue manager level security. If you try to do this, WebSphere MQ sets security checking on at both levels.

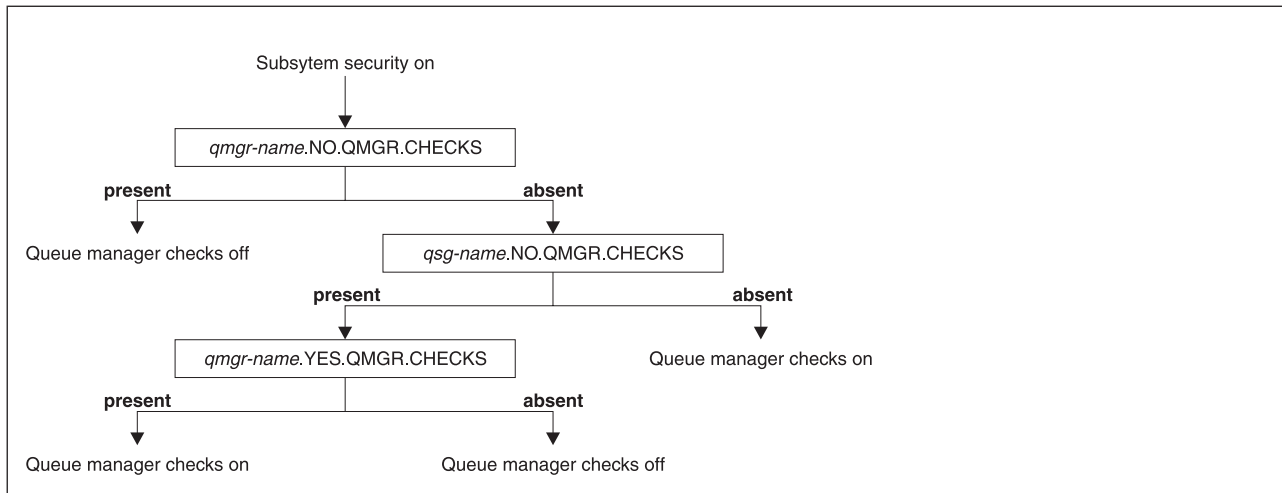


Figure 27. Checking for queue manager level security

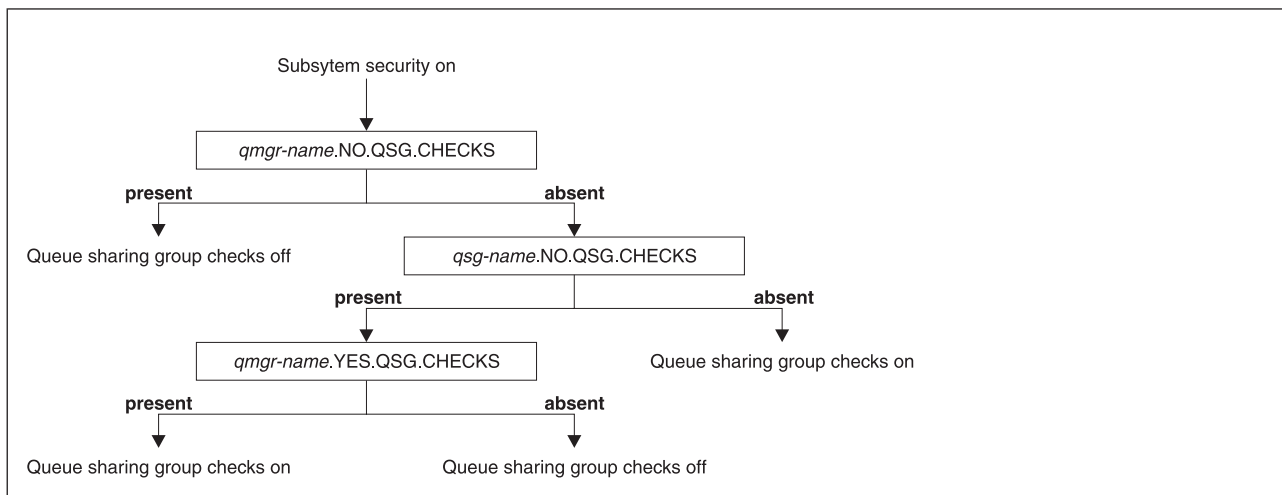


Figure 28. Checking for queue-sharing group level security

**Valid combinations of switches:**

Table 32, Table 33 on page 130, Table 34 on page 130, and Table 35 on page 130 show the sets of combinations of switch settings that are valid for each type of security level. If you use a combination of switch settings that is not valid, message CSQH026I is issued and security checking is set on at both queue-sharing group and queue manager level.

Table 32. Valid security switch combinations for queue manager level security

qmgr-name.NO.QSG.CHECKS
qsg-name.NO.QSG.CHECKS
qmgr-name.NO.QSG.CHECKS qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS
qsg-name.NO.QSG.CHECKS qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS

Table 33. Valid security switch combinations for queue-sharing group level security

qmgr-name.NO.QMGR.CHECKS
qsg-name.NO.QMGR.CHECKS
qmgr-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS

Table 34. Valid security switch combinations for queue manager and queue-sharing group level security

qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS No QSG.* profiles defined
No QMGR.* profiles defined qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
No profiles for either switch defined

Table 35. Other valid security switch combinations that switch both levels of checking on.

qmgr-name.NO.QMGR.CHECKS qmgr-name.NO.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS
qmgr-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qmgr-name.NO.QSG.CHECKS

## Resource level checks

Table 36 on page 131 shows the switch profiles used to control access to WebSphere MQ resources.

If your queue manager is part of a queue sharing group and you have both queue manager and queue-sharing group security active, you can use a YES.\* switch profile to override queue-sharing group level profiles and specifically turn on security for a particular queue manager.

Some profiles apply to both queue managers and queue-sharing groups. These are prefixed by the string *hlq* in this book and you should substitute the name of your queue-sharing group or queue manager, as applicable. Profile names shown prefixed by *qmgr-name* are queue-manager override profiles; you should substitute the name of your queue manager.

Table 36. Switch profiles for resource checking

Type of resource checking that is controlled	Switch profile name	Override profile for a particular queue manager
Connection security	hlq.NO.CONNECT.CHECKS	qmgr-name.YES.CONNECT.CHECKS
Queue security	hlq.NO.QUEUE.CHECKS	qmgr-name.YES.QUEUE.CHECKS
Process security	hlq.NO.PROCESS.CHECKS	qmgr-name.YES.PROCESS.CHECKS
Namelist security	hlq.NO.NLIST.CHECKS	qmgr-name.YES.NLIST.CHECKS
Context security	hlq.NO.CONTEXT.CHECKS	qmgr-name.YES.CONTEXT.CHECKS
Alternate user security	hlq.NO.ALTERNATE.USER.CHECKS	qmgr-name.YES.ALTERNATE.USER.CHECKS
Command security	hlq.NO.CMD.CHECKS	qmgr-name.YES.CMD.CHECKS
Command resource security	hlq.NO.CMD.RESC.CHECKS	qmgr-name.YES.CMD.RESC.CHECKS
Topic security	hlq.NO.TOPIC.CHECKS	qmgr-name.YES.TOPIC.CHECKS

**Note:** Generic switch profiles such as hlq.NO.\*\* are ignored by WebSphere MQ

For example, say you want to perform process security checks on queue manager QM01, which is a member of queue-sharing group QSG3 but you do not want to perform process security checks on any of the other queue managers in the group. Define the following switch profiles:

```
QSG3.NO.PROCESS.CHECKS
QM01.YES.PROCESS.CHECKS
```

If you want to have queue security checks performed on all the queue managers in the queue-sharing group, except QM02, define the following switch profile:

```
QM02.NO.QUEUE.CHECKS
```

(There is no need to define a profile for the queue sharing group because the checks are automatically enabled if there is no profile defined.)

### An example of defining switches

Four WebSphere MQ subsystems have been defined:

- MQP1 (a production system)
- MQP2 (a production system)
- MQD1 (a development system)
- MQT1 (a test system)

All four queue managers are members of queue-sharing group QS01. All WebSphere MQ RACF classes have been defined and activated.

These subsystems have different security requirements:

- The production systems require full WebSphere MQ security checking to be active at queue-sharing group level on both systems.

This is done by specifying the following profile:

```
RDEFINE MQADMIN QS01.NO.QMGR.CHECKS
```

This sets queue-sharing group level checking for all the queue managers in the queue-sharing group. You do not need to define any other switch profiles for the production queue managers because you want to check everything for these systems.

- Test queue manager MQT1 also requires full security checking. However, because you might want to change this later, security can be defined at queue-manager level so that you can change the security settings for this queue manager without affecting the other members of the queue-sharing group.

This is done by defining the NO.QSG.CHECKS profile for MQT1 as follows:

```
RDEFINE MQADMIN MQT1.NO.QSG.CHECKS
```

- Development queue manager MQD1 has different security requirements from the rest of the queue-sharing group. It requires only connection and queue security to be active.

This is done by defining a MQD1.YES.QMGR.CHECKS profile for this queue manager, and then defining the following profiles to switch off security checking for the resources that do not need to be checked:

```
RDEFINE MQADMIN MQD1.NO.CMD.CHECKS
RDEFINE MQADMIN MQD1.NO.CMD.RESC.CHECKS
RDEFINE MQADMIN MQD1.NO.PROCESS.CHECKS
RDEFINE MQADMIN MQD1.NO.NLIST.CHECKS
RDEFINE MQADMIN MQD1.NO.CONTEXT.CHECKS
RDEFINE MQADMIN MQD1.NO.ALTERNATE.USER.CHECKS
```

When the queue manager is active, you can display the current security settings by issuing the DISPLAY SECURITY MQSC command.

You can also change the switch settings when the queue manager is running by defining or deleting the appropriate switch profile in the MQADMIN class. To make the changes to the switch settings active, you must issue the REFRESH SECURITY command for the MQADMIN class.

See “Refreshing queue manager security” on page 180 for more details about using the DISPLAY SECURITY and REFRESH SECURITY commands.

---

## Profiles used to control access to WebSphere MQ resources

You must define RACF profiles to control access to WebSphere MQ resources, in addition to the switch profiles that might have been defined. If you do not have a resource profile defined for a particular security check, and a user issues a request that would involve making that check, WebSphere MQ denies access. You do not need to define profiles for security types relating to any security switches that you have deactivated.

This chapter discusses the following types of RACF profile:

- “Profiles for connection security” on page 133

- “Profiles for queue security” on page 135
- “Profiles for processes” on page 145
- “Profiles for namelists” on page 146
- “Profiles for alternate user security” on page 147
- “Profiles for context security” on page 149
- “Profiles for command security” on page 151
- “Profiles for topic security” on page 158
- “Profiles for command resource security” on page 161

## Profiles for connection security

If connection security is active, you must define profiles in the MQCONN class and permit the necessary groups or user IDs access to those profiles, so that they can connect to WebSphere MQ.

To enable a connection to be made, you must grant users RACF READ access to the appropriate profile. (If no queue manager level profile exists, and your queue manager is a member of a queue-sharing group, checks might be made against queue-sharing group level profiles, if the security is set up to do this.)

A connection profile qualified with a queue manager name controls access to a specific queue manager and users given access to this profile can connect to that queue manager. A connection profile qualified with queue-sharing group name controls access to all queue managers within the queue-sharing group for that connection type. For example, a user with access to QS01.BATCH can use a batch connection to any queue manager in queue-sharing group QS01 that has not got a queue manager level profile defined.

### Note:

1. For information about the user IDs checked for different security requests, see “User IDs for security checking” on page 170.
2. Resource level security (RESLEVEL) checks are also made at connection time. For details, see “Using the RESLEVEL security profile” on page 163.

WebSphere MQ security recognizes the following different types of connection:

- Batch (and batch-type) connections, these include:
  - z/OS batch jobs
  - TSO applications
  - USS sign-ons
  - DB2 stored procedures
- CICS connections
- IMS connections from control and application processing regions
- The WebSphere MQ channel initiator

### Connection security profiles for batch connections

Profiles for checking batch and batch-type connections take the form:

hlq.BATCH

where `hlq` can be either the `qmgr-name` (queue manager name) or `qsg-name` (queue-sharing group name). If you are using both queue manager and queue-sharing group level security, WebSphere MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue-sharing group name. If it fails to find either profile, the connection request fails.

For batch or batch-type connection requests, you must permit the user ID associated with the connecting address space to access the connection profile. For example, the following RACF command allows users in the `CONNTQM1` group to connect to the queue manager `TQM1`; these user IDs will be permitted to use any batch or batch-type connection.

```
RDEFINE MQCONN TQM1.BATCH UACC(NONE)
PERMIT TQM1.BATCH CLASS(MQCONN) ID(CONNTQM1) ACCESS(READ)
```

## Connection security profiles for CICS connections

Profiles for checking connections from CICS take the form:

```
hlq.CICS
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue-sharing group name). If you are using both queue manager and queue-sharing group level security, WebSphere MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue-sharing group name. If it fails to find either profile, the connection request fails.

For connection requests by CICS, you need only permit the CICS address space user ID access to the connection profile.

For example, the following RACF commands allow the CICS address space user ID `KCBCICS` to connect to the queue manager `TQM1`:

```
RDEFINE MQCONN TQM1.CICS UACC(NONE)
PERMIT TQM1.CICS CLASS(MQCONN) ID(KCBCICS) ACCESS(READ)
```

## Connection security profiles for IMS connections

Profiles for checking connections from IMS take the form:

```
hlq.IMS
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue-sharing group name). If you are using both queue manager and queue-sharing group level security, WebSphere MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue-sharing group name. If it fails to find either profile, the connection request fails.



For connection requests by IMS, permit access to the connection profile for the IMS control and dependent region user IDs.

For example, the following RACF commands allow:

- The IMS region user ID, IMSREG, to connect to the queue manager TQM1.
- Users in group BMPGRP to submit BMP jobs.

```
RDEFINE MQCONN TQM1.IMS UACC(NONE)
PERMIT TQM1.IMS CLASS(MQCONN) ID(IMSREG,BMPGRP) ACCESS(READ)
```

## Connection security profiles for the channel initiator

Profiles for checking connections from the channel initiator take the form:

```
hlq.CHIN
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue-sharing group name). If you are using both queue manager and queue-sharing group level security, WebSphere MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue-sharing group name. If it fails to find either profile, the connection request fails

For connection requests by the channel initiator, define access to the connection profile for the user ID used by the channel initiator started task address space.

For example, the following RACF commands allow the channel initiator address space running with user ID DQCTRL to connect to the queue manager TQM1:

```
RDEFINE MQCONN TQM1.CHIN UACC(NONE)
PERMIT TQM1.CHIN CLASS(MQCONN) ID(DQCTRL) ACCESS(READ)
```

## Profiles for queue security

If queue security is active, you must:

- Define profiles in the MQQUEUE or GMQUEUE classes if using uppercase profiles.
- Define profiles in the MXQUEUE or GMXQUEUE classes if using mixed case profiles.
- Permit the necessary groups or user IDs access to these profiles, so that they can issue WebSphere MQ API requests that use queues.

Profiles for queue security take the form:

```
hlq.queueename
```

where hlq can be either qmgr-name (queue manager name) or qsg-name (queue-sharing group name), and queue name is the name of the queue being opened, as specified in the object descriptor on the **MQOPEN** or **MQPUT1** call.

A profile prefixed by the queue manager name controls access to a single queue on that queue manager. A profile prefixed by the queue-sharing group name controls access to access to one or more queues with that queue name on all queue managers within the queue-sharing group, or access to a shared queue by any queue manager within the group. This access can be overridden on an individual queue manager by defining a queue-manager level profile for that queue on that queue manager.

If your queue manager is a member of a queue-sharing group and you are using both queue manager and queue-sharing group level security, WebSphere MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue-sharing group name.

If you are using shared queues, you are recommended to use queue-sharing group level security.

For details of how queue security operates when the queue name is that of an alias or a model queue, see “Considerations for alias queues” on page 138 and “Considerations for model queues” on page 139.

The RACF access required to open a queue depends on the **MQOPEN** or **MQPUT1** options specified. If more than one of the **MQOO\_\*** and **MQPMO\_\*** options is coded, the queue security check is performed for the highest RACF authority required.

*Table 37. Access levels for queue security using the MQOPEN or MQPUT1 calls*

<b>MQOPEN or MQPUT1 option</b>	<b>RACF access level required to access hlq.queue name</b>
MQOO_BROWSE	READ
MQOO_INQUIRE	READ
MQOO_BIND_*	UPDATE
MQOO_INPUT_*	UPDATE
MQOO_OUTPUT or MQPUT1	UPDATE
MQOO_PASS_ALL_CONTEXT MQPMO_PASS_ALL_CONTEXT	UPDATE
MQOO_PASS_IDENTITY_CONTEXT MQPMO_PASS_IDENTITY_CONTEXT	UPDATE
MQOO_SAVE_ALL_CONTEXT	UPDATE
MQOO_SET_IDENTITY_CONTEXT MQPMO_SET_IDENTITY_CONTEXT	UPDATE
MQOO_SET_ALL_CONTEXT MQPMO_SET_ALL_CONTEXT	UPDATE
MQOO_SET	ALTER

For example, on WebSphere MQ queue manager QM77, all user IDs in the RACF group PAYGRP are to be given access to get messages from or put messages to all queues with names beginning with 'PAY.'. You can do this using these RACF

commands:

```
RDEFINE MQQUEUE QM77.PAY.** UACC(NONE)
PERMIT QM77.PAY.** CLASS(MQQUEUE) ID(PAYGRP) ACCESS(UPDATE)
```

Also, all user IDs in the PAYGRP group must have access to put messages on queues that do not follow the PAY naming convention. For example:

```
REQUEST_QUEUE_FOR_PAYROLL
SALARY.INCREASE.SERVER
REPLIES.FROM.SALARY.MODEL
```

You can do this by defining profiles for these queues in the GMQQUEUE class and giving access to that class as follows:

```
RDEFINE GMQQUEUE PAYROLL.EXTRAS UACC(NONE)
      ADDMEM(QM77.REQUEST_QUEUE_FOR_PAYROLL,
             QM77.SALARY.INCREASE.SERVER,
             QM77.REPLIES.FROM.SALARY.MODEL)
PERMIT PAYROLL.EXTRAS CLASS(GMQQUEUE) ID(PAYGRP) ACCESS(UPDATE)
```

**Note:**

1. If the RACF access level that an application has to a queue security profile is changed, the changes only take effect for any new object handles obtained (that is, new **MQOPENS**) for that queue. Those handles already in existence at the time of the change retain their existing access to the queue. If an application is required to use its changed access level to the queue rather than its existing access level, it must close and reopen the queue for each object handle that requires the change.
2. In the example, the queue manager name QM77 could also be the name of a queue-sharing group.

Other types of security checks might also occur at the time the queue is opened depending on the open options specified and the types of security that are active. See also “Profiles for context security” on page 149 and “Profiles for alternate user security” on page 147. For a summary table showing the open options and the security authorization needed when queue, context, and alternate user security are all active, see Table 42 on page 143.

If you are using publish/subscribe you must consider the following. When an MQSUB request is processed a security check is performed to ensure that the user ID making the request has the required access to put messages to the target WebSphere MQ queue as well as the required access to subscribe to the WebSphere MQ topic.

*Table 38. Access levels for queue security using the MQSUB call*

MQSUB option	RACF access level required to access hlq.queueName
MQSO_ALTER, MQSO_CREATE, and MQSO_RESUME	UPDATE

**Note:**

1. The `hlq.queueName` is the destination queue for publications. When this is a managed queue, you need access to the appropriate model queue to be used for the managed queue and the dynamic queue that are created.
2. You can use a technique like this for the destination queue you provide on an MQSUB API call if you want to distinguish between the users making the subscriptions, and the users retrieving the publications from the destination queue.

**Considerations for alias queues**

When you issue an **MQOPEN** or **MQPUT1** call for an alias queue, WebSphere MQ makes a resource check against the queue name specified in the object descriptor (MQOD) on the call. It does not check if the user is allowed access to the target queue name.

For example, an alias queue called `PAYROLL.REQUEST` resolves to a target queue of `PAY.REQUEST`. If queue security is active, you need only be authorized to access the queue `PAYROLL.REQUEST`. No check is made to see if you are authorized to access the queue `PAY.REQUEST`.

**Using alias queues to distinguish between MQGET and MQPUT requests**

The range of MQI calls available in one access level can cause a problem if you want to restrict access to a queue to allow only the **MQPUT** call or only the **MQGET** call. A queue can be protected by defining two aliases that resolve to that queue: one that enables applications to get messages from the queue, and one that enable applications to put messages on the queue.

The following text gives you an example of how you can define your queues to WebSphere MQ:

```
DEFINE QLOCAL(MUST_USE_ALIAS_TO_ACCESS) GET(ENABLED)
    PUT(ENABLED)

DEFINE QALIAS(USE_THIS_ONE_FOR_GETS) GET(ENABLED)
    PUT(DISABLED) TARGQ(MUST_USE_ALIAS_TO_ACCESS)

DEFINE QALIAS(USE_THIS_ONE_FOR_PUTS) GET(DISABLED)
    PUT(ENABLED) TARGQ(MUST_USE_ALIAS_TO_ACCESS)
```

You must also make the following RACF definitions:

```
RDEFINE MQQUEUE hlq.MUST_USE_ALIAS_TO_ACCESS UACC(NONE)
RDEFINE MQQUEUE hlq.USE_THIS_ONE_FOR_GETS UACC(NONE)
RDEFINE MQQUEUE hlq.USE_THIS_ONE_FOR_PUTS UACC(NONE)
```

Then you ensure that no users have access to the queue `hlq.MUST_USE_ALIAS_TO_ACCESS`, and give the appropriate users or groups access to the alias. You can do this using the following RACF commands:

```
PERMIT h1q.USE_THIS_ONE_FOR_GETS CLASS(MQQUEUE)
      ID(GETUSER,GETGRP) ACCESS(UPDATE)
PERMIT h1q.USE_THIS_ONE_FOR_PUTS CLASS(MQQUEUE)
      ID(PUTUSER,PUTGRP) ACCESS(UPDATE)
```

This means user ID GETUSER and user IDs in the group GETGRP are only allowed to get messages on MUST\_USE\_ALIAS\_TO\_ACCESS through the alias queue USE\_THIS\_ONE\_FOR\_GETS; and user ID PUTUSER and user IDs in the group PUTGRP are only allowed to put messages through the alias queue USE\_THIS\_ONE\_FOR\_PUTS.

**Note:**

1. If you want to use a technique like this, you must inform your application developers, so that they can design their programs appropriately.
2. You can use a technique like this for the destination queue you provide on and MQSUB API request if you want to distinguish between the users making the subscriptions and the users 'getting' the publications from the destination queue.

## Considerations for model queues

When you open a model queue, WebSphere MQ security makes two queue security checks:

1. Are you authorized to access the model queue?
2. Are you authorized to access the dynamic queue to which the model queue resolves?

If the dynamic queue name contains a trailing \* character, this \* is replaced by a character string generated by WebSphere MQ, to create a dynamic queue with a unique name. However, because the whole name, including this generated string, is used for checking authority, you should define generic profiles for these queues.

For example, an **MQOPEN** call uses a model queue name of CREDIT.CHECK.REPLY.MODEL and a dynamic queue name of CREDIT.REPLY.\* on queue manager (or queue-sharing group) MQSP.

To do this, you must issue the following RACF commands to define the necessary queue profiles:

```
RDEFINE MQQUEUE MQSP.CREDIT.CHECK.REPLY.MODEL
RDEFINE MQQUEUE MQSP.CREDIT.REPLY.**
```

You must also issue the corresponding RACF PERMIT commands to allow the user access to these profiles.

A typical dynamic queue name created by an **MQOPEN** is something like CREDIT.REPLY.A346EF00367849A0. The precise value of the last qualifier is unpredictable; this is why you should use generic profiles for such queue names.

A number of WebSphere MQ utilities put messages on dynamic queues. You should define profiles for the following dynamic queue names, and provide RACF UPDATE access to the relevant user IDs (see "User IDs for security checking" on page 170

page 170 for the correct user IDs):

```
SYSTEM.CSQUTIL.* (used by CSQUTIL)
SYSTEM.CSQOREXX.* (used by the operations and control panels)
SYSTEM.CSQXCMD.* (used by the channel initiator when processing CSQINPX)
CSQ4SAMP.* (used by the WebSphere MQ supplied samples)
```

You might also consider defining a profile to control use of the dynamic queue name used by default in the application programming copy members. The WebSphere MQ-supplied copybooks contain a default *DynamicQName*, which is CSQ.\*. This enables an appropriate RACF profile to be established.

**Note:** Do not allow application programmers to specify a single \* for the dynamic queue name. If you do, you must define an hlq.\*\* profile in the MQQUEUE class, and you would have to give it wide-ranging access. This means that this profile could also be used for other non-dynamic queues that do not have a more specific RACF profile. Your users could, therefore, gain access to queues you do not want them to access.

### Close options on permanent dynamic queues

If an application opens a permanent dynamic queue that was created by another application and then attempts to delete that queue with an **MQCLOSE** option, some extra security checks are applied when the attempt is made. See Table 39.

Table 39. Access levels for close options on permanent dynamic queues

MQCLOSE option	RACF access level required to hlq.queueName
MQCO_DELETE	ALTER
MQCO_DELETE_PURGE	ALTER

### Security and remote queues

When a message is put on a remote queue, the queue security that is performed by the local queue manager depends on how the remote queue is specified when it is opened. For example:

1. If the remote queue has been defined on the local queue manager through the WebSphere MQ DEFINE QREMOTE command, the queue that is checked is the name of the remote queue. For example, if a remote queue is defined on queue manager MQS1 as follows:

```
DEFINE QREMOTE(BANK7.CREDIT.REFERENCE)
  RNAME(CREDIT.SCORING.REQUEST)
  RQMNAME(BNK7)
  XMITQ(BANK1.TO.BANK7)
```

In this case, a profile for BANK7.CREDIT.REFERENCE, must be defined in the MQQUEUE class.

2. If the *ObjectQMgrName* for the request does not resolve to the local queue manager, a security check is carried out against the resolved (remote) queue manager name except in the case of a cluster queue where the check is made against the cluster queue name.

For example, the transmission queue BANK1.TO.BANK7 is defined on queue manager MQS1. An **MQPUT1** request is then issued on MQS1 specifying *ObjectName* as BANK1.INTERBANK.TRANSFERS and an *ObjectQMGrName* of BANK1.TO.BANK7. In this case, the user performing the request must have access to BANK1.TO.BANK7.

3. If you make an **MQPUT** request to a queue and specify *ObjectQMGrName* as the name of an alias of the local queue manager, only the queue name is checked for security, not that of the queue manager.

When the message gets to the remote queue manager it might be subject to additional security processing. For more information, see the WebSphere MQ Intercommunication manual.

## Dead-letter queue security

Undelivered messages can be put on a special queue called the dead-letter queue. If you have sensitive data that could possibly end up on this queue, you must consider the security implications of this because you do not want unauthorized users to retrieve this data.

Each of the following must be allowed to put messages onto the dead-letter queue:

- Application programs.
- The channel initiator address space and any MCA user IDs. (If the RESLEVEL profile is not present, or is defined so that network-received user IDs are checked, the network-received user ID also needs authority to put messages on the dead-letter queue.)
- CKTI, the WebSphere MQ-supplied CICS task initiator.
- CSQQTRMN, the WebSphere MQ-supplied IMS trigger monitor.

The only application that can retrieve messages from the dead-letter queue should be a 'special' application that processes these messages. However, a problem arises if you give applications RACF UPDATE authority to the dead-letter queue for **MQPUTs** because they can then automatically retrieve messages from the queue using **MQGET** calls. You cannot disable the dead-letter queue for get operations because, if you do, not even the 'special' applications could retrieve the messages.

One solution to this problem is set up a two-level access to the dead-letter queue. CKTI, message channel agent transactions or the channel initiator address space, and 'special' applications have direct access; other applications can only access the dead-letter queue through an alias queue. This alias is defined to allow applications to put messages on the dead-letter queue, but not to get messages from it.

This is how it might work:

1. Define the real dead-letter queue with attributes PUT(ENABLED) and GET(ENABLED), as shown in the sample thlqual.SCSQPROC(CSQ4INYG).
2. Give RACF UPDATE authority for the dead-letter queue to the following user IDs:
  - User IDs that the CKTI and the MCAs or channel initiator address space run under.
  - The user IDs associated with the 'special' dead-letter queue processing application.
3. Define an alias queue that resolves to the real dead-letter queue, but give the alias queue these attributes: PUT(ENABLED) and GET(DISABLED). Give the

alias queue a name with the same stem as the dead-letter queue name but append the characters “.PUT” to this stem. For example, if the dead-letter queue name is hlq.DEAD.QUEUE, the alias queue name would be hlq.DEAD.QUEUE.PUT.

4. To put a message on the dead-letter queue, an application uses the alias queue. This is what your application must do:
  - Retrieve the name of the real dead-letter queue. To do this, it opens the queue manager object using **MQOPEN** and then issues an **MQINQ** to get the dead-letter queue name.
  - Build the name of the alias queue by appending the characters ‘.PUT’ to this name, in this case, hlq.DEAD.QUEUE.PUT.
  - Open the alias queue, hlq.DEAD.QUEUE.PUT.
  - Put the message on the real dead-letter queue by issuing an **MQPUT** against the alias queue.
5. Give the user ID associated with the application RACF UPDATE authority to the alias, but no access (authority NONE) to the real dead-letter queue. This means that:
  - The application can put messages onto the dead-letter queue using the alias queue.
  - The application cannot get messages from the dead-letter queue using the alias queue because the alias queue is disabled for get operations.

The application cannot get any messages from the real dead-letter queue either because it does not have the correct RACF authority.

Table 40 summarizes the RACF authority required for the various participants in this solution.

*Table 40. RACF authority to the dead-letter queue and its alias*

Associated user IDs	Real dead-letter queue (hlq.DEAD.QUEUE)	Alias dead-letter queue (hlq.DEAD.QUEUE.PUT)
MCA or channel initiator address space and CKTI	UPDATE	NONE
‘Special’ application (for dead-letter queue processing)	UPDATE	NONE
User-written application user IDs	NONE	UPDATE

If you use this method, the application cannot determine the maximum message length (MAXMSGL) of the dead-letter queue. This is because the MAXMSGL attribute cannot be retrieved from an alias queue. Therefore, your application should assume that the maximum message length is 100 MB, the maximum size WebSphere MQ for z/OS supports. The real dead-letter queue should also be defined with a MAXMSGL attribute of 100 MB.

**Note:** User-written application programs should not normally use alternate user authority to put messages on the dead-letter queue. This reduces the number of user IDs that have access to the dead-letter queue.

## System queue security

Many of the system queues are accessed by the ancillary parts of WebSphere MQ:

- The CSQUTIL utility



- The operations and control panels
- The channel initiator address space

The user IDs under which these run must be given RACF access to these queues, as shown in Table 41.

Table 41. Access required to the SYSTEM queues by WebSphere MQ

SYSTEM queue	CSQUTIL	Operations and control panels	Channel initiator for distributed queuing
SYSTEM.ADMIN.CHANNEL.EVENT	-	-	UPDATE
SYSTEM.CHANNEL.INITQ	-	-	UPDATE
SYSTEM.CHANNEL.SYNCQ	-	-	UPDATE
SYSTEM.CLUSTER.COMMAND.QUEUE	-	-	ALTER
SYSTEM.CLUSTER.REPOSITORY.QUEUE	-	-	UPDATE
SYSTEM.CLUSTER.TRANSMIT.QUEUE	-	-	ALTER
SYSTEM.COMMAND.INPUT	UPDATE	UPDATE	UPDATE
SYSTEM.COMMAND.REPLY.MODEL	UPDATE	UPDATE	UPDATE
SYSTEM.CSQOREXX.*	-	UPDATE	-
SYSTEM.CSQUTIL.*	UPDATE	-	-
SYSTEM.CSQXCMD.*	-	-	UPDATE
SYSTEM.HIERARCHY.STATE	-	-	UPDATE
SYSTEM.INTER.QMGR.CONTROL	-	-	UPDATE
SYSTEM.INTER.QMGR.PUBS	-	-	UPDATE
SYSTEM.INTER.QMGR.FANREQ	-	-	UPDATE
SYSTEM.QSG.CHANNEL.SYNCQ	-	-	UPDATE
SYSTEM.QSG.TRANSMIT.QUEUE	-	-	UPDATE

## API-resource security access quick reference

Table 42 summarizes the **MQOPEN**, **MQPUT1**, **MQSUB**, and **MQCLOSE** options and the access required by the different resource security types.

Table 42. **MQOPEN**, **MQPUT1**, **MQSUB**, and **MQCLOSE** options and the security authorization required. Callouts shown like this **(1)** refer to the notes following this table.

RACF class: RACF profile:	Minimum RACF access level required			
	MXTOPIC (15 or 16)	MQQUEUE or MXQUEUE(1) (2)	MQADMIN or MXADMIN (3)	MQADMIN or MXADMIN (4)
<b>MQOPEN</b> option				
MQOO_INQUIRE		READ (5)	No check	No check
MQOO_BROWSE		READ	No check	No check
MQOO_INPUT_*		UPDATE	No check	No check
MQOO_SAVE_ALL_CONTEXT (6)		UPDATE	No check	No check
MQOO_OUTPUT (USAGE=NORMAL) (7)		UPDATE	No check	No check
MQOO_PASS_IDENTITY_CONTEXT (8)		UPDATE	READ	No check
MQOO_PASS_ALL_CONTEXT (8) (9)		UPDATE	READ	No check

Table 42. MQOPEN, MQPUT1, MQSUB, and MQCLOSE options and the security authorization required (continued). Callouts shown like this (1) refer to the notes following this table.

RACF class: RACF profile:	Minimum RACF access level required			
	MXTOPIC (15 or 16)	MQQUEUE or MXQUEUE(1) (2)	MQADMIN or MXADMIN (3)	MQADMIN or MXADMIN (4)
MQOO_SET_IDENTITY_CONTEXT (8) (9)		UPDATE	UPDATE	No check
MQOO_SET_ALL_CONTEXT (8) (10)		UPDATE	CONTROL	No check
MQOO_OUTPUT (USAGE (XMITQ) (11)		UPDATE	CONTROL	No check
MQOO_OUTPUT (topic object)	UPDATE (16)			
MQOO_OUTPUT (alias queue to topic object)	UPDATE (16)	UPDATE		
MQOO_SET		ALTER	No check	No check
MQOO_ALTERNATE_USER_AUTHORITY		(12)	(12)	UPDATE
<b>MQPUT1 option</b>				
Put on a normal queue (7)		UPDATE	No check	No check
MQPMO_PASS_IDENTITY_CONTEXT		UPDATE	READ	No check
MQPMO_PASS_ALL_CONTEXT		UPDATE	READ	No check
MQPMO_SET_IDENTITY_CONTEXT		UPDATE	UPDATE	No check
MQPMO_SET_ALL_CONTEXT		UPDATE	CONTROL	No check
MQOO_OUTPUT Put on a transmission queue (11)		UPDATE	CONTROL	No check
MQOO_OUTPUT (topic object)	UPDATE (16)			
MQOO_OUTPUT (alias queue to topic object)	UPDATE (16)	UPDATE		
MQPMO_ALTERNATE_USER_AUTHORITY		(13)	(13)	UPDATE
<b>MQCLOSE option</b>				
MQCO_DELETE (14)		ALTER	No check	No check
MQCO_DELETE_PURGE (14)		ALTER	No check	No check
MQCO_REMOVE_SUB	ALTER (15)			
<b>MQSUB option</b>				
MQSO_CREATE	ALTER (15)	(17)	(18)	
MQSO_ALTER	ALTER (15)	(17)	(18)	
MQSO_RESUME	READ (15)	(17)	No check	
MQSO_ALTERNATE_USER_AUTHORITY				UPDATE
MQSO_SET_IDENTITY_CONTEXT			(18)	

**Note:**

1. This option is not restricted to queues. Use the MQNLIST or MXNLIST class for namelists, and the MQPROC or MXPROC class for processes.
2. Use RACF profile: hlq.resourcename
3. Use RACF profile: hlq.CONTEXT.queueuname
4. Use RACF profile: hlq.ALTERNATE.USER.alternateuserid  
alternateuserid is the user identifier that is specified in the *AlternateUserId* field of the object descriptor. Note that up to 12 characters of the

*AlternateUserId* field are used for this check, unlike other checks where only the first 8 characters of a user identifier are used.

5. No check is made when opening the queue manager for inquiries.
6. MQOO\_INPUT\_\* must be specified as well. This is valid for a local, model or alias queue.
7. This check is done for a local or model queue that has a *Usage* queue attribute of MQUS\_NORMAL, and also for an alias or remote queue (that is defined to the connected queue manager.) If the queue is a remote queue that is opened specifying an *ObjectQMgrName* (not the name of the connected queue manager) explicitly, the check is carried out against the queue with the same name as *ObjectQMgrName* (which must be a local queue with a *Usage* queue attribute of MQUS\_TRANSMISSION).
8. MQOO\_OUTPUT must be specified as well.
9. MQOO\_PASS\_IDENTITY\_CONTEXT is implied as well by this option.
10. MQOO\_PASS\_IDENTITY\_CONTEXT, MQOO\_PASS\_ALL\_CONTEXT and MQOO\_SET\_IDENTITY\_CONTEXT are implied as well by this option.
11. This check is done for a local or model queue that has a *Usage* queue attribute of MQUS\_TRANSMISSION, and is being opened directly for output. It does not apply if a remote queue is being opened.
12. At least one of MQOO\_INQUIRE, MQOO\_BROWSE, MQOO\_INPUT\_\*, MQOO\_OUTPUT or MQOO\_SET must be specified as well. The check carried out is the same as that for the other options specified.
13. The check carried out is the same as that for the other options specified.
14. This applies only for permanent dynamic queues that have been opened directly, that is, not opened through a model queue. No security is required to delete a temporary dynamic queue.
15. Use RACF profile hlq.SUBSCRIBE.topicname.
16. Use RACF profile hlq.PUBLISH.topicname.
17. If on the MQSUB request you specified a destination queue for the publications to be sent to, then a security check is carried out against that queue to ensure that you have put authority to that queue.
18. If on the MQSUB request, with MQSO\_CREATE or MQSO\_ALTER options specified, you want to set any of the identity context fields in the MQSD structure, you also need to specify the MQSO\_SET\_IDENTITY\_CONTEXT option and you also need the appropriate authority to the context profile for the destination queue.

## Profiles for processes

If process security is active, you must:

- Define profiles in the MQPROC or GMQPROC classes if using uppercase profiles.
- Define profiles in the MXPROC or GMXPROC classes if using mixed case profiles.
- Permit the necessary groups or user IDs access to these profiles, so that they can issue WebSphere MQ API requests that use processes.

Profiles for processes take the form:

hlq.processname

where hlq can be either qmgr-name (queue manager name) or qsg-name (queue-sharing group name), and processname is the name of the process being opened.

A profile prefixed by the queue manager name controls access to a single process definition on that queue manager. A profile prefixed by the queue-sharing group name controls access to one or more process definitions with that name on all queue managers within the queue-sharing group. This access can be overridden on an individual queue manager by defining a queue-manager level profile for that process definition on that queue manager.

If your queue manager is a member of a queue-sharing group and you are using both queue manager and queue-sharing group level security, WebSphere MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue-sharing group name.

The following table shows the access required for opening a process.

Table 43. Access levels for process security

MQOPEN option	RACF access level required to hlq.processname
MQOO_INQUIRE	READ

For example, on queue manager MQS9, the RACF group INQVPRC must be able to inquire (**MQINQ**) on all processes starting with the letter V. The RACF definitions for this would be:

```
RDEFINE MQPROC MQS9.V* UACC(NONE)
PERMIT MQS9.V* CLASS(MQPROC) ID(INQVPRC) ACCESS(READ)
```

Alternate user security might also be active, depending on the open options specified when a process definition object is opened.

## Profiles for namelists

If namelist security is active, you must:

- Define profiles in the MQNLIST or GMQNLIST classes if using uppercase profiles.
- Define profiles in the MXNLIST or GMXNLIST classes if using mixed case profiles.
- Permit the necessary groups or user IDs access to these profiles.

Profiles for namelists take the form:

hlq.namelistname

where hlq can be either qmgr-name (queue manager name) or qsg-name (queue-sharing group name), and namelistname is the name of the namelist being opened.

A profile prefixed by the queue manager name controls access to a single namelist on that queue manager. A profile prefixed by the queue-sharing group name controls access to access to one or more namelists with that name on all queue managers within the queue-sharing group. This access can be overridden on an individual queue manager by defining a queue-manager level profile for that namelist on that queue manager.

If your queue manager is a member of a queue-sharing group and you are using both queue manager and queue-sharing group level security, WebSphere MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue-sharing group name.

The following table shows the access required for opening a namelist.

*Table 44. Access levels for namelist security*

MQOPEN option	RACF access level required to hlq.namelistname
MQOO_INQUIRE	READ

For example, on queue manager (or queue-sharing group) PQM3, the RACF group DEPT571 must be able to inquire (**MQINQ**) on these namelists:

- All namelists starting with “DEPT571”.
- PRINTER/DESTINATIONS/DEPT571
- AGENCY/REQUEST/QUEUES
- WAREHOUSE.BROADCAST

The RACF definitions to do this are:

```
RDEFINE MQNLIST PQM3.DEPT571.** UACC(NONE)
PERMIT PQM3.DEPT571.** CLASS(MQNLIST) ID(DEPT571) ACCESS(READ)

RDEFINE GMQNLIST NLISTS.FOR.DEPT571 UACC(NONE)
  ADDMEM(PQM3.PRINTER/DESTINATIONS/DEPT571,
        PQM3.AGENCY/REQUEST/QUEUES,
        PQM3.WAREHOUSE.BROADCAST)
PERMIT NLISTS.FOR.DEPT571 CLASS(GMQNLIST) ID(DEPT571) ACCESS(READ)
```

Alternate user security might be active, depending on the options specified when a namelist object is opened.

## Profiles for alternate user security

If alternate user security is active, you must:

- Define profiles in the MQADMIN or GMQADMIN classes if you are using uppercase profiles.
- Define profiles in the MXADMIN or GMXADMIN classes if you are using mixed case profiles.

Permit the necessary groups or user IDs access to these profiles, so that they can use the ALTERNATE\_USER\_AUTHORITY options when the object is opened.

Profiles for alternate user security can be specified at subsystem level or at queue-sharing group level and take the following form:

hlq.ALTERNATE.USER.alternateuserid

where hlq can be either qmgr-name (queue manager name) or qsg-name (queue-sharing group name), and alternateuserid is the value of the *AlternateUserId* field in the object descriptor.

A profile prefixed by the queue manager name controls use of an alternate user ID on that queue manager. A profile prefixed by the queue-sharing group name controls use of an alternate user ID on all queue managers within the queue-sharing group. This alternate user ID can be used on any queue manager within the queue-sharing group by a user that has the correct access. This access can be overridden on an individual queue manager by defining a queue-manager level profile for that alternate user ID on that queue manager.

If your queue manager is a member of a queue-sharing group and you are using both queue manager and queue-sharing group level security, WebSphere MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue-sharing group name.

The following table shows the access when specifying an alternate user option.

*Table 45. Access levels for alternate user security*

MQOPEN, MQSUB, or MQPUT1 option	RACF access level required
MQOO_ALTERNATE_USER_AUTHORITY MQSO_ALTERNATE_USER_AUTHORITY MQPMO_ALTERNATE_USER_AUTHORITY	UPDATE

In addition to alternate user security checks, other security checks for queue, process, namelist, and context security can also be made. The alternate user ID, if provided, is only used for security checks on queue, process definition, or namelist resources. For alternate user and context security checks, the user ID requesting the check is used. For details about how user IDs are handled, see “User IDs for security checking” on page 170. For a summary table showing the open options and the security checks required when queue, context and alternate user security are all active, see Table 42 on page 143.

An alternate user profile gives the requesting user ID access to resources associated with the user ID specified in the alternate user ID. For example, the payroll server running under user ID PAYSERV on queue manager QMPY processes requests from personnel user IDs, all of which start with PS. To cause the work performed by the payroll server to be carried out under the user ID of the requesting user, alternate user authority is used. The payroll server knows which user ID to specify as the alternate user ID because the requesting programs generate messages using the MQPMO\_DEFAULT\_CONTEXT put message option. See “User IDs for security checking” on page 170 for more details about from where alternate user IDs are obtained.

The following example RACF definitions enable the server program to specify alternate user IDs starting with the characters PS:

```
RDEFINE MQADMIN QMPY.ALTERNATE.USER.PS* UACC(NONE)
PERMIT QMPY.ALTERNATE.USER.PS* CLASS(MQADMIN) ID(PAYSERV) ACCESS(UPDATE)
```

**Note:**

1. The *AlternateUserId* fields in the object descriptor and subscription descriptor are 12 bytes long. All 12 bytes are used in the profile checks, but only the first eight bytes are used as the user ID by WebSphere MQ. If this user ID truncation is not desirable, application programs making the request should translate any alternate user ID over 8 bytes into something more appropriate.
2. If you specify MQOO\_ALTERNATE\_USER\_AUTHORITY, MQSO\_ALTERNATE\_USER\_AUTHORITY, or MQPMO\_ALTERNATE\_USER\_AUTHORITY and you do not specify an *AlternateUserId* field in the object descriptor, a user ID of blanks is used. For the purposes of the alternate user security check the user ID used for the *AlternateUserId* qualifier is -BLANK-. For example RDEF MQADMIN hlq.ALTERNATE.USER.-BLANK-.

If the user is allowed to access this profile, all further checks are made with a user ID of blanks. For details of blank user IDs, see “Blank user IDs and UACC levels” on page 178.

The administration of alternate user IDs is easier if you have a naming convention for user IDs that enables you to use generic alternate user profiles. If they do not, you could use the RACF RACVARS feature. For details about using RACVARS, see the *z/OS SecureWay Security Server RACF Security Administrator's Guide*.

When a message is put to a queue that has been opened with alternate user authority and the context of the message has been generated by the queue manager, the MQMD\_USER\_IDENTIFIER field is set to the alternate user ID.

## Profiles for context security

If context security is active, you must:

- Define a profile in the MQADMIN class if using uppercase profiles.
- Define profile in the MXADMIN class if using mixed case profiles.

The profile is called hlq.CONTEXT.queueName, where:

**hlq** Can be either qmgr-name (queue manager name) or qsg-name (queue-sharing group name).

**queueName**

Can be either the full name of the queue you want to define the context profile for, or a generic profile.

Special considerations apply if you are migrating from a previous version; see *WebSphere MQ Migration Information*.

A profile prefixed by the queue manager name, and with \*\* specified as the queue name, allows control for context security on all queues belonging to that queue manager. This can be overridden on an individual queue by defining a queue level profile for context on that queue.

A profile prefixed by the queue-sharing group name, and with \*\* specified as the queue name, allows control for context on all queues belonging to the queue

managers within the queue-sharing group. This can be overridden on an individual queue manager by defining a queue-manager level profile for context on that queue manager, by specifying a profile prefixed by the queue manager name. It can also be overridden on an individual queue by specifying a profile suffixed with the queue name.

If your queue manager is a member of a queue-sharing group and you are using both queue manager and queue-sharing group level security, WebSphere MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue-sharing group name.

You must give the necessary groups or user IDs access to this profile. The following table shows the access level required, depending on the specification of the context options when the queue is opened.

*Table 46. Access levels for context security*

<b>MQOPEN or MQPUT1 option</b>	<b>RACF access level required to hlq.CONTEXT.queueName</b>
MQPMO_NO_CONTEXT	No context security check
MQPMO_DEFAULT_CONTEXT	No context security check
MQOO_SAVE_ALL_CONTEXT	No context security check
MQOO_PASS_IDENTITY_CONTEXT MQPMO_PASS_IDENTITY_CONTEXT	READ
MQOO_PASS_ALL_CONTEXT MQPMO_PASS_ALL_CONTEXT	READ
MQOO_SET_IDENTITY_CONTEXT MQPMO_SET_IDENTITY_CONTEXT	UPDATE
MQOO_SET_ALL_CONTEXT MQPMO_SET_ALL_CONTEXT	CONTROL
MQOO_OUTPUT or MQPUT1 (USAGE(XMITQ))	CONTROL
<b>MQSUB option</b>	
MQSO_SET_IDENTITY_CONTEXT( <b>Note 2</b> )	UPDATE
<b>Note:</b>	
<ol style="list-style-type: none"> <li>1. The user IDs used for distributed queuing require CONTROL access to hlq.CONTEXT.queueName to put messages on the destination queue. See "User IDs used by the channel initiator" on page 173 for information about the user IDs used.</li> <li>2. If on the MQSUB request, with MQSO_CREATE or MQSO_ALTER options specified, you want to set any of the identity context fields in the MQSD structure, you need to specify the MQSO_SET_IDENTITY_CONTEXT option. You require also, the appropriate authority to the context profile for the destination queue.</li> </ol>	

If you put commands on the system-command input queue, use the default context put message option to associate the correct user ID with the command.

For example, the WebSphere MQ-supplied utility program CSQUTIL can be used to off-load and reload messages in queues. When off-loaded messages are restored to a queue, the CSQUTIL utility uses the MQOO\_SET\_ALL\_CONTEXT option to return the messages to their original state. In addition to the queue security required by this open option, context authority is also required. For example, if this authority is required by the group BACKGRP on queue manager MQS1, this would be defined by:



```
RDEFINE MQADMIN MQS1.CONTEXT.** UACC(NONE)
PERMIT MQS1.CONTEXT.** CLASS(MQADMIN) ID(BACKGRP) ACCESS(CONTROL)
```

Depending on the options specified, and the types of security performed, other types of security checks might also occur when the queue is opened. These include queue security (see “Profiles for queue security” on page 135), and alternate user security (see “Profiles for alternate user security” on page 147). For a summary table showing the open options and the security checks required when queue, context and alternate user security are all active, see Table 42 on page 143.

## Profiles for command security

If you want security checking for commands (so you have not defined the command security switch profile `hlq.NO.CMD.CHECKS`) you must add profiles to the MQCMDS class.

The same security profiles control both MQSC and PCF commands. The names of the RACF profiles for command security checking are based on the MQSC command names themselves. These profiles take the form:

```
hlq.verb.pkw
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue-sharing group name).

For example, the profile name for the `ALTER QLOCAL` command in subsystem `CSQ1` is:

```
CSQ1.ALTER.QLOCAL
```

A profile prefixed by the queue manager name controls the use of the command on that queue manager. A profile prefixed by the queue-sharing group name controls the use of the command on all queue managers within the queue-sharing group. This access can be overridden on an individual queue manager by defining a queue-manager level profile for that command on that queue manager.

If your queue manager is a member of a queue-sharing group and you are using both queue manager and queue-sharing group level security, WebSphere MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue-sharing group name.

By setting up command profiles at queue manager level, a user can be restricted from issuing commands on a particular queue manager. Alternatively, you can define one profile for a queue-sharing group for each command verb, and all security checks take place against that profile instead of individual queue managers.

If both subsystem security and queue-sharing group security are active and a local profile is not found, a command security check is performed to see if the user has access to a queue-sharing group profile.

If you use the CMDSCOPE attribute to route a command to other queue managers in a queue-sharing group, security is checked on each queue manager where the command is executed, but not necessarily on the queue manager where the command is entered.

Table 47 shows, for each WebSphere MQ MQSC command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

Table 48 on page 155 shows, for each WebSphere MQ PCF command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

*Table 47. MQSC commands, profiles, and their access levels*

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
ALTER AUTHINFO	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
ALTER BUFFPOOL	hlq.ALTER.BUFFPOOL	ALTER	No check	–
ALTER CFSTRUCT	hlq.ALTER.CFSTRUCT	ALTER	No check	–
ALTER CHANNEL	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
ALTER NAMELIST	hlq.ALTER.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
ALTER PROCESS	hlq.ALTER.PROCESS	ALTER	hlq.PROCESS.process	ALTER
ALTER PSID	hlq.ALTER.PSID	ALTER	No check	–
ALTER QALIAS	hlq.ALTER.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
ALTER QLOCAL	hlq.ALTER.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QMGR	hlq.ALTER.QMGR	ALTER	No check	–
ALTER QMODEL	hlq.ALTER.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QREMOTE	hlq.ALTER.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
ALTER SECURITY	hlq.ALTER.SECURITY	ALTER	No check	–
ALTER STGCLASS	hlq.ALTER.STGCLASS	ALTER	No check	–
ALTER SUB	hlq.ALTER.SUB	ALTER	No check	–
ALTER TOPIC	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
ALTER TRACE	hlq.ALTER.TRACE	ALTER	No check	–
ARCHIVE LOG	hlq.ARCHIVE.LOG	CONTROL	No check	–
BACKUP CFSTRUCT	hlq.BACKUP.CFSTRUCT	CONTROL	No check	–
CLEAR QLOCAL	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
CLEAR TOPICSTR	hlq.CLEAR.TOPICSTR	ALTER	hlq.TOPIC.topic	ALTER
DEFINE AUTHINFO	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
DEFINE BUFFPOOL	hlq.DEFINE.BUFFPOOL	ALTER	No check	–
DEFINE CFSTRUCT	hlq.DEFINE.CFSTRUCT	ALTER	No check	–
DEFINE CHANNEL	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
DEFINE MAXSMSGS	hlq.DEFINE.MAXSMSGS	ALTER	No check	–
DEFINE NAMELIST	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER

Table 47. MQSC commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
DEFINE PROCESS	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
DEFINE PSID	hlq.DEFINE.PSID	ALTER	No check	–
DEFINE QALIAS	hlq.DEFINE.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QLOCAL	hlq.DEFINE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QMODEL	hlq.DEFINE.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QREMOTE	hlq.DEFINE.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
DEFINE STGCLASS	hlq.DEFINE.STGCLASS	ALTER	No check	–
DEFINE SUB	hlq.DEFINE.SUB	ALTER	No check	–
DEFINE TOPIC	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
DELETE AUTHINFO	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
DELETE BUFFPOOL	hlq.DELETE.BUFFPOOL	ALTER	No check	–
DELETE CFSTRUCT	hlq.DELETE.CFSTRUCT	ALTER	No check	–
DELETE CHANNEL	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
DELETE NAMELIST	hlq.DELETE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
DELETE PROCESS	hlq.DELETE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
DELETE PSID	hlq.DELETE.PSID	ALTER	No check	–
DELETE QALIAS	hlq.DELETE.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
DELETE QLOCAL	hlq.DELETE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
DELETE QMODEL	hlq.DELETE.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
DELETE QREMOTE	hlq.DELETE.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
DELETE STGCLASS	hlq.DELETE.STGCLASS	ALTER	No check	–
DELETE SUB	hlq.DELETE.SUB	ALTER	No check	–
DELETE TOPIC	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
DISPLAY ARCHIVE <sup>(1)</sup>	hlq.DISPLAY.ARCHIVE	READ	No check	–
DISPLAY AUTHINFO	hlq.DISPLAY.AUTHINFO	READ	No check	–
DISPLAY CFSTATUS	hlq.DISPLAY.CFSTATUS	READ	No check	–
DISPLAY CFSTRUCT	hlq.DISPLAY.CFSTRUCT	READ	No check	–
DISPLAY CHANNEL	hlq.DISPLAY.CHANNEL	READ	No check	–
DISPLAY CHINIT	hlq.DISPLAY.CHINIT	READ	No check	–
DISPLAY CHSTATUS	hlq.DISPLAY.CHSTATUS	READ	No check	–
DISPLAY CLUSQMGR	hlq.DISPLAY.CLUSQMGR	READ	No check	–
DISPLAY CMDSERV	hlq.DISPLAY.CMDSERV	READ	No check	–
DISPLAY CONN <sup>(1)</sup>	hlq.DISPLAY.CONN	READ	No check	–
DISPLAY GROUP	hlq.DISPLAY.GROUP	READ	No check	–
DISPLAY LOG <sup>(1)</sup>	hlq.DISPLAY.LOG	READ	No check	–
DISPLAY MAXSMSGS	hlq.DISPLAY.MAXSMSGS	READ	No check	–
DISPLAY NAMELIST	hlq.DISPLAY.NAMELIST	READ	No check	–

Table 47. MQSC commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
DISPLAY PROCESS	hlq.DISPLAY.PROCESS	READ	No check	–
DISPLAY PUBSUB	hlq.DISPLAY.PUBSUB	READ	No check	–
DISPLAY QALIAS	hlq.DISPLAY.QALIAS	READ	No check	–
DISPLAY QCLUSTER	hlq.DISPLAY.QCLUSTER	READ	No check	–
DISPLAY QLOCAL	hlq.DISPLAY.QLOCAL	READ	No check	–
DISPLAY QMGR	hlq.DISPLAY.QMGR	READ	No check	–
DISPLAY QMODEL	hlq.DISPLAY.QMODEL	READ	No check	–
DISPLAY QREMOTE	hlq.DISPLAY.QREMOTE	READ	No check	–
DISPLAY QSTATUS	hlq.DISPLAY.QSTATUS	READ	No check	–
DISPLAY QUEUE	hlq.DISPLAY.QUEUE	READ	No check	–
DISPLAY SBSTATUS	hlq.DISPLAY.SBSTATUS	READ	No check	–
DISPLAY SUB	hlq.DISPLAY.SUB	READ	No check	–
DISPLAY SECURITY	hlq.DISPLAY.SECURITY	READ	No check	–
DISPLAY STGCLASS	hlq.DISPLAY.STGCLASS	READ	No check	–
DISPLAY SYSTEM <sup>(1)</sup>	hlq.DISPLAY.SYSTEM	READ	No check	–
DISPLAY THREAD	hlq.DISPLAY.THREAD	READ	No check	–
DISPLAY TPSTATUS	hlq.DISPLAY.TPSTATUS	READ	No check	–
DISPLAY TOPIC	hlq.DISPLAY.TOPIC	READ	No check	–
DISPLAY TPSTATUS	hlq.DISPLAY.TPSTATUS	READ	No check	–
DISPLAY TRACE	hlq.DISPLAY.TRACE	READ	No check	–
DISPLAY USAGE <sup>(1)</sup>	hlq.DISPLAY.USAGE	READ	No check	–
MOVE QLOCAL	hlq.MOVE.QLOCAL	ALTER	hlq.QUEUE.from-queue hlq.QUEUE.to-queue	ALTER
PING CHANNEL	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
RECOVER BSDS	hlq.RECOVER.BSDS	CONTROL	No check	–
RECOVER CFSTRUCT	hlq.RECOVER.CFSTRUCT	CONTROL	No check	–
REFRESH CLUSTER	hlq.REFRESH.CLUSTER	ALTER	No check	–
REFRESH QMGR	hlq.REFRESH.QMGR	ALTER	No check	–
REFRESH SECURITY	hlq.REFRESH.SECURITY	ALTER	No check	–
RESET CHANNEL	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
RESET CLUSTER	hlq.RESET.CLUSTER	CONTROL	No check	–
RESET QMGR	hlq.RESET.QMGR	CONTROL	No check	–
RESET QSTATS	hlq.RESET.QSTATS	CONTROL	hlq.QUEUE.queue	CONTROL
RESET TPIPE	hlq.RESET.TPIPE	CONTROL	No check	–
RESOLVE CHANNEL	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
RESOLVE INDOUBT	hlq.RESOLVE.INDOUBT	CONTROL	No check	–
RESUME QMGR	hlq.RESUME.QMGR	CONTROL	No check	–

Table 47. MQSC commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
RVERIFY SECURITY	hlq.RVERIFY.SECURITY	ALTER	No check	–
SET ARCHIVE	hlq.SET.ARCHIVE	CONTROL	No check	–
SET LOG	hlq.SET.LOG	CONTROL	No check	–
SET SYSTEM	hlq.SET.SYSTEM	CONTROL	No check	–
START CHANNEL	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
START CHINIT	hlq.START.CHINIT	CONTROL	No check	–
START CMDSERV	hlq.START.CMDSERV	CONTROL	No check	–
START LISTENER	hlq.START.LISTENER	CONTROL	No check	–
START QMGR	None <sup>(2)</sup>	–	–	–
START TRACE	hlq.START.TRACE	CONTROL	No check	–
STOP CHANNEL	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
STOP CHINIT	hlq.STOP.CHINIT	CONTROL	No check	–
STOP CMDSERV	hlq.STOP.CMDSERV	CONTROL	No check	–
STOP LISTENER	hlq.STOP.LISTENER	CONTROL	No check	–
STOP QMGR	hlq.STOP.QMGR	CONTROL	No check	–
STOP TRACE	hlq.STOP.TRACE	CONTROL	No check	–
SUSPEND QMGR	hlq.SUSPEND.QMGR	CONTROL	No check	–

**Notes:**

1. These commands might be issued internally by the queue manager; no authority is checked in these cases.
2. WebSphere MQ does not check the authority of the user who issues the START QMGR command. However, you can use RACF facilities to control access to the START xxxxMSTR command that is issued as a result of the START QMGR command. This is done by controlling access to the MVS.START.STC.xxxxMSTR profile in the RACF operator commands (OPERCMD) class. For details of this, see the *z/OS Secureway Security Server RACF Security Administrator's Guide*. If you use this technique, and an unauthorized user tries to start the queue manager, it terminates with a reason code of 00F30216.

Table 48. PCF commands, profiles, and their access levels

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
Backup CF Structure	hlq.BACKUP.CFSTRUCT	CONTROL	No check	–
Change Authentication Information Object	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Change CF Structure	hlq.ALTER.CFSTRUCT	ALTER	No check	–
Change Channel	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Change Namelist	hlq.ALTER.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Change Process	hlq.ALTER.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Change Queue	hlq.ALTER.QUEUE	ALTER	hlq.QUEUE.queue	ALTER

Table 48. PCF commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
Change Queue Manager	hlq.ALTER.QMGR	ALTER	No check	–
Change Security	hlq.ALTER.SECURITY	ALTER	No check	–
Change Storage Class	hlq.ALTER.STGCLASS	ALTER	No check	–
Change Subscription	hlq.ALTER.SUB	ALTER	No check	–
Change Topic	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Clear Queue	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
Clear Topic String	hlq.CLEAR.TOPICSTR	ALTER	hlq.TOPIC.topic	ALTER
Copy Authentication Information Object	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Copy CF Structure	hlq.DEFINE.CFSTRUCT	ALTER	No check	–
Copy Channel	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Copy Namelist	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Copy Process	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Copy Queue	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Copy Subscription	hlq.DEFINE.SUB	ALTER	No check	–
Copy Storage Class	hlq.DEFINE.STGCLASS	ALTER	No check	–
Copy Topic	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Create Authentication Information Object	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Create CF Structure	hlq.DEFINE.CFSTRUCT	ALTER	No check	–
Create Channel	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Create Namelist	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Create Process	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Create Queue	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Create Storage Class	hlq.DEFINE.STGCLASS	ALTER	No check	–
Create Subscription	hlq.DEFINE.SUB	ALTER	No check	–
Create Topic	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Delete Authentication Information Object	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Delete CF Structure	hlq.DELETE.CFSTRUCT	ALTER	No check	–
Delete Channel	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Delete Namelist	hlq.DELETE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Delete Process	hlq.DELETE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Delete Queue	hlq.DELETE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Delete Storage Class	hlq.DELETE.STGCLASS	ALTER	No check	–
Delete Subscription	hlq.DELETE.SUB	ALTER	No check	–
Delete Topic	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER

Table 48. PCF commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
Inquire Archive	hlq.DISPLAY.ARCHIVE	READ	No check	–
Inquire Authentication Information Object	hlq.DISPLAY.AUTHINFO	READ	No check	–
Inquire Authentication Information Object Names	hlq.DISPLAY.AUTHINFO	READ	No check	–
Inquire CF Structure	hlq.DISPLAY.CFSTRUCT	READ	No check	–
Inquire CF Structure Names	hlq.DISPLAY.CFSTRUCT	READ	No check	–
Inquire CF Structure Status	hlq.DISPLAY.CFSTATUS	READ	No check	–
Inquire Channel	hlq.DISPLAY.CHANNEL	READ	No check	–
Inquire Channel Initiator	hlq.DISPLAY.CHINIT	READ	No check	–
Inquire Channel Names	hlq.DISPLAY.CHANNEL	READ	No check	–
Inquire Channel Status	hlq.DISPLAY.CHSTATUS	READ	No check	–
Inquire Cluster Queue Manager	hlq.DISPLAY.CLUSQMGR	READ	No check	–
Inquire Connection	hlq.DISPLAY.CONN	READ	No check	–
Inquire Group	hlq.DISPLAY.GROUP	READ	No check	–
Inquire Log	hlq.DISPLAY.LOG	READ	No check	–
Inquire Namelist	hlq.DISPLAY.NAMELIST	READ	No check	–
Inquire Namelist Names	hlq.DISPLAY.NAMELIST	READ	No check	–
Inquire Process	hlq.DISPLAY.PROCESS	READ	No check	–
Inquire Process Names	hlq.DISPLAY.PROCESS	READ	No check	–
Inquire Pub/Sub Status	hlq.DISPLAY.PUBSUB	READ	No check	–
Inquire Queue	hlq.DISPLAY.QUEUE	READ	No check	–
Inquire Queue Manager	hlq.DISPLAY.QMGR	READ	No check	–
Inquire Queue Names	hlq.DISPLAY.QUEUE	READ	No check	–
Inquire Queue Status	hlq.DISPLAY.QSTATUS	READ	No check	–
Inquire Security	hlq.DISPLAY.SECURITY	READ	No check	–
Inquire Storage Class	hlq.DISPLAY.STGCLASS	READ	No check	–
Inquire Storage Class Names	hlq.DISPLAY.STGCLASS	READ	No check	–
Inquire Subscription	hlq.INQUIRE.SUB	READ	No check	–
Inquire Subscription Status	hlq.INQUIRE.SBSTATUS	READ	No check	–
Inquire System	hlq.DISPLAY.SYSTEM	READ	No check	–
Inquire Topic	hlq.DISPLAY.TOPIC	READ	No check	–

Table 48. PCF commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
Inquire Topic Names	hlq.DISPLAY.TOPIC	READ	No check	–
Inquire Topic Status	hlq.DISPLAY.TPSTATUS	READ	No check	–
Inquire Usage	hlq.DISPLAY.USAGE	READ	No check	–
Move Queue	hlq.MOVE.QLOCAL	ALTER	hlq.QUEUE.from-queue hlq.QUEUE.to-queue	ALTER
Ping Channel	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Recover CF Structure	hlq.RECOVER.CFSTRUCT	CONTROL	No check	–
Refresh Cluster	hlq.REFRESH.CLUSTER	ALTER	No check	–
Refresh Queue Manager	hlq.REFRESH.QMGR	ALTER	No check	–
Refresh Security	hlq.REFRESH.SECURITY	ALTER	No check	–
Reset Channel	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Reset Cluster	hlq.RESET.CLUSTER	CONTROL	No check	–
Reset Queue Manager	hlq.RESET.QMGR	CONTROL	No check	–
Reset Queue Statistics	hlq.RESET.QSTATS	CONTROL	hlq.QUEUE.queue	CONTROL
Resolve Channel	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Resume Queue Manager	hlq.RESUME.QMGR	CONTROL	No check	–
Resume Queue Manager Cluster	hlq.RESUME.QMGR	CONTROL	No check	–
Reverify Security	hlq.RVERIFY.SECURITY	ALTER	No check	–
Set Archive	hlq.SET.ARCHIVE	CONTROL	No check	–
Set Log	hlq.SET.LOG	CONTROL	No check	–
Set System	hlq.SET.SYSTEM	CONTROL	No check	–
Start Channel	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Start Channel Initiator	hlq.START.CHINIT	CONTROL	No check	–
Start Channel Listener	hlq.START.LISTENER	CONTROL	No check	–
Stop Channel	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Stop Channel Initiator	hlq.STOP.CHINIT	CONTROL	No check	–
Stop Channel Listener	hlq.STOP.LISTENER	CONTROL	No check	–
Suspend Queue Manager	hlq.SUSPEND.QMGR	CONTROL	No check	–
Suspend Queue Manager Cluster	hlq.SUSPEND.QMGR	CONTROL	No check	–

## Profiles for topic security

If topic security is active, you must:

- Define profiles in the MXTOPIC or GMXTOPIC classes.



- Permit the necessary groups or user IDs access to these profiles, so that they can issue WebSphere MQ API requests that use topics.

Profiles for topic security take the form:

```
hlq.SUBSCRIBE.topicname
hlq.PUBLISH.topicname
```

where *hlq* can be either *qmgr-name* (queue manager name) or *qsg-name* (queue-sharing group name), and *topicname* is the name of the topic administration node in the topic tree, associated either with the topic being subscribed to through an MQSUB call, or being published to through an MQOPEN call.

A profile prefixed by the queue manager name controls access to a single topic on that queue manager. A profile prefixed by the queue-sharing group name controls access to access to one or more topics with that topic name on all queue managers within the queue-sharing group. This access can be overridden on an individual queue manager by defining a queue-manager level profile for that topic on that queue manager.

If your queue manager is a member of a queue-sharing group and you are using both queue manager and queue-sharing group level security, WebSphere MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue-sharing group name.

## Subscribe

In order to subscribe to a topic, you need access to both the topic you are trying to subscribe to, and the target queue for the publications.

When you issue an MQSUB request, a security check takes place to ensure that you:

- Are allowed to subscribe to that topic, and also that the target queue (if specified) is opened for output
- Have the appropriate level of access to that target queue.

*Table 49. Access level required for topic security to subscribe*

MQSUB to a topic	RACF access required to <i>hlq.SUBSCRIBE.topicname</i> profile in MXTOPIC class
MQSO_CREATE and MQSO_ALTER	ALTER
MQSO_RESUME	READ
MQSUB - additional authority to non managed destination queues.	RACF access required to <i>hlq.CONTEXT.queueename</i> profile in MQADMIN or MXADMIN class
MQSO_CREATE, MQSO_ALTER, and MQSO_RESUME	UPDATE
	RACF access required to <i>hlq.queueename</i> profile in MQQUEUE or MXQUEUE class
MQSO_CREATE and MQSO_ALTER	UPDATE
	RACF access required to <i>hlq.ALTERNATE.USER.alternateuserid</i> profile in MQADMIN or MXADMIN class

Table 49. Access level required for topic security to subscribe (continued)

MQSO_ALTERNATE_USER	UPDATE
---------------------	--------

## Considerations for managed queues for subscriptions

A security check is carried out to see if you are allowed to subscribe to the topic. However, no security checks are carried out when the managed queue is created, or to determine if you are allowed to 'Put' to this destination queue.

You are not allowed to close delete a managed queue.

The model queues used are:SYSTEM.DURABLE.MODEL.QUEUE and SYSTEM.NDURABLE.MODEL.QUEUE.

The managed queues created from these model queues are of the form SYSTEM.MANAGED.DURABLE.A346EF00367849A0 and SYSTEM.MANAGED.NDURABLE.A346EF0036785EA0 where the last qualifier is unpredictable.

You should not give any user access to these queues. The queues can be protected using generic profiles of the form SYSTEM.MANAGED.DURABLE.\* and SYSTEM.MANAGED.NDURABLE.\* with no authorities granted.

Messages can be retrieved from these queues using the handle returned on the MQSUB request.

If you explicitly issue an MQCLOSE call for a subscription with the MQCO\_REMOVE\_SUB option specified, and you did not create the subscription you are closing under this handle, a security check is performed at the time of closure to ensure that you have the correct authority to perform the operation.

Table 50. Access level required to profiles for topic security for closure of a subscribe operation

MQCLOSE (of a subscription)	RACF access required to <i>hlq.SUBSCRIBE.topicname</i> profile in MXTOPIC class
MQCO_REMOVE_SUB	ALTER

## Publish

In order to publish on a topic you need access to the topic and, if you are using alias queues, to the alias queue as well.

Table 51. Access level required to profiles for topic security for a publish operation

MQOPEN (of a topic)	RACF access required to <i>hlq.PUBLISH.topicname</i> profile in MXTOPIC class
MQOO_OUTPUT or MQPUT1	UPDATE
MQOPEN (Alias queue to topic)	RACF access required to <i>hlq.queueName</i> profile in MQQUEUE or MXQUEUE class for the alias queue
MQOO_OUTPUT or MQPUT1	UPDATE

For details of how topic security operates when an alias queue that resolves to a

topic name is opened for publish, see “Considerations for alias queues that resolve to topics for a publish operation.”

When you consider alias queues used for destination queues for PUT or GET restrictions, see “Considerations for alias queues” on page 138.

If the RACF access level that an application has to a topic security profile is changed, the changes only take effect for any new object handles obtained, (that is, a new MQSUB or MQOPEN) for that topic. Those handles already in existence at the time of the change retain their existing access to the topic. Also, existing subscribers retain their access to any subscriptions they have already made.

### **Considerations for alias queues that resolve to topics for a publish operation**

When you issue an MQOPEN or MQPUT1 call for an alias queue that resolves to a topic, WebSphere MQ makes two resource checks:

- The first one against the alias queue name specified in the object descriptor (MQOD) on the MQOPEN or MQPUT1 call.
- The second against the topic to which the alias queue resolves

You need to be aware that this is different from the behavior you get when alias queues resolve to other queues. You need the correct access to both profiles in order for the publish action to proceed.

## **Profiles for command resource security**

If you have not defined the command resource security switch profile, `hlq.NO.CMD.RESC.CHECKS`, because you want security checking for resources associated with commands, you must:

- Add a resource profile in the MQADMIN class, if using uppercase profiles, for each resource.
- Add a resource profile in the MXADMIN class, if using mixed case profiles, for each resource.

The same security profiles control both MQSC and PCF commands.

Profiles for command resource security checking take the form:

<code>hlq.type.resourcename</code>
------------------------------------

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue-sharing group name).

A profile prefixed by the queue manager name controls access to the resources associated with commands on that queue manager. A profile prefixed by the queue-sharing group name controls access to the resources associated with commands on all queue managers within the queue-sharing group. This access can be overridden on an individual queue manager by defining a queue-manager level profile for that command resource on that queue manager.

If your queue manager is a member of a queue-sharing group and you are using both queue manager and queue-sharing group level security, WebSphere MQ

checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue-sharing group name.

For example, the RACF profile name for command resource security checking against the model queue CREDIT.WORTHY in subsystem CSQ1 is:

CSQ1.QUEUE.CREDIT.WORTHY
--------------------------

Because the profiles for all types of command resource are held in the MQADMIN class, the “type” part of the profile name is needed in the profile to distinguish between resources of different types that have the same name. The “type” part of the profile name can be CHANNEL, QUEUE, TOPIC, PROCESS, or NAMELIST. For example, a user might be authorized to define hlq.QUEUE.PAYROLL.ONE, but not authorized to define hlq.PROCESS.PAYROLL.ONE

If the resource type is a queue, and the profile is a queue-sharing group level profile, it controls access to one or more local queues within the queue sharing group, or access to a single shared queue from any queue manager in the queue-sharing group.

Table 47 on page 152 shows, for each WebSphere MQ MQSC command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

Table 48 on page 155 shows, for each WebSphere MQ PCF command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

## Command resource security checking for alias queues

When you define an alias queue, command resource security checks are only performed against the name of the alias queue, not against the name of the target queue to which the alias resolves.

Alias queues can resolve to both local and remote queues. If you do not want to permit users access to certain local or remote queues, you must do both of the following:

1. Do not allow the users access to these local and remote queues.
2. Restrict the users from being able to define aliases for these queues. That is, prevent them from being able to issue DEFINE QALIAS and ALTER QALIAS commands.

## Command resource security checking for remote queues

When you define a remote queue, command resource security checks are performed only against the name of the remote queue. No checks are performed against the names of the queues specified in the RNAME or XMITQ attributes in the remote queue object definition. For more information about the attributes of queues, see the WebSphere MQ Script (MQSC) Command Reference manual.

---

## Using the RESLEVEL security profile

You can define a special profile in the MQADMIN class to control the number of user IDs checked for API-resource security. How this RESLEVEL profile affects API-resource security depends on how you are accessing WebSphere MQ.

This chapter discusses the following subjects:

- “RESLEVEL and batch connections” on page 164
- “RESLEVEL and system functions” on page 165
- “RESLEVEL and CICS connections” on page 165
- “RESLEVEL and IMS connections” on page 166
- “RESLEVEL and channel initiator connections” on page 167
- “RESLEVEL and intra-group queuing” on page 168
- “The RESLEVEL profile”

### Important notes about using RESLEVEL:

1. RESLEVEL is a very powerful option; it can cause the bypassing of all resource security checks for a particular connection. This means that RACF cannot audit these resource checks.
2. You can use the RESAUDIT system parameter to switch RESLEVEL auditing off.
3. Using the RESLEVEL profile means that normal security audit records are not taken. For example, if you put UAUDIT on a user, the access to the hlq.RELEVEL profile in MQADMIN is not audited.
4. If you use the RACF WARNING option on the hlq.RELEVEL profile, no RACF warning messages are produced for profiles in the RESLEVEL class.
5. If you do not have a RESLEVEL profile defined, you must be careful that no other profile in the MQADMIN class matches hlq.RELEVEL. For example, if you have a profile in MQADMIN called hlq.\*\* and no hlq.RELEVEL profile, beware of the consequences of the hlq.\*\* profile because it is used for the RESLEVEL check.

You should define an hlq.RELEVEL profile and set the UACC to NONE, rather than not have a RESLEVEL profile at all. You should have as few users or groups in the access list as possible. For details about how to audit RESLEVEL access, see “Auditing considerations” on page 188.

6. If you make any changes to the RESLEVEL profile users must disconnect and connect again before the change takes place. (This includes stopping and restarting the channel initiator if the access that the distributed queuing address space user ID has to the RESLEVEL profile is changed.)
7. You can define a special profile in the MQADMIN class, if using uppercase profiles, or the MXADMIN class, if you are using mixed case profiles, to control the number of user IDs checked for API-resource security. How this RESLEVEL profile affects API-resource security depends on how you are accessing WebSphere MQ.

## The RESLEVEL profile

When an application tries to connect to WebSphere MQ, WebSphere MQ checks the access that the user ID associated with the connection has to a profile in the MQADMIN class if you are using uppercase profiles, or MXADMIN if you are

using mixed case profiles, called:

hlq.RESLEVEL
--------------

where hlq can be either ssid (subsystem ID) or qsg (queue-sharing group ID).

The user IDs associated with each connection type are:

- The user ID of the connecting task for batch connections
- The CICS address space user ID for CICS connections
- The IMS region address space user ID for IMS connections
- The channel initiator address space user ID for channel initiator connections

If you are using queue manager level security only, WebSphere MQ performs RESLEVEL checks against the qmgr-name.RESLEVEL profile. If you are using queue-sharing group level security only, WebSphere MQ performs RESLEVEL checks against the qsg-name.RESLEVEL profile. If you are using a combination of both queue manager and queue-sharing group level security, WebSphere MQ first checks for the existence of a RESLEVEL profile at queue manager level. If it does not find one, it checks for a RESLEVEL profile at queue-sharing group level.

If it cannot find a RESLEVEL profile, WebSphere MQ enables checking of both the job and task (or alternate user) ID for a CICS or an IMS connection. For a batch connection, WebSphere MQ enables checking of the job (or alternate) user ID. For the channel initiator, WebSphere MQ enables checking of the channel user ID and the MCA (or alternate) user ID.

If there is a RESLEVEL profile, the level of checking depends on the environment and access level for the profile.

Remember that if your queue manager is a member of a queue-sharing group and you do not define this profile at queue-manager level, there might be one defined at queue-sharing group level that will effect the level of checking. To activate the checking of two user IDs, you should define a RESLEVEL profile (prefixed with either the queue manager name of the queue-sharing group name) with a UACC(NONE) and ensure that the relevant users do not have access granted against this profile.

## RESLEVEL and batch connections

By default, when a WebSphere MQ resource is being accessed through batch and batch-type connections, the user must be authorized to access that resource for the particular operation. You can bypass the security check by setting up an appropriate RESLEVEL definition.

Whether the user is checked or not is based on the user ID used at connect time, the same user ID used for the connection check.

For example, you can set up RESLEVEL so that when a user you trust accesses certain resources through a batch connection, no API-resource security checks are done; but when a user you do not trust tries to access the same resources, security checks are carried out as normal. You should set up RESLEVEL checking to bypass API-resource security checks only when you sufficiently trust the user and the programs run by that user.

The following table shows the checks made for batch connections.

Table 52. Checks made at different RACF access levels for batch connections

RACF access level	Level of checking
NONE	Resource checks performed
READ	Resource checks performed
UPDATE	Resource checks performed
CONTROL	No check.
ALTER	No check.

## RESLEVEL and system functions

The application of RESLEVEL to the operation and control panels, and to CSQUTIL.

The operation and control panels and the CSQUTIL utility are batch-type applications that make requests to the queue manager's command server, and so they are subject to the considerations described in "RESLEVEL and batch connections" on page 164. You can use RESLEVEL to bypass security checking for the SYSTEM.COMMAND.INPUT and SYSTEM.COMMAND.REPLY.MODEL queues that they use, but not for the dynamic queues SYSTEM.CSQXCMD.\*, SYSTEM.CSQOREXX.\*, and SYSTEM.CSQUTIL.\*.

The command server is an integral part of the queue manager and so does not have connection or RESLEVEL checking associated with it. To maintain security, therefore, the command server must confirm that the user ID of the requesting application has authority to open the queue being used for replies. For the operations and control panels this is SYSTEM.CSQOREXX.\*. For CSQUTIL it is SYSTEM.CSQUTIL.\*. Users must be authorized to use these queues, as described in "System queue security" on page 142, in addition to any RESLEVEL authorization they are given.

For other applications using the command server it is the queue they name as their reply-to queue. Such other applications might deceive the command server into placing messages on unauthorized queues by passing (in the message context) a more trusted user ID than its own to the command server. To prevent this, use a CONTEXT profile to protect the identity context of messages placed on SYSTEM.COMMAND.INPUT.

## RESLEVEL and CICS connections

By default, when an API-resource security check is made on a CICS connection, two user IDs are checked to see if access is allowed to the resource.

### User IDs checked

The first user ID checked is that of the CICS address space. This is the user ID on the job card of the CICS job, or the user ID assigned to the CICS started task by the z/OS STARTED class or the started procedures table. (It is not the CICS DFLTUSER.)

The second user ID checked is the user ID associated with the CICS transaction.

## Completion codes

If one of these user IDs does not have access to the resource, the request fails with a completion code of MQRC\_NOT\_AUTHORIZED. Both the CICS address space user ID and the user ID of the person running the CICS transaction must have access to the resource at the correct level.

## How RESLEVEL can affect the checks made

Depending on how you set up your RESLEVEL profile, you can change which user IDs are checked when access to a resource is requested. The possible checks are:

- Check the CICS address space user ID and the transaction user ID.
- Check the CICS address space user ID only.
- If the transaction is defined to CICS with RESSEC(NO), check the CICS address space user ID only. (The status of the CICS security is NOT checked when taking into consideration the transaction RESSEC setting. For example, if CICS has been started with SEC=NO, but the transaction has been defined with RESSEC(YES), WebSphere MQ still checks both user IDs.)
- If the transaction is defined to CICS with RESSEC(YES), check the CICS address space user ID and the transaction user ID.
- Do not check any user IDs.

The user IDs checked depend on the user ID used at connection time, that is, the CICS address space user ID. This control enables you to bypass API-resource security checking for WebSphere MQ requests coming from one system (for example, a test system, TESTCICS,) but to implement them for another (for example, a production system, PRODCICS).

**Note:** If you set up your CICS address space user ID with the “trusted” attribute in the STARTED class or the RACF started procedures table ICHRIN03, this overrides any user ID checks for the CICS address space established by the RESLEVEL profile for your queue manager (that is, the queue manager does not perform the security checks for the CICS address space). For more information, see the *CICS Transaction Server for OS/390 V1.3 CICS RACF Security Guide*.

The following table shows the checks made for CICS connections.

*Table 53. Checks made at different RACF access levels for CICS connections*

RACF access level	Level of checking
NONE	Check the CICS address space user ID and the task or alternate user ID.
READ	Check the CICS address space user ID.
UPDATE	Check the CICS address space user ID and, if the transaction has been defined with RESSEC=YES, also check the task or alternate user ID.
CONTROL	No check.
ALTER	No check.

## RESELEVEL and IMS connections

By default, when an API-resource security check is made for an IMS connection, two user IDs are checked to see if access is allowed to the resource.



The first user ID checked is that of the address space of the IMS region. This is taken from either the USER field from the job card or the user ID assigned to the region from the z/OS STARTED class or the started procedures table (SPT).

The second user ID checked is associated with the work being done in the dependent region. It is determined according to the type of the dependent region as shown in Table 60 on page 173.

The setting of WebSphere MQ RESLEVEL profiles cannot alter the user ID under which IMS transactions are scheduled from the IBM-supplied MQ-IMS trigger monitor program CSQQTRMN. This user ID is the PSBNAME of that trigger monitor, which by default is CSQQTRMN.

### Completion codes

If either the first or second IMS user ID does not have access to the resource, the request fails with a completion code of MQRC\_NOT\_AUTHORIZED.

### How RESLEVEL can affect the checks made

Depending on how you set up your RESLEVEL profile, you can change which user IDs are checked when access to a resource is requested. The possible checks are:

- Check the IMS region address space user ID and the second user ID or alternate user ID.
- Check IMS region address space user ID only.
- Do not check any user IDs.

The following table shows the checks made for IMS connections.

*Table 54. Checks made at different RACF access levels for IMS connections*

RACF access level	Level of checking
NONE	Check the IMS address space user ID and the IMS second user ID or alternate user ID.
READ	Check the IMS address space user ID.
UPDATE	Check the IMS address space user ID.
CONTROL	No check.
ALTER	No check.

## RESLEVEL and channel initiator connections

By default, when an API-resource security check is made by the channel initiator, two user IDs are checked to see if access is allowed to the resource.

The user IDs checked can be that specified by the MCAUSER channel attribute, that received from the network, that of the channel initiator address space, or the alternate user ID for the message descriptor. This depends on the communication protocol you are using and the setting of the PUTAUT channel attribute. See “User IDs used by the channel initiator” on page 173 for more information.

### Completion codes

If one of these user IDs does not have access to the resource, the request fails with a completion code of MQRC\_NOT\_AUTHORIZED.

## How RESLEVEL can affect the checks made

Depending on how you set up your RESLEVEL profile, you can change which user IDs are checked when access to a resource is requested, and how many are checked.

The following table shows the checks made for channel initiator connections.

Table 55. Checks made at different RACF access levels for channel initiator connections

RACF access level	Level of checking
NONE	Check two user IDs.
READ	Check one user ID.
UPDATE	Check one user ID.
CONTROL	No check.
ALTER	No check.
<b>Note:</b> See "User IDs used by the channel initiator" on page 173 for a definition of the user IDs checked	

## RESLEVEL and intra-group queuing

By default, when an API-resource security check is made by the intra-group queuing agent, two user IDs are checked to see if access is allowed to the resource.

The user IDs checked can be the user ID determined by the IGQUSER attribute of the receiving queue manager, the user ID of the queue manager within the queue-sharing group that put the message on to the SYSTEM.QSG.TRANSMIT.QUEUE, or the alternate user ID specified in the *UserIdentifier* field of the message descriptor of the message. See "User IDs used by the intra-group queuing agent" on page 177 for more information.

Because the intra-group queuing agent is an internal queue manager task, it does not issue an explicit connect request and runs under the user ID of the queue manager. The intra-group queuing agent starts at queue manager initialization. During the initialization of the intra-group queuing agent, WebSphere MQ checks the access that the user ID associated with the queue manager has to a profile in the MQADMIN class called:

hlq.RESLEVEL

This check is always performed unless the hlq.NO.SUBSYS.SECURITY switch has been set.

If there is no RESLEVEL profile, WebSphere MQ enables checking for two user IDs. If there is a RESLEVEL profile, the level of checking depends on the access level granted to the user ID of the queue manager for the profile. Table 56 shows the checks made for the intra-group queuing agent.

Table 56. Checks made at different RACF access levels for the intra-group queuing agent

RACF access level	Level of checking
NONE	Check two user IDs.

Table 56. Checks made at different RACF access levels for the intra-group queuing agent (continued)

RACF access level	Level of checking
READ	Check one user ID.
UPDATE	Check one user ID.
CONTROL	No check.
ALTER	No check.
<b>Note:</b> See “User IDs used by the intra-group queuing agent” on page 177 for a definition of the user IDs checked	

If the permissions granted to the RESLEVEL profile for the queue manager’s user ID are changed, the intra-group queuing agent must be stopped and restarted to pick up the new permissions. Because there is no way to independently stop and restart the intra-group queuing agent, the queue manager must be stopped and restarted to achieve this.

## RESLEVEL and the user IDs checked

Table 57 on page 171 through Table 63 on page 176 show how RESLEVEL affects which user IDs are checked for different MQI requests.

For example, you have a queue manager called QM66, where:

- User WS21B is to be exempt from resource security.
- CICS started task WXNCICS running under address space user ID CICSWXN is to perform full resource checking only for transactions defined with RESSEC(YES).

To define the appropriate RESLEVEL profile, issue the RACF command:

```
RDEFINE MQADMIN QM66.RESLEVEL UACC(NONE)
```

Then give the users access to this profile:

```
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(WS21B) ACCESS(CONTROL)
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(CICSWXN) ACCESS(UPDATE)
```

If you make these changes while the user IDs are connected to queue manager QM66, the users must disconnect and connect again before the change takes place.

If subsystem security is not active when a user connects but, while this user is still connected, subsystem security becomes active, full resource security checking is applied to the user. The user must reconnect to get the correct RESLEVEL processing.

## User IDs for security checking

WebSphere MQ initiates security checks based on user IDs associated with users, terminals, applications, and so on. The following sections show the contents of the user IDs used for each type of security check.

This chapter discusses the following topics:

- “User IDs for connection security”
- “User IDs for command security and command resource security”
- “User IDs for resource security (MQOPEN, MQSUB, and MQPUT1)” on page 171
- “Blank user IDs and UACC levels” on page 178

### User IDs for connection security

Connection type	User ID contents
Batch connection	The user ID of the connecting task. For example: <ul style="list-style-type: none"><li>• The TSO user ID</li><li>• The user ID assigned to a batch job by the USER JCL parameter</li><li>• The user ID assigned to a started task by the STARTED class or the started procedures table</li></ul>
CICS connection	The CICS address space user ID.
IMS connection	The IMS region address space user ID.
Channel initiator connection	The channel initiator address space user ID.

### User IDs for command security and command resource security

Issued from...	User ID contents
CSQINP1 or CSQINP2	No check is made.
System command input queue	The user ID found in the <i>UserIdentifier</i> of the message descriptor of the message that contains the command. If the message does not contain a <i>UserIdentifier</i> , a user ID of blanks is passed to the security manager.
Console	The user ID signed onto the console. If the console is not signed on, the default user ID set by the CMDUSER system parameter in CSQ6SYSP.  To issue commands from a console, the console must have the z/OS SYS AUTHORITY attribute.
SDSF/TSO console	TSO or job user ID.
Operations and control panels	TSO user ID.  If you are going to use the operations and control panels, you must have the appropriate authority to issue the commands corresponding to the actions that you choose. In addition, you must have READ access to all the hlq.DISPLAY. <i>object</i> profiles in the MQCMDS class because the panels use the various DISPLAY commands to gather the information that they present.

Issued from...	User ID contents
MGCRE	If MGCRE is used with Utoken, the user ID in the Utoken.  If MGCRE is issued without the Utoken, the TSO or job user ID is used.
CSQUTIL	Job user ID.
CSQINPX	User ID of the channel initiator address space.

## User IDs for resource security (MQOPEN, MQSUB, and MQPUT1)

This information shows the contents of the user IDs for normal and alternate user IDs for each type of connection. The number of checks is defined by the RESLEVEL profile. The user ID checked is that used for MQOPEN, MQSUB, or MQPUT1 calls.

**Note:** All user ID fields are checked *exactly* as they are received. No conversions take place, and, for example, three user ID fields containing Bob, BOB, and bob are not equivalent.

### User IDs checked for batch connections

Table 57. User ID checking against profile name for batch connections

Alternate user ID specified on open?	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queueprofile	hlq.resourcename profile
No	–	JOB	JOB
Yes	JOB	JOB	ALT

Key:

**ALT** Alternate user ID.

**JOB**

- The user ID of a TSO or USS sign-on.
- The user ID assigned to a batch job.
- The user ID assigned to a started task by the STARTED class or the started procedures table.
- The user ID associated with the executing DB2 stored procedure

### Batch connection example:

A Batch job is performing an MQPUT1 to a queue called Q1 with RESLEVEL set to READ and alternate user ID checking turned off.

Table 52 on page 165 and Table 57 show that the job user ID is checked against profile hlq.Q1.

### User IDs checked for CICS connections

Table 58. User ID checking against profile name for CICS-type user IDs

Alternate user ID specified on open?	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queueprofile	hlq.resourcename profile
No, 1 check	–	ADS	ADS

Table 58. User ID checking against profile name for CICS-type user IDs (continued)

Alternate user ID specified on open?	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queueprofile	hlq.resourcename profile
No, 2 checks	–	ADS+TXN	ADS+TXN
Yes, 1 check	ADS	ADS	ADS
Yes, 2 checks	ADS+TXN	ADS+TXN	ADS+ALT
Key:			
<b>ALT</b>	Alternate user ID		
<b>ADS</b>	The user ID associated with the CICS batch job or, if CICS is running as a started task, through the STARTED class or the started procedures table.		
<b>TXN</b>	The user ID associated with the CICS transaction. This is normally the user ID of the terminal user who started the transaction. It can be the CICS DFLTUSER, a PRESET security terminal, or a manually signed-on user.		

### CICS example:

Determine the user IDs checked for the following conditions:

- The RACF access level to the RESLEVEL profile, for a CICS address space user ID, is set to NONE.
- An **MQOPEN** call is made against a queue with MQOO\_OUTPUT and MQOO\_PASS\_IDENTITY\_CONTEXT.

*Answer:* First, see how many CICS user IDs are checked based on the CICS address space user ID access to the RESLEVEL profile. From Table 53 on page 166, two user IDs are checked if the RESLEVEL profile is set to NONE. Then, from Table 58 on page 171, these checks are carried out:

- The hlq.ALTERNATE.USER.userid profile is not checked.
- The hlq.CONTEXT.queueprofile profile is checked with both the CICS address space user ID and the CICS transaction user ID.
- The hlq.resourcename profile is checked with both the CICS address space user ID and the CICS transaction user ID.

This means that four security checks are made for this **MQOPEN** call.

### User IDs checked for IMS connections

Table 59. User ID checking against profile name for IMS-type user IDs

Alternate user ID specified on open?	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queueprofile	hlq.resourcename profile
No, 1 check	–	REG	REG
No, 2 checks	–	REG+SEC	REG+SEC
Yes, 1 check	REG	REG	REG
Yes, 2 checks	REG+SEC	REG+SEC	REG+ALT
Key:			
<b>ALT</b>	Alternate user ID.		
<b>REG</b>	The user ID is normally set through the STARTED class or the started procedures table or, if IMS is running, from a submitted job, by the USER JCL parameter.		
<b>SEC</b>	The second user ID is associated with the work being done in a dependent region. It is determined according to Table 60 on page 173.		

Table 60. How the second user ID is determined for the IMS connection

Types of dependent region	Hierarchy for determining the second user ID
<ul style="list-style-type: none"> <li>• BMP message driven and successful GET UNIQUE issued.</li> <li>• IFP and GET UNIQUE issued.</li> <li>• MPP.</li> </ul>	User ID associated with the IMS transaction if the user is signed on.  LTERM name if available.  PSBNAME.
<ul style="list-style-type: none"> <li>• BMP message driven and successful GET UNIQUE not issued.</li> <li>• BMP not message driven.</li> <li>• IFP and GET UNIQUE not issued.</li> </ul>	User ID associated with the IMS dependent region address space if this is not all blanks or all zeros.  PSBNAME.

## User IDs used by the channel initiator

The following sections describe the user IDs used and checked for the following:

- TCP/IP receiving channels.
- LU 6.2 receiving channels.
- Client MQI requests issued over server-connection channels for both TCP/IP and LU 6.2.

You can use the PUTAUT parameter of the receiving channel definition to determine the type of security checking used. To get consistent security checking throughout your WebSphere MQ network, you can use the ONLYMCA and ALTMCA options.

You can use the DISPLAY CHSTATUS command to determine the user identifier used by the MCA. See WebSphere MQ Script (MQSC) Command Reference.

### Receiving channels using TCP/IP:

#### MCA user ID (MCA)

The user ID specified for the MCAUSER channel attribute at the receiver; if blank, the channel initiator address space user ID of the receiver or requester side is used.

#### Channel user ID (CHL)

On TCP/IP, security is not supported by the communication system for the channel. If the Secure Sockets Layer (SSL) is being used and a digital certificate has been flowed from the partner, the user ID associated with this certificate (if installed), or the user ID associated with a matching filter found by using RACF's Certificate Name Filter (CNF), is used. If no associated user ID is found, or if SSL is not being used, the user ID of the channel initiator address space of the receiver or requester end is used as the channel user ID on channels defined with the PUTAUT parameter set to DEF or CTX.

**Note:** The use of RACF's Certificate Name Filter (CNF) allows you to assign the same RACF user ID to multiple remote users, for example all the users in the same organization unit, who would naturally all have the same security authority. This means that the server does not have to have a copy of the certificate of every possible remote end user across the world and greatly simplifies certificate management and distribution.

If the PUTAUT parameter is set to ONLYMCA or ALTMCA for the channel, the channel user ID is ignored and the MCA user ID of the receiver or requester is used. This also applies to TCP/IP channels using SSL.

#### Alternate user ID (ALT)

The user ID from the context information (that is, the *UserIdentifier* field) within the message descriptor of the message. This user ID is moved into the *AlternateUserID* field in the object descriptor before an MQOPEN or MQPUT1 call is issued for the target destination queue.

Table 61. User IDs checked against profile name for TCP/IP channels

PUTAUT option specified on receiver or requester channel	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queueprofile	hlq.resourcename profile
DEF, 1 check	–	CHL	CHL
DEF, 2 checks	–	CHL + MCA	CHL + MCA
CTX, 1 check	CHL	CHL	CHL
CTX, 2 checks	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 check	–	MCA	MCA
ONLYMCA, 2 checks	–	MCA	MCA
ALTMCA, 1 check	MCA	MCA	MCA
ALTMCA, 2 checks	MCA	MCA	MCA + ALT

Key:

ALT Alternate user ID.  
 CHL Channel user ID.  
 MCA MCA user ID.

#### Receiving channels using LU 6.2:

##### MCA user ID (MCA)

The user ID specified for the MCAUSER channel attribute at the receiver; if blank, the channel initiator address space user ID of the receiver or requester side is used.

##### Channel user ID (CHL)

###### Requester-server channels

If the channel is started from the requester, there is no opportunity to receive a network user ID (the channel user ID).

If the PUTAUT parameter is set to DEF or CTX on the requester channel, the channel user ID is that of the channel initiator address space of the requester because no user ID is received from the network.

If the PUTAUT parameter is set to ONLYMCA or ALTMCA, the channel user ID is ignored and the MCA user ID of the requester is used.

###### Other channel types

If the PUTAUT parameter is set to DEF or CTX on the receiver or requester channel, the channel user ID is the user ID received from the communications system when the channel is initiated.



- If the sending channel is on z/OS, the channel user ID received is the channel initiator address space user ID of the sender.
- If the sending channel is on a different platform (for example, AIX® or HP-UX), the channel user ID received is typically provided by the USERID parameter of the channel definition.

If the user ID received is blank, or no user ID is received, a channel user ID of blanks is used.

#### Alternate user ID (ALT)

The user ID from the context information (that is, the *UserIdentifier* field) within the message descriptor of the message. This user ID is moved into the *AlternateUserID* field in the object descriptor before an **MQOPEN** or **MQPUT1** call is issued for the target destination queue.

Table 62. User IDs checked against profile name for LU 6.2 channels

PUTAUT option specified on receiver or requester channel	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queueuname profile	hlq.resourcename profile
DEF, 1 check	-	CHL	CHL
DEF, 2 checks	-	CHL + MCA	CHL + MCA
CTX, 1 check	CHL	CHL	CHL
CTX, 2 checks	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 check	-	MCA	MCA
ONLYMCA, 2 checks	-	MCA	MCA
ALTMCA, 1 check	MCA	MCA	MCA
ALTMCA, 2 checks	MCA	MCA	MCA + ALT
Key:			
ALT	Alternate user ID.		
CHL	Channel user ID.		
MCA	MCA user ID.		

#### Client MQI requests:

This section describes the user IDs checked for client MQI requests issued over server-connection channels for TCP/IP and LU 6.2. The MCA user ID and channel user ID are as for the TCP/IP and LU 6.2 channels described in the previous sections.

For server-connection channels, the user ID received from the client is used if the MCAUSER attribute is blank. However, for the clients that can use the MQ\_USER\_ID environment variable to supply the user ID, it is possible that no environment variable has been set. In this case, the user ID that started the server channel is used. This is the user ID assigned to the channel initiator started task by the z/OS started procedures table.

See the WebSphere MQ Clients manual for more information.

For client **MQOPEN**, **MQSUB**, and **MQPUT1** requests, use the following rules to determine the profile that is checked:

- If the request specifies alternate-user authority, a check is made against the *hlq.ALTERNATE.USER.userid* profile.
- If the request specifies context authority, a check is made against the *hlq.CONTEXT.queueName* profile.
- For all **MQOPEN**, **MQSUB**, and **MQPUT1** requests, a check is made against the *hlq.resourcename* profile.

When you have determined which profiles are checked, use the following table to determine which user IDs are checked against these profiles.

Table 63. User IDs checked against profile name for LU 6.2 and TCP/IP server-connection channels

PUTAUT option specified on server-connection channel	Alternate user ID specified on open?	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queueName profile	hlq.resourcename profile
DEF, 1 check	No	–	CHL	CHL
DEF, 1 check	Yes	CHL	CHL	CHL
DEF, 2 checks	No	–	CHL + MCA	CHL + MCA
DEF, 2 checks	Yes	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 check	No	–	MCA	MCA
ONLYMCA, 1 check	Yes	MCA	MCA	MCA
ONLYMCA, 2 checks	No	–	MCA	MCA
ONLYMCA, 2 checks	Yes	MCA	MCA	MCA + ALT

Key:

**ALT** Alternate user ID.

**CHL** Channel user ID.

**MCA** MCA user ID.

#### Channel initiator example:

A user performs an **MQPUT1** operation to a queue on queue manager QM01 that resolves to a queue called QB on queue manager QM02. The message is sent on a TCP/IP channel called QM01.TO.QM02. RESLEVEL is set to NONE, and the open is performed with alternate user ID and context checking. The receiver channel definition has PUTAUT(CTX) and the MCA user ID is set. Which user IDs are used on the receiving channel to put the message to queue QB?

**Answer:** Table 55 on page 168 shows that two user IDs are checked because RESLEVEL is set to NONE.

Table 61 on page 174 shows that, with PUTAUT set to CTX and 2 checks, the following user IDs are checked:

- The channel initiator user ID and the MCAUSER user ID are checked against the hlq.ALTERNATE.USER.userid profile.
- The channel initiator user ID and the MCAUSER user ID are checked against the hlq.CONTEXT.queueename profile.
- The channel initiator user ID and the alternate user ID specified in the message descriptor (MQMD) are checked against the hlq.Q2 profile.

## User IDs used by the intra-group queuing agent

This section describes the user IDs that are checked when the intra-group queuing agent opens destination queues. The user IDs used are determined by the values of the IGQAUT and IGQUSER queue manager attributes. The possible user IDs are:

### Intra-group queuing user ID (IGQ)

The user ID determined by the IGQUSER attribute of the receiving queue manager. If this is set to blanks, the user ID of the receiving queue manager is used. However, because the receiving queue manager has authority to access all queues defined to it, security checks are not performed for the receiving queue manager's user ID. In this case:

- If only one user ID is to be checked and the user ID is that of the receiving queue manager, no security checks take place. This can occur when IGAUT is set to ONLYIGQ or ALTIGQ.
- If two user IDs are to be checked and one of the user IDs is that of the receiving queue manager, security checks take place for the other user ID only. This can occur when IGAUT is set to DEF, CTX, or ALTIGQ.
- If two user IDs are to be checked and both user IDs are that of the receiving queue manager, no security checks take place. This can occur when IGAUT is set to ONLYIGQ.

### Sending queue manager user ID (SND)

The user ID of the queue manager within the queue-sharing group that put the message on to the SYSTEM.QSG.TRANSMIT.QUEUE.

### Alternate user ID (ALT)

The user ID specified in the *UserIdentifier* field in the message descriptor of the message.

Table 64. User IDs checked against profile name for intra-group queuing

IGQAUT option specified on receiving queue manager	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queueename profile	hlq.resourcename profile
DEF, 1 check	–	SND	SND
DEF, 2 checks	–	SND +IGQ	SND +IGQ
CTX, 1 check	SND	SND	SND
CTX, 2 checks	SND + IGQ	SND +IGQ	SND + ALT
ONLYIGQ, 1 check	–	IGQ	IGQ
ONLYIGQ, 2 checks	–	IGQ	IGQ
ALTIGQ, 1 check	–	IGQ	IGQ
ALTIGQ, 2 checks	IGQ	IGQ	IGQ + ALT

Table 64. User IDs checked against profile name for intra-group queuing (continued)

IGQAUT option specified on receiving queue manager	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queuenam profile	hlq.resourcename profile
Key:			
ALT	Alternate user ID.		
IGQ	IGQ user ID.		
SND	Sending queue manager user ID.		

## Blank user IDs and UACC levels

Blank user IDs can exist when a user is manipulating messages using context or alternate-user security, or when WebSphere MQ is passed a blank user ID. For example, a blank user ID is used when a message is written to the system-command input queue without context.

**Note:** A user ID of "\*" (that is, an asterisk character followed by seven spaces) is treated as a blank user ID.

WebSphere MQ passes the blank user ID to RACF and a RACF undefined user is signed on. All security checks then use the universal access (UACC) for the relevant profile. Depending on how you have set your access levels, the UACC might give the undefined user a wide-ranging access.

For example, if you issue this RACF command from TSO:

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.EVERYONE UACC(UPDATE)
```

you define a profile that enables both z/OS-defined user IDs (that have not been put in the access list) and the RACF undefined user ID to put messages on, and get messages from, that queue.

To protect against blank user IDs you must plan your access levels carefully, and limit the number of people who can use context and alternate-user security. You must prevent people using the RACF undefined user ID from getting access to resources that they should not. However, at the same time, you must allow access to people with defined user IDs. To do this, you can specify a user ID of asterisk (\*) in a RACF command PERMIT. For example, these RACF commands prevent the RACF undefined user ID from gaining access to the queue to put or get messages:

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.RACF.DEFINED.USERS.ONLY UACC(NONE)
PERMIT Q.AVAILABLE.TO.RACF.DEFINED.USERS.ONLY CLASS(MQQUEUE) ACCESS(UPDATE) ID(*)
```

## WebSphere MQ security management

WebSphere MQ uses an in-storage table to hold information relating to each user and the access requests made by each user.

To manage this table efficiently and to reduce the number of requests made from WebSphere MQ to the external security manager (ESM), these controls are available:

- User ID reverification
- User ID timeouts
- Security refreshes
- Displaying security status

These controls are available through both the operations and control panels and WebSphere MQ commands.

The chapter also discusses the following subjects:

- “Security installation tasks” on page 184
- “Auditing considerations” on page 188
- “Customizing security” on page 190
- “Security problem determination” on page 190

## User ID reverification

If the RACF definition of a user who is using WebSphere MQ resources has been changed—for example, by connecting the user to a new group—you can tell the queue manager to sign this user on again the next time it tries to access a WebSphere MQ resource. You can do this by using the WebSphere MQ command RVERIFY SECURITY. For example:

- User HX0804 is getting and putting messages to the PAYROLL queues on queue manager PRD1. However HX0804 now requires access to some of the PENSION queues on the same queue manager (PRD1).
- The data security administrator connects user HX0804 to the RACF group that allows access to the PENSION queues.
- So that HX0804 can access the PENSION queues immediately—that is, without shutting down queue manager PRD1, or waiting for HX0804 to time out—you must use the WebSphere MQ command:

```
RVERIFY SECURITY(HX0804)
```

**Note:** If you turn off user ID timeout for long periods of time (days or even weeks), while the queue manager is running, you must remember to perform an RVERIFY SECURITY for any users that have been revoked or deleted in that time.

## User ID timeouts

When a user accesses a WebSphere MQ resource, the queue manager tries to sign this user on to the queue manager (if subsystem security is active). This means that the user is authenticated to the ESM. This user remains signed on to WebSphere MQ until either the queue manager is shut down, or until the user ID is “timed out” (the authentication lapses) or reverified (reauthenticated).

When a user is timed out, the user ID is “signed off” within the queue manager and any security-related information retained for this user is discarded. The signing on and off of the user within the queue manager is transparent to the application program and to the end user.

Users are eligible for time out when they have not used any WebSphere MQ resources for a predetermined amount of time. This time period is set by the MQSC ALTER SECURITY command. For a description of the command syntax, see WebSphere MQ Script (MQSC) Command Reference.

Two values can be specified in the ALTER SECURITY command:

#### **TIMEOUT**

The time period in minutes that an unused user ID and its associated resources can remain within the WebSphere MQ queue manager.

#### **INTERVAL**

The time period in minutes between checks for user IDs and their associated resources, to determine whether the TIMEOUT has expired.

For example, if the TIMEOUT value is 30 and the INTERVAL value is 10, every 10 minutes WebSphere MQ checks user IDs and their associated resources to determine whether any have not been used for 30 minutes. If a timed-out user ID is found, that user ID is signed off within the queue manager. If any timed-out resource information associated with non-timed out user IDs is found, that resource information is discarded. If you do not want to time-out user IDs, set the INTERVAL value to zero. However, if the INTERVAL value is zero, storage occupied by user IDs and their associated resources is not freed until you issue a REFRESH SECURITY or RVERIFY SECURITY MQSC command.

Tuning this value can be important if you have many one-off users. If you set small interval and timeout values, resources that are no longer required are freed.

**Note:** If you use values for INTERVAL or TIMEOUT other than the defaults, you must reenter the command at every queue manager startup. You can do this automatically by putting the ALTER SECURITY command in the CSQINP1 data set for that queue manager.

## **Refreshing queue manager security**

When a queue is opened for the first time (or for the first time since a security refresh) WebSphere MQ performs a RACF check to obtain the user's access rights and places this information in the cache. The cached data includes user IDs and resources on which security checking has been performed. If the queue is opened again by the same user the presence of the cached data means WebSphere MQ does not have to issue RACF checks, which improves performance. The action of a security refresh is to discard any cached security information and so force WebSphere MQ to make a new check against RACF. Whenever you add, change or delete a RACF resource profile that is held in the MQADMIN, MXADMIN, MQPROC, MXPROC, MQQUEUE, MXQUEUE, MQNLIST, MXNLIST, or MXTOPIC class, you must tell the queue managers that use this class to refresh the security information that they hold. To do this, issue the following commands:

- The RACF SETROPTS RACLIST(classname) REFRESH command to refresh at the RACF level.
- The WebSphere MQ REFRESH SECURITY command to refresh the security information held by the queue manager (described in the WebSphere MQ Script (MQSC) Command Reference manual). This command needs to be issued by each queue manager that accesses the profiles that have changed. If you have a queue-sharing group, you can use the command scope attribute to direct the command to all the queue managers in the group.

If you are using generic profiles in any of the WebSphere MQ classes, you must also issue normal RACF refresh commands if you change, add, or delete any generic profiles. For example, SETROPTS GENERIC(classname) REFRESH.

However, because WebSphere MQ utilizes the RACF dataspace, WebSphere MQ can use RACF profiles as soon as they become available. If a RACF resource profile is added, changed or deleted and the resource to which it applies has not yet been accessed (so no information is cached), WebSphere MQ will use the new RACF information without a security refresh being carried out.

If RACF auditing is turned on, (for example, by using the RACF RALTER AUDIT(access-attempt (audit\_access\_level)) command), no caching takes place, and therefore WebSphere MQ refers directly to the RACF dataspace for every check. Changes are therefore picked up immediately and REFRESH SECURITY is not necessary to access the changes. You can confirm whether RACF auditing is on by using the RACF RLIST command. For example, you could issue the command

```
RLIST MQQUEUE (qmgr.SYSTEM.COMMAND.INPUT) GEN
```

and receive the results

```
CLASS      NAME
-----
MQQUEUE    QP*.SYSTEM.COMMAND.*.** (G)
           AUDITING
           -----
           FAILURES(READ)
```

This would indicate that auditing is set on. For more information, refer to the *z/OS Security Server RACF Auditor's Guide* and the *z/OS Security Server RACF Command Language Reference*.

Figure 29 on page 182 summarizes the situations in which security information is cached and in which cached information is used.

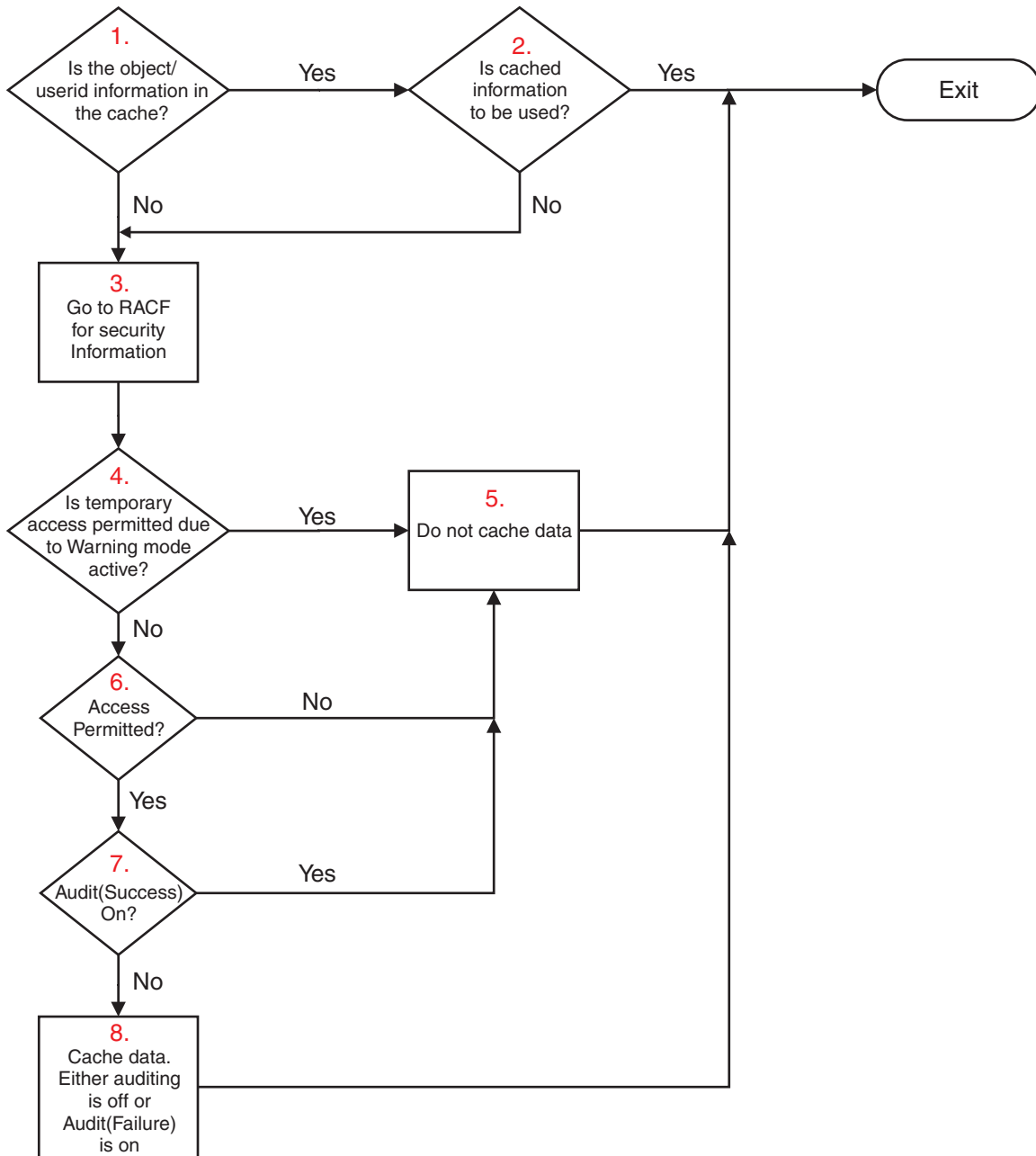


Figure 29. Logic flow for WebSphere MQ security caching

If you change your security settings by adding or deleting switch profiles in the MQADMIN or MXADMIN classes, use one of these commands to pick up these changes dynamically:

```

REFRESH SECURITY(*)
REFRESH SECURITY(MQADMIN)
REFRESH SECURITY(MXADMIN)
  
```

This means you can activate new security types, or deactivate them without having to restart the queue manager.



For performance reasons, these are the only classes affected by the REFRESH SECURITY command. You do *not* need to use REFRESH SECURITY if you change a profile in either the MQCONN or MQCMD classes.

**Note:** A refresh of the MQADMIN or MXADMIN class is not required if you change a RESLEVEL security profile.

For performance reasons, use REFRESH SECURITY as infrequently as possible, ideally at off-peak times. You can minimize the number of security refreshes by connecting users to RACF groups that are already in the access list for WebSphere MQ profiles, rather than putting individual users in the access lists. In this way, you change the user rather than the resource profile. You can also RVERIFY SECURITY the appropriate user instead of refreshing security.

As an example of REFRESH SECURITY, suppose you define the new profiles to protect access to queues starting with INSURANCE.LIFE on queue manager PRMQ. You use these RACF commands:

```
RDEFINE MQQUEUE PRMQ.INSURANCE.LIFE.** UACC(NONE)
PERMIT PRMQ.INSURANCE.LIFE.** ID(LIFEGRP) ACCESS(UPDATE)
```

You must issue the following command to tell RACF to refresh the security information that it holds, for example:

```
SETROPTS RACLIST(MQQUEUE) REFRESH
```

Because these profiles are generic, you must tell RACF to refresh the generic profiles for MQQUEUE. For example:

```
SETROPTS GENERIC(MQQUEUE) REFRESH
```

Then you must use this command to tell queue manager PRMQ that the queue profiles have changed:

```
REFRESH SECURITY(MQQUEUE)
```

## Refreshing SSL security

To refresh the cached view of the SSL Key Repository, issue the REFRESH SECURITY command with the option TYPE(SSL). This enables you to update some of your SSL settings without having to restart your channel initiator. For more information about REFRESH SECURITY TYPE(SSL) see WebSphere MQ Security. For details of the command see WebSphere MQ Script (MQSC) Command Reference.

## Displaying security status

To display the status of the security switches, and other security controls, you can issue the MQSC DISPLAY SECURITY command. For a description of the command syntax, see WebSphere MQ Script (MQSC) Command Reference.

Figure 30 shows a typical output of the DISPLAY SECURITY ALL command.

```
CSQH015I +CSQ1 Security timeout = 54 MINUTES
CSQH016I +CSQ1 Security interval = 12 MINUTES
CSQH030I +CSQ1 Security switches ...
CSQH034I +CSQ1 SUBSYSTEM: ON, 'SQ05.NO.SUBSYS.SECURITY' not found
CSQH032I +CSQ1 QMGR: ON, 'CSQ1.YES.QMGR.CHECKS' found
CSQH031I +CSQ1 QSG: OFF, 'SQ05.NO.QSG.CHECKS' found
CSQH031I +CSQ1 CONNECTION: OFF, 'CSQ1.NO.CONNECT.CHECKS' found
CSQH034I +CSQ1 COMMAND: ON, 'CSQ1.NO.COMMAND.CHECKS' not found
CSQH031I +CSQ1 CONTEXT: OFF, 'CSQ1.NO.CONTEXT.CHECKS' found
CSQH034I +CSQ1 ALTERNATE USER: ON, 'CSQ1.NO.ALTERNATE.USER.CHECKS' not found
CSQH034I +CSQ1 PROCESS: ON, 'CSQ1.NO.PROCESS.CHECKS' not found
CSQH034I +CSQ1 NAMLIST: ON, 'CSQ1.NO.NLIST.CHECKS' not found
CSQH034I +CSQ1 QUEUE: ON, 'CSQ1.NO.QUEUE.CHECKS' not found
CSQH034I +CSQ1 TOPIC: ON, 'CSQ1.NO.TOPIC.CHECKS' not found
CSQH031I +CSQ1 COMMAND RESOURCES: OFF, 'CSQ1.NO.CMD.RESC.CHECKS' found
CSQ9022I +CSQ1 CSQHPDTC ' DISPLAY SECURITY' NORMAL COMPLETION
```

Figure 30. Typical output from the DISPLAY SECURITY command

The example shows that the queue manager that replied to the command has subsystem, command, alternate user, process, namelist, and queue security active at queue manager level but not at queue-sharing group level. Connection, command resource, and context security are not active. It also shows that user ID timeouts are active, and that every 12 minutes the queue manager checks for user IDs that have not been used in this queue manager for 54 minutes and removes them.

**Note:** This command shows the current security status. It does not necessarily reflect the current status of the switch profiles defined to RACF, or the status of the RACF classes. For example, the switch profiles might have been changed since the last restart of this queue manager or REFRESH SECURITY command.

## Security installation tasks

When WebSphere MQ is first installed and customized, you must perform these security-related tasks:

1. Set up WebSphere MQ data set and system security by:
  - Authorizing the queue manager started-task procedure xxxxMSTR and the distributed queuing started-task procedure xxxxCHIN to run under RACF.
  - Authorizing access to queue manager data sets.
  - Authorizing access to resources for those user IDs that will use the queue manager and utility programs.
  - Authorizing access for those queue managers that will use the coupling facility list structures.
  - Authorizing access for those queue managers that will use DB2.
2. Set up RACF definitions for WebSphere MQ security.
3. If you want to use the Secure Sockets Layer (SSL), prepare your system to use certificates and keys (see “Configuring your system to use the Secure Sockets Layer (SSL)” on page 186).

## Setting up WebSphere MQ data set security

The possible users of WebSphere MQ data sets include:

- The queue manager itself.
- The channel initiator
- WebSphere MQ administrators who need to create WebSphere MQ data sets, run utility programs, and so on.
- Application programmers, who need to use the WebSphere MQ-supplied copybooks, include data sets, macros, and so on.
- Applications involving one or more of the following:
  - Batch jobs
  - TSO users
  - CICS regions
  - IMS regions
- Data sets CSQOUTX and CSQSNAP
- Dynamic queues SYSTEM.CSQXCMD.\*

For all these potential users, protect the WebSphere MQ data sets with RACF.

You must also control access to all your 'CSQINP' data sets.

### **RACF authorization of started-task procedures:**

Some WebSphere MQ data sets should be for the exclusive use of the queue manager. If you protect your WebSphere MQ data sets using RACF, you must also authorize the queue manager started-task procedure xxxxMSTR, and the distributed queuing started-task procedure xxxxCHIN, using RACF. To do this, use the STARTED class. Alternatively, you can use the started procedures table (ICHRIN03), but then you need to IPL your z/OS system before the changes take effect.

For more information, see the *z/OS Security Server RACF System Programmer's Guide*.

The RACF user ID identified must have the required access to the data sets in the started-task procedure. For example, if you associate a queue manager started task procedure called CSQ1MSTR with the RACF user ID QMGRCSQ1, the user ID QMGRCSQ1 must have access to the z/OS resources accessed by the CSQ1 queue manager.

The RACF user IDs associated with the queue manager and channel initiator started task procedures should not have the TRUSTED attribute set.

### **Authorizing access to data sets:**

The WebSphere MQ data sets should be protected so that no unauthorized user can run a queue manager instance, or gain access to any queue manager data. To do this, use normal z/OS RACF data set protection. For more information, see the *z/OS Security Server RACF Security Administrator's Guide*.

Table 65 on page 186 summarizes the RACF access that the queue manager started task procedure must have to the different data sets.

Table 65. RACF access to data sets associated with a queue manager

RACF access	Data sets
READ	<ul style="list-style-type: none"> <li>• thlqual.SCSQAUTH and thlqual.SCSQANLx (where x is the language letter for your national language).</li> <li>• The data sets referred to by CSQINP1, CSQINP2 and CSQXLIB in the queue manager's started task procedure.</li> </ul>
UPDATE	<ul style="list-style-type: none"> <li>• All page sets and log and BSDS data sets.</li> </ul>
ALTER	<ul style="list-style-type: none"> <li>• All archive data sets.</li> </ul>

Table 66 summarizes the RACF access that the started task procedure for distributed queuing must have to the different data sets.

Table 66. RACF access to data sets associated with distributed queuing

RACF access	Data sets
READ	<ul style="list-style-type: none"> <li>• thlqual.SCSQAUTH, thlqual.SCSQANLx (where x is the language letter for your national language), and thlqual.SCSQMVR1.</li> <li>• LE library data sets.</li> <li>• The data sets referred to by CSQXLIB and CSQINPX in the distributed queuing started task procedure.</li> </ul>
UPDATE	<ul style="list-style-type: none"> <li>• Data sets CSQOUTX and CSQSNAP</li> <li>• Dynamic queues SYSTEM.CSQXCMD.*</li> </ul>

## Setting up WebSphere MQ resource security

The possible users of WebSphere MQ resources, such as queues and channels, are the same as the possible users of WebSphere MQ data sets, as listed above. For all these potential users, protect the WebSphere MQ resources with RACF. In particular, note that the channel initiator needs access to various resources, as described in "Security considerations for distributed queuing" on page 192, and so the user ID under which it runs must be authorized to access these resources.

If you are using a queue-sharing group, the queue manager might issue various commands internally, so the user ID it uses must be authorized to issue such commands. The commands are:

- DEFINE, ALTER, and DELETE for every object that has QSGDISP(GROUP)
- START and STOP CHANNEL for every channel used with CHLDISP(SHARED)

## Configuring your system to use the Secure Sockets Layer (SSL)

If you want to use the Secure Sockets Layer (SSL) for channel security, there are a number of tasks you need to perform to set this up on your system. (See the WebSphere MQ Security book for more information about SSL.)

1. Create a key ring in RACF to hold all the keys and certificates for your system, using the RACF RACDCERT command. For example:

```
RACDCERT ID(QM1) ADDRING(QM1RING)
```

The ID should be either the channel initiator address space user ID or the user ID you wish to own the keyring if it is to be a shared keyring.

2. Create a digital certificate for each queue manager, using the RACF RACDCERT command.

The label of the certificate must be of the form `ibmWebSphereMQqmgr-name`, so in this example it is `ibmWebSphereMQQM1`.

For example:

```
RACDCERT ID(QM1) GENCERT
SUBJECTSDN(CN('username') O('IBM') OU('departmentname') C('England'))
WITHLABEL('ibmWebSphereMQQM1')
```

3. Connect the certificate in RACF to the key ring, using the RACF RACDCERT command. For example:

```
RACDCERT CONNECT(ID(QM1) LABEL('ibmWebSphereMQQM1') RING(QM1RING))
```

You also need to connect any relevant signer certificates (from a Certification Authority) to the key ring. That is, all Certification Authorities for this queue manager's SSL certificate and all Certification Authorities for all SSL certificates that this queue manager communicates with. For example:

```
RACDCERT ID(userid)
CONNECT(CERTAUTH LABEL('My CA') RING(ring-name) USAGE(CERTAUTH))
```

4. On each of your queue managers, use the WebSphere MQ ALTER QMGR command to specify the key repository that the queue manager needs to point to. For example, if the keyring is owned by the channel initiator address space:

```
ALTER QMGR SSLKEYR(QM1RING)
```

or if you are using a shared keyring:

```
ALTER QMGR SSLKEYR(userid/QM1RING)
```

where *userid* is the user ID that owns the shared keyring.

5. Certificate Revocation Lists (CRLs) allow the Certification Authorities to revoke certificates that can no longer be trusted. CRLs are stored in LDAP servers. To access this list on the LDAP server, you first need to create an AUTHINFO object of AUTHTYPE CRLLDAP, using the WebSphere MQ DEFINE AUTHINFO command. For example:

```
DEFINE AUTHINFO(LDAP1)
AUTHTYPE(CRLLDAP)
CONNAME(ldap.server(389))
LDAPUSER('')
LDAPPWD('')
```

In this example, the certificate revocation list is stored in a public area of the LDAP server, so the LDAPUSER and LDAPPWD fields are not necessary. Next, put your AUTHINFO object into a namelist, using the WebSphere MQ DEFINE NAMELIST command. For example:

```
DEFINE NAMELIST(LDAPNL) NAMES(LDAP1)
```

Finally, associate the namelist with each queue manager, using the WebSphere MQ ALTER QMGR command. For example:

```
ALTER QMGR SSLCRLNL(LDAPNL)
```

6. Set up your queue manager to run SSL calls, using the WebSphere MQ ALTER QMGR command. This defines server subtasks that handle SSL calls only, which leaves the normal dispatchers to continue processing as normal without being impacted by any SSL calls. You must have at least two of these subtasks. For example:

```
ALTER QMGR SSLTASKS(8)
```

This change only takes effect when the channel initiator is restarted.

7. Specify the cipher specification to be used for each channel, using the WebSphere MQ DEFINE CHANNEL or ALTER CHANNEL command. For example:

```
ALTER CHANNEL(LDAPCHL)  
CHLTYPE(SDR)  
SSLCIPH(RC4_MD5_US)
```

Both ends of the channel must specify the same cipher specification.

## Auditing considerations

The normal RACF auditing controls are available for conducting a security audit of a queue manager. The RACF auditing can be based upon:

- User IDs
- Resource classes
- Profiles

For more details, see the *z/OS Security Server RACF Auditor's Guide*.

**Note:** Auditing degrades performance; the more auditing you implement, the more performance is degraded. This is also a consideration for the use of the RACF WARNING option.

## Auditing RESLEVEL

You can decide whether to produce RESLEVEL audit records by setting the RESAUDIT system parameter to YES or NO. If the RESAUDIT parameter is set to NO, audit records are not produced. For more details about setting this parameter, see "Using CSQ6SYSP" on page 28.

If RESAUDIT is set to YES, no normal RACF audit records are taken when the RESLEVEL check is made to see what access an address space user ID has to the hlq.RESLEVEL profile. Instead, WebSphere MQ requests that RACF create a GENERAL audit record (event number 27). These checks are only carried out at connect time, so the overhead should be minimal.

You can report the WebSphere MQ general audit records using the RACF report writer (RACFRW). You could use the following RACFRW commands to report the RESLEVEL access:

```
RACFRW
SELECT PROCESS
EVENT GENERAL
LIST
END
```

A sample report from RACFRW, excluding the *Date*, *Time*, and *SYSID* fields, is shown in Figure 31.

```

RACF REPORT - LISTING OF PROCESS RECORDS                                PAGE 4
      E
      V Q
      E U
*JOB/USER *STEP/  --TERMINAL--  N A
  NAME    GROUP   ID    LVL  T  L
WS21B    MQMGRP  IGJZM000  0   27 0  JOBID=(WS21B 05.111 09:44:57),USERDATA=()
  TRUSTED USER                                AUTH=(NONE),REASON=(NONE)
                                              SESSION=TSOLOGON,TERMINAL=IGJZM000,
                                              LOGSTR='CSQH RESLEVEL CHECK PERFORMED AGAINST PROFILE(QM66.RESLEVEL),
                                              CLASS(MQADMIN), ACCESS EQUATES TO (CONTROL)',RESULT=SUCCESS,MQADMIN
```

Figure 31. Sample output from RACFRW showing RESLEVEL general audit records

From checking the LOGSTR data in the output above, you can see that TSO user WS21B has CONTROL access to QM66.RESLEVEL. This means that all resource security checks are bypassed when user WS21B access QM66 resources.

For more information about using RACFRW, see the *z/OS Security Server RACF Auditor's Guide*.

## Statistics

WebSphere MQ does not gather any security statistics of its own. The only statistics are those that can be created by auditing.

## Customizing security

If you want to change the way WebSphere MQ security operates, you must do this through the SAF exit (ICHRFR00), or exits in your external security manager. To find out more about RACF exits, see the *z/OS Security Server RACROUTE Macro Reference* manual.

**Note:** Because WebSphere MQ optimizes calls to the ESM, RACROUTE requests might not be made on, for example, every open for a particular queue by a particular user.

## Security problem determination

This section describes the conditions under which violation messages can be generated in a WebSphere MQ application program and provides a checklist to be implemented if the ESM is not controlling access in the way that you expect.

### Violation messages

A return code of MQRC\_NOT\_AUTHORIZED can be returned to an application program because:

- A user is not allowed to connect to the queue manager. In this case, you get an ICH408I message in the Batch/TSO, CICS, or IMS job log.
- A user sign-on to the queue manager has failed because, for example, the job user ID is not valid or appropriate, or the task user ID or alternate user ID is not valid. One or more of these user IDs might not be valid because they have been revoked or deleted. In this case, you get an ICHxxxx message and possibly an IRRxxxx message in the queue manager job log giving the reason for the sign-on failure. For example:

```
ICH408I USER(NOTDFND ) GROUP(      ) NAME(???)
LOGON/JOB INITIATION - USER AT TERMINAL      NOT RACF-DEFINED
IRR012I VERIFICATION FAILED. USER PROFILE NOT FOUND
```

- An alternate user has been requested, but the job or task user ID does not have access to the alternate user ID. For this failure, you get a violation message in the job log of the relevant queue manager.
- A context option has been used or is implied by opening a transmission queue for output, but the job user ID or, where applicable, the task or alternate user ID does not have access to the context option. In this case, a violation message is put in the job log of the relevant queue manager.
- An unauthorized user has attempted to access a secured queue manager object, for example, a queue. In this case, an ICH408I message for the violation is put in the job log of the relevant queue manager. This violation might be due to the job or, when applicable, the task or alternate user ID.

Violation messages for command security and command resource security can also be found in the job log of the queue manager.

If the ICH408I violation message shows the queue manager jobname rather than a user ID, this is normally the result of a blank alternate user ID being specified. For example:



```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
      MQS1.PAYROLL.REQUEST CL(MQQUEUE)
      INSUFFICIENT ACCESS AUTHORITY
      ACCESS INTENT(UPDATE ) ACCESS ALLOWED(NONE )
```

You can find out who is allowed to use blank alternate user IDs by checking the access list of the MQADMIN profile hlq.ALTERNATE.USER.-BLANK-.

An ICH408I violation message can also be generated by:

- A command being sent to the system-command input queue without context. User-written programs that write to the system-command input queue should always use a context option. For more information, see “Profiles for context security” on page 149.
- When the job accessing the WebSphere MQ resource does not have a user ID associated with it, or when a WebSphere MQ adapter cannot extract the user ID from the adapter environment.

Violation messages might also be issued if you are using both queue-sharing group and queue manager level security. You might get messages indicating that no profile has been found at queue manager level, but still be granted access because of a queue-sharing group level profile.

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
      MQS1.PAYROLL.REQUEST CL(MQQUEUE)
      PROFILE NOT FOUND - REQUIRED FOR AUTHORITY CHECKING
      ACCESS INTENT(UPDATE ) ACCESS ALLOWED(NONE )
```

## What to do if access is allowed or disallowed incorrectly

In addition to the steps detailed in the *z/OS Security Server RACF Security Administrator's Guide*, use this checklist if access to a resource appears incorrectly controlled:

- Are the switch profiles correctly set?
  - Is RACF active?
  - Are the WebSphere MQ RACF classes installed and active?  
Use the RACF command, SETROPTS LIST, to check this.
  - Use the WebSphere MQ DISPLAY SECURITY command to display the current switch status from the queue manager.
  - Check the switch profiles in the MQADMIN class.  
Use the RACF commands, SEARCH and RLIST, for this.
  - Recheck the RACF switch profiles by issuing the WebSphere MQ REFRESH SECURITY(MQADMIN) command.
- Has the RACF resource profile changed? For example, has universal access on the profile changed or has the access list of the profile changed?
  - Is the profile generic?  
If it is, issue the RACF command, SETROPTS GENERIC(classname) REFRESH.
  - Have you refreshed the security on this queue manager?  
If required, issue the RACF command SETROPTS RACLIST(classname) REFRESH.

- If required, issue the WebSphere MQ REFRESH SECURITY(\*) command.
- Has the RACF definition of the user changed? For example, has the user been connected to a new group or has the user access authority been revoked?
    - Have you reverified the user by issuing the WebSphere MQ RVERIFY SECURITY(userid) command?
  - Are security checks being bypassed due to RESLEVEL?
    - Check the connecting user ID's access to the RESLEVEL profile. Use the RACF audit records to determine what the RESLEVEL is set to.
    - If you are running from CICS, check the transaction's RESSEC setting.
    - If RESLEVEL has been changed while a user is connected, they must disconnect and reconnect before the new RESLEVEL setting takes effect.
  - Are you using queue-sharing groups?
    - If you are using both queue-sharing group and queue manager level security, check that you have defined all the correct profiles. If queue manager profile is not defined, a message is sent to the log stating that the profile was not found.
    - Have you used a combination of switch settings that is not valid so that full security checking has been set on?
    - Do you need to define security switches to override some of the queue-sharing group settings for your queue manager?
    - Is a queue manager level profile taking precedence over a queue-sharing group level profile?

---

## Security considerations for distributed queuing

This chapter discusses security considerations for distributed queuing.

This chapter also discusses security considerations for using clusters.

### The channel initiator

Security considerations for distributed queuing.

If you are using resource security, you should consider the following if you are using distributed queuing:

#### System queues

The channel initiator address space needs RACF UPDATE access to the system queues listed at "System queue security" on page 142, and to all the user destination queues and the dead-letter queue (but see "Dead-letter queue security" on page 141).

#### Transmission queues

The channel initiator address space needs ALTER access to all the user transmission queues.

#### Context security

The channel user ID (and the MCA user ID if one has been specified) also need RACF CONTROL access to the hlq.CONTEXT.queuname profiles in the MQADMIN class. Depending on the RESLEVEL profile, the network-received user ID might also need CONTROL access to these profiles. See "Profiles for context security" on page 149 "RESLEVEL and channel initiator connections" on page 167 and "User IDs for security checking" on page 170 for more information.

## CSQINPX

If you are using the CSQINPX input data set, the channel initiator also needs READ access to CSQINPX, and UPDATE access to data set CSQOUTX and dynamic queues SYSTEM.CSQXCMD.\*.

## Connection security

The channel initiator address space connection requests use a connection type of CHIN, for which appropriate access security must be set, see “Connection security profiles for the channel initiator” on page 135.

## Data sets

The channel initiator address space needs appropriate access to queue manager data sets, see “Authorizing access to data sets” on page 185.

## Commands

The distributed queuing commands (for example, DEFINE CHANNEL, START CHINIT, START LISTENER, and so on) should have appropriate command security set, see Table 47 on page 152.

If you are using a queue-sharing group, the channel initiator might issue various commands internally, so the user ID it uses must be authorized to issue such commands. These commands are START and STOP CHANNEL for every channel used with CHLDISP(SHARED).

## Channel security

Channels, particularly receivers and server-connections, need appropriate security to be set up; see “User IDs for security checking” on page 170 for more information.

You can also use the Secure Sockets Layer (SSL) protocol to provide security on channels. See the WebSphere MQ Security book for a detailed description of SSL.

See also the WebSphere MQ Clients manual for information about server-connection security.

## User IDs

The user IDs described in “User IDs used by the channel initiator” on page 173 and “User IDs used by the intra-group queuing agent” on page 177 need the following:

- RACF UPDATE access to the appropriate destination queues and the dead-letter queue
- RACF CONTROL access to the hlq.CONTEXT.queuename profile if context checking is performed at the receiver
- Appropriate access to the hlq.ALTERNATE.USER.userid profiles they might need to use.
- For clients, the appropriate RACF access to the resources to be used.

## APPC security

Set appropriate APPC security if you are using the LU 6.2 transmission protocol. (Use the APPCLU RACF class for example.) For information about setting up security for APPC, see the following manuals:

- *z/OS V1R2.0 MVS Planning: APPC Management*
- *Multiplatform APPC Configuration Guide*, an IBM Redbooks publication

Outbound transmissions use the “SECURITY(SAME)” APPC option. This means that the user ID of the channel initiator address space and its

default profile (RACF GROUP) are flowed across the network to the receiver with an indicator that the user ID has already been verified (ALREADYV).

If the receiving side is also z/OS, the user ID and profile are verified by APPC and the user ID is presented to the receiver channel and used as the channel user ID.

In an environment where the queue manager is using APPC to communicate with another queue manager on the same or another z/OS system, you need to ensure that either:

- The VTAM® definition for the communicating LU specifies SETACPT(ALREADYV)
- There is a RACF APPCLU profile for the connection between LUs that specifies CONVSEC(ALREADYV)

### Changing security settings

If the RACF access level that either the channel user ID or MCA user ID has to a destination queue is changed, this change will only take effect for new object handles (that is, new **MQOPENS**) for the destination queue. The times when MCAs open and close queues is variable; if a channel is already running when such an access change is made, the MCA can continue to put messages on the destination queue using the existing security access of the user ID(s) rather than the updated security access. To avoid this, you should stop and restart the channels to enforce the updated access level.

### Automatic restart

If you are using the z/OS Automatic Restart Manager (ARM) to restart the channel initiator, the user ID associated with the XCFAS address space must be authorized to issue the WebSphere MQ START CHINIT command.

## Cluster support

This section discusses the security considerations for cluster support.

You can use the MCA user ID and security exits to authenticate cluster channels (as with conventional channels). The security exit on the cluster-receiver channel must check that the queue manager is permitted access to the server queue manager's clusters. You can start to use WebSphere MQ cluster support without having to change your existing queue access security, however you must allow other queue managers in the cluster to write to the SYSTEM.CLUSTER.COMMAND.QUEUE if they are to join the cluster.

WebSphere MQ cluster support does not provide a mechanism to limit a member of a cluster to the client role only. As a result, you must be sure that you trust any queue managers that you allow into the cluster. If any queue manager in the cluster creates a queue with a particular name, it can receive messages for that queue, regardless of whether the application putting messages to that queue intended this or not.

To restrict the membership of a cluster, you need to take the same action that you would take to prevent queue managers connecting to receiver channels. You can achieve this by writing a security exit program on the receiver channel or by writing an exit program to prevent unauthorized queue managers from writing to the SYSTEM.CLUSTER.COMMAND.QUEUE.

**Note:** It is not advisable to permit applications to open the SYSTEM.CLUSTER.TRANSMIT.QUEUE directly, just as it is not advisable to permit an application to open any other transmission queue directly.

If you are using resource security you should consider the following in addition to the considerations discussed in “Security considerations for distributed queuing” on page 192:

#### **System queues**

The channel initiator needs RACF ALTER access to the following system queues:

- SYSTEM.CLUSTER.COMMAND QUEUE
- SYSTEM.CLUSTER.TRANSMIT.QUEUE.

and UPDATE access to SYSTEM.CLUSTER.REPOSITORY.QUEUE

It also needs READ access to any namelists used for clustering.

#### **Commands**

The cluster support commands (REFRESH and RESET CLUSTER, SUSPEND and RESUME QMGR) should have appropriate command security set (as described in Table 47 on page 152).

---

## **Security considerations for using WebSphere MQ with CICS**

The CICS adapter provides the following information to WebSphere MQ specifically for use in WebSphere MQ security:

- Whether CICS resource-level security is active for this transaction—as specified on the RESSEC or RSLC operand of the RDO TRANSACTION definition.
- User IDs.

For terminal tasks where a user has not signed on, the user ID is the CICS user ID associated with the terminal and is either:

- The default CICS user ID as specified on the CICS parameter DFLTUSER SIT
- A preset security user ID specified on the terminal definition

For non-terminal tasks, the CICS adapter tries to get a user ID with an EXEC CICS ASSIGN command. If this is unsuccessful, the adapter tries to get the user ID using EXEC CICS INQUIRE TASK. If security is active in CICS, and the non-terminal attached transaction is defined with CMDSEC(YES), the CICS adapter passes a user ID of blanks to WebSphere MQ.

For more information about RACF security management in the CICS environment, see the *CICS Transaction Server for OS/390 V1.3 CICS RACF Security Guide*.

## **Controlling the security of CICS transactions supplied by WebSphere MQ**

The CKTI and CKAM transactions are designed to be run without a terminal; no user should have access to these transactions. These transactions are examples of what the *CICS RACF Security Guide* calls “category 1 transactions”. For information about how to set these transactions up in CICS and RACF, see the information about category 1 transactions in the *CICS RACF Security Guide*.

If you want a user to administer the CICS adapter, you must grant the user authorization to these transactions:

CKQC	Controls the CICS adapter functions
CKBM	Controls the CICS adapter functions
CKRT	Controls the CICS adapter functions
CKCN	Connect
CKSD	Disconnect
CKRS	Statistics
CKDP	Full screen display
CKDL	Line mode display
CKSQ	CKTI START/STOP

If required, you can restrict access to specific functions of the adapter. For example, if you want to allow users to display the current status of the adapter through the full screen interface, but nothing else, give them access to CKQC, CKBM, CKRT, and CKDP only.

You should define these transactions to CICS with RESSEC(NO) and CMDSEC(NO). For more details, see the *CICS RACF Security Guide*. For information about the security of the CICS transactions supplied by WebSphere MQ for remote queuing, see the WebSphere MQ Intercommunication manual.

## CICS adapter user IDs

The user ID associated with the CICS adapter is that of the WebSphere MQ-supplied task initiator transaction, CKTI. This section describes some of the implications of this.

### User ID checking for WebSphere MQ resources during PLTPI and PLTSD

If a WebSphere MQ resource is accessed during the CICS PLTPI phase, the user ID passed to WebSphere MQ is blanks. If a WebSphere MQ resource is accessed during the CICS PLTSD phase, the user ID passed to WebSphere MQ is the user ID associated with the shutdown transaction.

If CKTI is started during the CICS PLTPI phase, the user ID of the CKTI task is the CICS sysidnt. This means that a user ID with the same name as the CICS sysidnt must be defined and given access to the required WebSphere MQ resources, for example, initiation queues.

### Terminal user IDs

If CKTI is started from a terminal from the CKQC transaction or a user-written program that links to CSQCSSQ, the user ID that CKTI uses is the same as the user ID of the terminal that started CKTI.

### Automating starting of CKTI

To automate the starting of CKTIs under a specific user ID, you can use an automation product, for example, NetView®. You can use this to sign on a CICS console and issue the STARTCKTI command.

You can also use preset security sequential terminals, which have been defined to emulate a CRLP terminal, with the sequential terminal input containing the CKQC STARTCKTI command.

However, when the CICS adapter alert monitor reconnects CICS to WebSphere MQ, after, for example, a WebSphere MQ restart, only the CKTI specified at the initial WebSphere MQ connection is restarted. You must automate starting any extra CKTIs yourself.

### **Propagating the CKTI user ID to other CICS transactions**

If CKTI starts other CICS transactions, for example, message channel agents (MCAs) or user-written CICS applications, the user ID of CKTI is propagated to these applications. For example, if CKTI is running under user ID CIC1 and a trigger event occurs that requires the sender MCA transaction, CKSG, to be started, the CKSG transaction also runs under user ID CIC1. Therefore user ID CIC1 must have access to the required transmission queue.

## **Security considerations for the CICS bridge**

When you run the CICS bridge, you can specify the level of authentication you want to take place. If requested, the bridge checks the user ID and password extracted from the WebSphere MQ request message before running the CICS program named in the request message. The queue manager uses the external security manager (ESM) (for example RACF) to do authentication. Therefore user IDs in the request message have to be defined to the ESM.

#### **Note:**

1. If you have not specified a user ID in the message descriptor (MQMD) or password in the CICS bridge header (MQCIH) of a message, the bridge task runs with the LOCAL level of authentication, even if you started the bridge monitor with a different authentication option.
2. The options that include password (or passticket) validation require an MQCIH to be provided. See the WebSphere MQ Application Programming Reference manual for more information about the MQCIH header.
3. Passticket validation is performed using WebSphere MQ services, not EXEC CICS VERIFY, as the CICS service does not allow you to specify an APPLID.

The level of authentication you can use is described below:

#### **LOCAL**

This is the default. CICS programs run by the bridge task are started with the CICS DFLTUSER user ID, therefore run with the authority associated with this user ID. There is no checking of user IDs or passwords. If a CICS program is run that tries to access protected resources, it will probably fail.

#### **IDENTIFY**

When you start the monitor task with the IDENTIFY authentication option, the bridge task is started with the user ID specified in the message (MQMD). CICS programs run by the bridge run with the user ID from the MQMD. There is no password checking, the user ID is treated as trusted.

#### **VERIFY\_UOW**

When you start the monitor task with the VERIFY\_UOW authentication option, the bridge monitor checks the user ID (in the MQMD) and password (in the CIH) before starting the bridge task. CICS programs run by the bridge run with the user ID extracted from the MQMD. If the user

ID or password is invalid, the request fails with return code MQCRC\_SECURITY\_ERROR. Subsequent messages processed by this transaction are not checked.

#### VERIFY\_ALL

This is the same as VERIFY\_UOW except that the bridge task checks the user ID and password in **every** message. This is not applicable for 3270 transactions when using CICS earlier than CICS Transaction Server Version 2 Release 2.

A passticket can be used in place of a password to avoid the need to flow passwords in messages (see Security Server RACF System Programmer's Guide). When generating a passticket an APPLID must be specified. If you are using a single bridge monitor, the APPLID is the CICS APPLID unless a different value was specified when the bridge was started. If you are using multiple bridge monitors for a queue, you must specify the APPLID to be used via the PASSTKTA=*applid* parameter at bridge startup.

If you have not specified a user ID in a message, or you have not provided a password, the CICS program started by the CICS bridge runs with the user ID set to the user ID used to start the bridge monitor, regardless of the option requested. If you want more than one level of authentication checking performed, run a monitor task for each level you need.

When a CICS DPL request is read by the bridge monitor it starts the transaction specified in the CICS bridge header (MQCIH) or, if this is blank, transaction CKBP. The user IDs under which the bridge monitor runs must have authority to start the various transactions that might be requested. The default transaction ID for the CICS bridge monitor is CKBR but you can change this or define additional transaction IDs if you want more granular access to queues and transactions. You can use CICS surrogate security to restrict which user ID and transaction combinations a bridge monitor transaction and user ID can start.

Table 67 and Table 68 summarize the level of authority of the bridge monitor and the bridge tasks, and the use of the MQMD user ID.

Table 67. CICS bridge monitor security

Monitor started by	At a signed on terminal	Monitor authority
From a terminal or EXEC CICS LINK within a program	Yes	Signed on user ID
From a terminal or EXEC CICS LINK within a program	No	CICS default user ID
EXEC CICS START with user ID	–	User ID from START
EXEC CICS START without user ID	–	CICS default user ID
The WebSphere MQ trigger monitor CKTI	–	CICS default user ID

Table 68. CICS bridge task security

AUTH	Bridge task authority
LOCAL	CICS default user ID
IDENTIFY	MQMD UserIdentifier
VERIFY_UOW	MQMD UserIdentifier
VERIFY_ALL	MQMD UserIdentifier



The options IDENTIFY, VERIFY\_UOW, and VERIFY\_ALL need the user ID of the bridge monitor defined to RACF as a surrogate of all the user IDs used in request messages. This is in addition to the user ID in the message being defined to RACF. (A surrogate user is one who has the authority to start work on behalf of another user, without knowing the other user's password.)

For more information on surrogate user security, see the *CICS RACF Security Guide*.

**Note:** When IDENTIFY security is being used, you might see abend AICO for CKBP if you try to run with a user ID that has been revoked. The error reply will have return code MQCRC\_BRIDGE\_ERROR with reason MQFB\_CICS\_BRIDGE\_FAILURE.

### Authority

Components of the bridge need authority to either put to or get from the various WebSphere MQ queues. In summary:

- The monitor and all bridge tasks need authority to get messages from the bridge request queue.
- A bridge task need authority to put messages to its reply-to queue.
- To ensure any error replies are received, the monitor should have authority to put messages to all reply-to queues.
- Bridge tasks should have authority to put messages to the dead-letter queue.
- The monitor needs authority to put messages to the dead-letter queue, unless you want the bridge to stop if an error occurs.
- The monitor and all bridge tasks should have authority to put messages to the backout requeue queue, if one is defined

See Table 67 on page 198 to determine the correlation between user IDs and authority.

---

## Security considerations for using WebSphere MQ with IMS

The following section describes security considerations for using WebSphere MQ with IMS.

### Using the OPERCMDS class

If you are using RACF to protect resources in the OPERCMDS class, ensure that the userid associated with your WebSphere MQ queue manager address space has authority to issue the MODIFY command to any IMS system to which it can connect.

### Security considerations for the IMS bridge

There are four aspects that you must consider when deciding your security requirements for the IMS bridge, these are:

- What security authorization is needed to connect WebSphere MQ to IMS (“Connecting to IMS” on page 200)
- How much security checking is performed on applications using the bridge to access IMS (“Application access control” on page 200)

- Which IMS resources these applications are allowed to use (“Security checking on IMS” on page 202)
- What authority is to be used for messages that are put and got by the bridge (“Security checking done by the bridge” on page 202)

When you define your security requirements for the IMS bridge you must consider the following:

- Messages passing across the bridge might have originated from applications on platforms that do not offer strong security features
- Messages passing across the bridge might have originated from applications that are not controlled by the same enterprise or organization

## Connecting to IMS

The IMS bridge is an OTMA client. The connection to IMS operates under the user ID of the WebSphere MQ queue manager address space. This is normally defined as a member of the started task group. This user ID must be granted access to the OTMA group (unless the /SECURE OTMA setting is NONE).

To do this, define the following profile in the FACILITY class:

```
IMSXCF.xcfgname.mqxcfmname
```

Where xcfgname is the XCF group name and mqxcfmname is the XCF member name of WebSphere MQ.

You must give your WebSphere MQ queue manager user ID read access to this profile.

### Note:

1. If you change the authorities in the FACILITY class, you must issue the RACF command SETROPTS RACLIST(FACILITY) REFRESH to activate the changes.
2. If profile hlq.NO.SUBSYS.SECURITY exists in the MQADMIN class, no user ID will be passed to IMS and the connection will fail unless the /SECURE OTMA setting is NONE.

## Application access control

For each IMS system that the IMS bridge connects to, you can define the following RACF profile in the FACILITY class to determine how much security checking is performed for each message passed to the IMS system.

```
IMSXCF.xcfgname.imsxcfmname
```

Where xcfgname is the XCF group name and imsxcfmname is the XCF member name for IMS. (You need to define a separate profile for each IMS system.)

The access level you allow for the WebSphere MQ queue manager user ID in this profile is returned to WebSphere MQ when the IMS bridge connects to IMS, and indicates the level of security that is required on subsequent transactions. For subsequent transactions, WebSphere MQ requests the appropriate services from RACF and, where the user ID is authorized, passes the message to IMS.

OTMA does not support the IMS /SIGN command; however, WebSphere MQ allows you to set the access checking for each message to enable implementation of the necessary level of control.

The following access level information can be returned:

#### **NONE or NO PROFILE FOUND**

This indicates that maximum security is required, that is, authentication is required for every transaction. A check is made to verify that the user ID specified in the *UserIdentifier* field of the MQMD structure, and the password or passticket in the *Authenticator* field of the MQIHL structure are known to RACF, and are a valid combination. A Utoken is created with a password or passticket, and passed to IMS; the Utoken is not cached.

**Note:** If profile hlq.NO.SUBSYS.SECURITY exists in the MQADMIN class, this level of security overrides whatever is defined in the profile.

**READ** This indicates that the same authentication is to be performed as above under the following circumstances:

- The first time that a specific user ID is encountered
- When the user ID has been encountered before but the cached Utoken was not created with a password or passticket

WebSphere MQ requests a Utoken if required, and passes it to IMS.

**Note:** If a request to reverify security has been actioned, all cached information is lost and a Utoken is requested the first time each user ID is subsequently encountered.

#### **UPDATE**

A check is made that the user ID in the *UserIdentifier* field of the MQMD structure is known to RACF.

A Utoken is built and passed to IMS; the Utoken is cached.

#### **CONTROL/ALTER**

These indicate that no security Utokens need to be provided for any user IDs for this IMS system. (You would probably only use this for development and test systems.)

#### **Note:**

1. This access is defined when WebSphere MQ connects to IMS, and lasts for the duration of the connection. To change the security level, the access to the security profile must be changed and then the bridge stopped and restarted (for example, by stopping and restarting OTMA).
2. If you change the authorities in the FACILITY class, you must issue the RACF command SETROPTS RACLIST(FACILITY) REFRESH to activate the changes.
3. You can use a password or a passticket, but you must remember that the IMS bridge does not encrypt data. For information about using passtickets, see "Using RACF passtickets in the IMS header" on page 203.
4. Some of the above might be affected by security settings in IMS, using the /SECURE OTMA command.
5. Cached Utoken information is held for the duration defined by the INTERVAL and TIMEOUT parameters of the WebSphere MQ ALTER SECURITY command.

## Security checking on IMS

Each WebSphere MQ message that passes across the bridge contains the following security information:

- A user ID contained in the *UserIdentifier* field of the MQMD structure
- The security scope contained in the *SecurityScope* field of the MQIIH structure (if the MQIIH structure is present)
- A Utoken (unless the WebSphere MQ sub system has CONTROL or ALTER access to the relevant IMSXCF.xcfgname.imsxcfmname profile)

The security checks made depend on the setting by the IMS command `/SECURE OTMA`, as follows:

### **/SECURE OTMA NONE**

No security checks are made for the transaction.

### **/SECURE OTMA CHECK**

The *UserIdentifier* field of the MQMD structure is passed to IMS for transaction or command authority checking.

An ACEE (Accessor Environment Element) is built in the IMS control region.

### **/SECURE OTMA FULL**

The *UserIdentifier* field of the MQMD structure is passed to IMS for transaction or command authority checking.

An ACEE is built in the IMS dependent region as well as the IMS control region.

### **/SECURE OTMA PROFILE**

The *UserIdentifier* field of the MQMD structure is passed to IMS for transaction or command authority checking

The *SecurityScope* field in the MQIIH structure is used to determine whether to build an ACEE in the IMS dependent region as well as the control region.

### **Note:**

1. If you change the authorities in the TIMS or CIMS class, or the associated group classes GIMS or DIMS, you must issue the following IMS commands to activate the changes:
  - `/MODIFY PREPARE RACF`
  - `/MODIFY COMMIT`
2. If you do not use `/SECURE OTMA PROFILE`, any value specified in the *SecurityScope* field of the MQIIH structure is ignored.

## Security checking done by the bridge

When the bridge puts or gets a message, the following authorities are used:

### **Getting a message from the bridge queue**

No security checks are performed.

### **Putting an exception, or COA report message**

Uses the authority of the user ID in the *UserIdentifier* field of the MQMD structure.

### Putting a reply message

Uses the authority of the user ID in the *UserIdentifier* field of the MQMD structure of the original message

### Putting a message to the dead-letter queue

No security checks are performed.

#### Note:

1. If you change the WebSphere MQ class profiles, you must issue the WebSphere MQ REFRESH SECURITY(\*) command to activate the changes.
2. If you change the authority of a user, you must issue the MQSC RVERIFY SECURITY command to activate the change.

## Using RACF passtickets in the IMS header

If you want to use a passticket instead of a password in the IMS header (MQIIH), specify the application name against which the passticket will be validated in the PASSTKTA attribute of the STGCLASS definition of the IMS Bridge queue to which the message will be routed.

If the PASSTKTA value is left blank, you must arrange to have a passticket generated. The application name in this case must be of the form MVSxxxx, where xxxx is the SMFID of the z/OS system on which the target queue manager runs.

A passticket is built from a user ID, the target application name, and a secret key. It is an 8-byte value containing uppercase alphabetic and numeric characters. It can be used only once, and is valid for a 20 minute period. If a passticket is generated by a local RACF system, RACF only checks that the profile exists and not that the user has authority against the profile. If the passticket was generated on a remote system, RACF will validate the access of the userid to the profile. For full information about passtickets, see the *z/OS SecureWay Security Server RACF Security Administrator's Guide*.

Passtickets in IMS headers are given to RACF by WebSphere MQ, not IMS.

---

## Example security scenarios

This chapter describes two example security scenarios, showing the security settings required.

The first scenario uses two queue managers on z/OS, called QM1 and QM2. In the second scenario, the two queue managers are members of a queue-sharing group called QSGA. In this scenario, queue-sharing group level security is illustrated. These examples use uppercase RACF profiles.

### The two queue managers scenario

An application uses the MQPUT1 call to put messages to queues on queue manager QM1. Some of the messages are then forwarded to queues on QM2, using TCP and LU 6.2 channels. The TCP channels can either use SSL or not. The application could be a batch application or a CICS application, and the messages are put using the MQPMO\_SET\_ALL\_CONTEXT option. This is illustrated in Figure 32 on page 204.

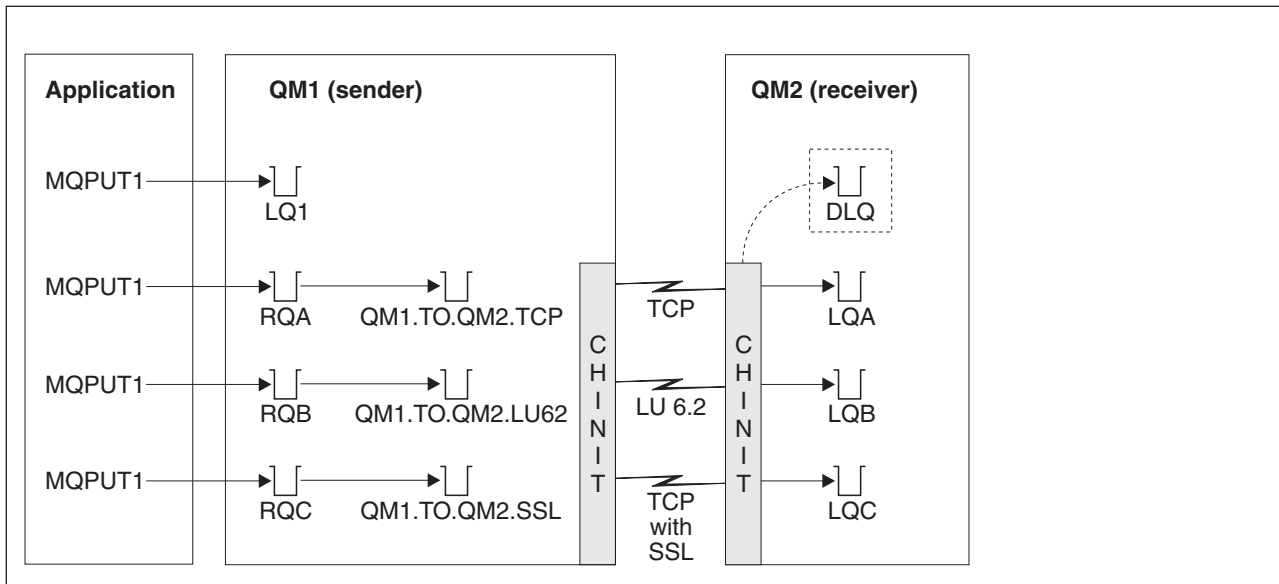


Figure 32. Example security scenario

The following assumptions are made about the queue managers:

- All the required WebSphere MQ definitions have been predefined or have been made through the CSQINP2 data set processed at queue manager startup.

If they have not, you need the appropriate access authority to the commands needed to define these objects.

- All the RACF profiles required have been defined and appropriate access authorities have been granted, before the queue manager and channel initiators started.

If they have not, you need the appropriate authority to issue the RACF commands required to define all the profiles needed and grant the appropriate access authorities to those profiles. You also need the appropriate authority to issue the MQSC security commands to start using the new security profiles.

- All digital certificates required have been created and connected to key rings. The digital certificate sent by QM1 as part of the SSL handshake is recognized by RACF on QM2's system, either because it is also installed in that RACF, or because a matching Certificate Name File (CNF) filter exists.

## Security switch settings

The following security switches are set for both queue managers:

- Subsystem security on
- Queue security on
- Alternate user security on
- Context security on
- Process security off
- Namelist security off
- Connection security on
- Command security on
- Command resource security on

The following profiles are defined in the MQADMIN class to turn process and namelist security off:

```
QM1.NO.PROCESS.CHECKS
QM1.NO.NLIST.CHECKS
QM2.NO.PROCESS.CHECKS
QM2.NO.NLIST.CHECKS
```

## WebSphere MQ object definitions

The following objects are defined on the two queue managers. The definitions use the defaults supplied with WebSphere MQ, unless otherwise stated.

### Queue manager QM1:

The following queues are defined on queue manager QM1:

**LQ1** A local queue.

**RQA** A remote queue definition, with the following attributes:

- RNAME(LQA)
- RQMNAME(QM2)
- XMITQ(QM1.TO.QM2.TCP)

**RQB** A remote queue definition, with the following attributes:

- RNAME(LQB)
- RQMNAME(QM2)
- XMITQ(QM1.TO.QM2.LU62)

**RQC** A remote queue definition, with the following attributes:

- RNAME(LQC)
- RQMNAME(QM2)
- XMITQ(QM1.TO.QM2.SSL)

### **QM1.TO.QM2.TCP**

A transmission queue.

### **QM1.TO.QM2.LU62**

A transmission queue.

### **QM1.TO.QM2.SSL**

A transmission queue.

The following channels are defined on QM1:

### **QM1.TO.QM2.TCP**

A sender channel definition, with the following attributes:

- CHLTYPE(SDR)
- TRPTYPE(TCP)
- XMITQ(QM1.TO.QM2.TCP)
- CONNAME(QM2TCP)

### **QM1.TO.QM2.LU62**

A sender channel definition, with the following attributes:

- CHLTYPE(SDR)
- TRPTYPE(LU62)

- XMITQ(QM1.TO.QM2.LU62)
- CONNAME(QM2LU62)

(See “Security considerations for distributed queuing” on page 192 for information about setting up APPC security.)

#### **QM1.TO.QM2.SSL**

A sender channel definition, with the following attributes:

- CHLTYPE(SDR)
- TRPTYPE(TCP)
- XMITQ(QM1.TO.QM2.SSL)
- CONNAME(QM2TCP)
- SSLCIPH(RC4\_MD5\_EXPORT)

#### **Queue manager QM2:**

The following queues have been defined on queue manager QM2:

**LQA** A local queue.

**LQB** A local queue.

**LQC** A local queue.

**DLQ** A local queue that is used as the dead-letter queue.

The following channels have been defined on QM2:

#### **QM1.TO.QM2.TCP**

A receiver channel definition, with the following attributes:

- CHLTYPE(RCVR)
- TRPTYPE(TCP)
- PUTAUT(CTX)
- MCAUSER(MCATCP)

#### **QM1.TO.QM2.LU62**

A receiver channel definition, with the following attributes:

- CHLTYPE(RCVR)
- TRPTYPE(LU62)
- PUTAUT(CTX)
- MCAUSER(MCALU62)

(See “Security considerations for distributed queuing” on page 192 for information about setting up APPC security.)

#### **QM1.TO.QM2.SSL**

A receiver channel definition, with the following attributes:

- CHLTYPE(RCVR)
- TRPTYPE(TCP)
- PUTAUT(CTX)
- MCAUSER(MCASSL)
- SSLCIPH(RC4\_MD5\_EXPORT)

#### **User IDs used in scenario**

The following user IDs are used:



**BATCHID**

Batch application (Job or TSO ID)

**MSGUSR***UserIdentifier* in MQMD (context user ID)**MOVER1**

QM1 channel initiator address space user ID

**MOVER2**

QM2 channel initiator address space user ID

**MCATCP**

MCAUSER specified on the TCP/IP without SSL receiver channel definition

**MCALU62**

MCAUSER specified on the LU 6.2 receiver channel definition

**MCASSL**

MCAUSER specified on the TCP/IP with SSL receiver channel definition

**CICSAD1**

CICS address space ID

**CICSTX1**

CICS task user ID

**CERTID**

The user ID associated by RACF with the flowed certificate.

**Security profiles and accesses required**

Table 69 through Table 74 on page 210 show the security profiles that are required to enable the scenario to work:

*Table 69. Security profiles for the example scenario*

Class	Profile	User ID	Access
MQCONN	QM1.CHIN	MOVER1	READ
MQADMIN	QM1.RESLEVEL	BATCHID CICSAD1 MOVER1	NONE
MQADMIN	QM1.CONTEXT.**	MOVER1	CONTROL
MQQUEUE	QM1.SYSTEM.COMMAND.INPUT	MOVER1	UPDATE
MQQUEUE	QM1.SYSTEM.CHANNEL.SYNCQ	MOVER1	UPDATE
MQQUEUE	QM1.SYSTEM.CHANNEL.INITQ	MOVER1	UPDATE
MQQUEUE	QM1.SYSTEM.COMMAND.REPLY.MODEL	MOVER1	UPDATE
MQQUEUE	QM1.SYSTEM.ADMIN.CHANNEL.EVENT	MOVER1	UPDATE
MQQUEUE	QM1.QM1.TO.QM2.TCP	MOVER1	ALTER
MQQUEUE	QM1.QM1.TO.QM2.LU62	MOVER1	ALTER
MQQUEUE	QM1.QM1.TO.QM2.SSL	MOVER1	ALTER
MQCONN	QM2.CHIN	MOVER2	READ
MQADMIN	QM2.RESLEVEL	MOVER2	NONE
MQADMIN	QM2.CONTEXT.**	MOVER2	CONTROL
MQQUEUE	QM2.SYSTEM.COMMAND.INPUT	MOVER2	UPDATE

Table 69. Security profiles for the example scenario (continued)

Class	Profile	User ID	Access
MQQUEUE	QM2.SYSTEM.CHANNEL.SYNCQ	MOVER2	UPDATE
MQQUEUE	QM2.SYSTEM.CHANNEL.INITQ	MOVER2	UPDATE
MQQUEUE	QM2.SYSTEM.COMMAND.REPLY.MODEL	MOVER2	UPDATE
MQQUEUE	QM2.SYSTEM.ADMIN.CHANNEL.EVENT	MOVER2	UPDATE
MQQUEUE	QM2.DLQ	MOVER2	UPDATE

### Security profiles required for a batch application:

The batch application runs under user ID BATCHID on QM1. It connects to queue manager QM1 and puts messages to the following queues:

- LQ1
- RQA
- RQB
- RQC

It uses the MQPMO\_SET\_ALL\_CONTEXT and MQPMO\_ALTERNATE\_USER\_AUTHORITY options. The alternate user ID found in the *UserIdentifier* field of the message descriptor (MQMD) is MSGUSR.

The following profiles are required on queue manager QM1:

Table 70. Sample security profiles for the batch application on queue manager QM1

Class	Profile	User ID	Access
MQCONN	QM1.BATCH	BATCHID	READ
MQADMIN	QM1.CONTEXT.**	BATCHID	CONTROL
MQQUEUE	QM1.LQ1	BATCHID	UPDATE
MQQUEUE	QM1.RQA	BATCHID	UPDATE
MQQUEUE	QM1.RQB	BATCHID	UPDATE
MQQUEUE	QM1.RQC	BATCHID	UPDATE

The following profiles are required on queue manager QM2 for messages put to queue RQA on queue manager QM1 (for the TCP/IP channel not using SSL):

Table 71. Sample security profiles for queue manager QM2 using TCP/IP and not SSL

Class	Profile	User ID	Access
MQADMIN	QM2.ALTERNATE.USER.MSGUSR	MCATCP MOVER2	UPDATE
MQADMIN	QM2.CONTEXT.**	MCATCP MOVER2	CONTROL
MQQUEUE	QM2.LQA	MOVER2 MSGUSR	UPDATE
MQQUEUE	QM2.DLQ	MOVER2 MSGUSR	UPDATE

### Notes:

1. The user ID passed in the MQMD of the message is used as the user ID for the **MQPUT1** on queue manager QM2 because the receiver channel was defined with PUTAUT(CTX) and MCAUSER(MCATCP).
2. The MCAUSER field of the receiver channel definition is set to MCATCP; this user ID is used in addition to the channel initiator address space user ID for the checks carried out against the alternate user ID and context profile.
3. The MOVER2 user ID and the *UserIdentifier* in the message descriptor (MQMD) are used for the resource checks against the queue.
4. The MOVER2 and MSGUSR user IDs both need access to the dead-letter queue so that messages that cannot be put to the destination queue can be sent there.
5. Two user IDs are checked on all three checks performed because RESLEVEL is set to NONE.

The following profiles are required on queue manager QM2 for messages put to queue RQB on queue manager QM1 (for the LU 6.2 channel):

*Table 72. Sample security profiles for queue manager QM2 using LU 6.2*

Class	Profile	User ID	Access
MQADMIN	QM2.ALTERNATE.USER.MSGUSR	MCALU62 MOVER1	UPDATE
MQADMIN	QM2.CONTEXT.**	MCALU62 MOVER1	CONTROL
MQQUEUE	QM2.LQB	MOVER1 MSGUSR	UPDATE
MQQUEUE	QM2.DLQ	MOVER1 MSGUSR	UPDATE

**Notes:**

1. The user ID passed in the MQMD of the message is used as the user ID for the **MQPUT1** on queue manager QM2 because the receiver channel was defined with PUTAUT(CTX) and MCAUSER(MCALU62).
2. The MCA user ID is set to the value of the MCAUSER field of the receiver channel definition (MCALU62).
3. Because LU 6.2 supports security on the communications system for the channel, the user ID received from the network is used as the channel user ID (MOVER1).
4. Two user IDs are checked on all three checks performed because RESLEVEL is set to NONE.
5. MCALU62 and MOVER1 are used for the checks performed against the alternate user ID and Context profiles, and MSGUSR and MOVER1 are used for the checks against the queue profile.
6. The MOVER1 and MSGUSR user IDs both need access to the dead-letter queue so that messages that cannot be put to the destination queue can be sent there.

The following profiles are required on queue manager QM2 for messages put to queue RQC on queue manager QM1 (for the TCP/IP channel using SSL):

*Table 73. Sample security profiles for queue manager QM2 using TCP/IP and SSL*

Class	Profile	User ID	Access
MQADMIN	QM2.ALTERNATE.USER.MSGUSR	MCASSL CERTID	UPDATE
MQADMIN	QM2.CONTEXT.**	MCASSL CERTID	CONTROL

Table 73. Sample security profiles for queue manager QM2 using TCP/IP and SSL (continued)

Class	Profile	User ID	Access
MQQUEUE	QM2.LQC	CERTID MSGUSR	UPDATE
MQQUEUE	QM2.DLQ	CERTID MSGUSR	UPDATE

**Notes:**

1. The user ID passed in the MQMD of the message is used as the user ID for the **MQPUT1** on queue manager QM2 because the receiver channel was defined with PUTAUT(CTX) and MCAUSER(MCASSL).
2. The MCA user ID is set to the value of the MCAUSER field of the receiver channel definition (MCASSL).
3. Because the certificate flowed by the channel from QM1 as part of the SSL handshake might be installed on QM2's system, or might match a certificate name filter on QM2's system, the user ID found during that matching is used as the channel user ID (CERTID).
4. Two user IDs are checked on all three checks performed because RESLEVEL is set to NONE.
5. MCASSL and CERTID are used for the checks performed against the alternate user ID and Context profiles, and MSGUSR and MOVER1 are used for the checks against the queue profile.
6. The CERTID and MSGUSR user IDs both need access to the dead-letter queue so that messages that cannot be put to the destination queue can be sent there.

**Security profiles required for a CICS application:**

The CICS application uses a CICS address space user ID of CICSAD1 and a CICS task user ID of CICSTX1. The security profiles required on queue manager QM1 are different to those required for the batch application. The profiles required on queue manager QM2 are the same as for the batch application.

The following profiles are required on queue manager QM1:

Table 74. Sample security profiles for the CICS application on queue manager QM1

Class	Profile	User ID	Access
MQCONN	QM1.CICS	CICSAD1	READ
MQADMIN	QM1.CONTEXT.**	CICSAD1 CICSTX1	CONTROL
MQQUEUE	QM1.LQ1	CICSAD1 CICSTX1	UPDATE
MQQUEUE	QM1.RQA	CICSAD1 CICSTX1	UPDATE
MQQUEUE	QM1.RQB	CICSAD1 CICSTX1	UPDATE

## The queue-sharing group scenario

An application uses the **MQPUT1** call to put messages to queues on queue manager QM1. Some of the messages are then forwarded to queues on QM2, using TCP and LU 6.2 channels. The application is a batch application, and the messages are put using the MQPMO\_SET\_ALL\_CONTEXT option. This is illustrated in Figure 32 on page 204.

The following assumptions are made about the queue managers:

- All the required WebSphere MQ definitions have been predefined or have been made through the CSQINP2 data set processed at queue manager startup.

If they have not, you need the appropriate access authority to the commands needed to define these objects.

- All the RACF profiles required have been defined and appropriate access authorities have been granted, before the queue manager and channel initiators started.

If they have not, you need the appropriate authority to issue the RACF commands required to define all the profiles needed and grant the appropriate access authorities to those profiles. You also need the appropriate authority to issue the MQSC security commands to start using the new security profiles.

## Security switch settings

The following security switches are set for the queue-sharing group:

- Subsystem security on
- Queue-sharing group security on
- Queue manager security off
- Queue security on
- Alternate user security on
- Context security on
- Process security off
- Namelist security off
- Connection security on
- Command security on
- Command resource security on

The following profiles are defined in the MQADMIN class to turn process, namelist, and queue-manager level security off:

```
QSGA.NO.PROCESS.CHECKS
QSGA.NO.NLIST.CHECKS
QSGA.NO.QMGR.CHECKS
```

## WebSphere MQ object definitions

The following objects are defined on the two queue managers. The definitions use the defaults supplied with WebSphere MQ, unless otherwise stated.

### Queue manager QM1:

The following queues are defined on queue manager QM1:

**LQ1** A local queue.

**RQA** A remote queue definition, with the following attributes:

- RNAME(LQA)
- RQMNAME(QM2)
- XMITQ(QM1.TO.QM2.TCP)

**RQB** A remote queue definition, with the following attributes:

- RNAME(LQB)
- RQMNAME(QM2)
- XMITQ(QM1.TO.QM2.LU62)

**QM1.TO.QM2.TCP**

A transmission queue.

**QM1.TO.QM2.LU62**

A transmission queue.

The following channels are defined on QM1:

**QM1.TO.QM2.TCP**

A sender channel definition, with the following attributes:

- CHLTYPE(SDR)
- TRPTYPE(TCP)
- XMITQ(QM1.TO.QM2.TCP)
- CONNAME(QM2TCP)

**QM1.TO.QM2.LU62**

A sender channel definition, with the following attributes:

- CHLTYPE(SDR)
- TRPTYPE(LU62)
- XMITQ(QM1.TO.QM2.LU62)
- CONNAME(QM2LU62)

(See “Security considerations for distributed queuing” on page 192 for information about setting up APPC security.)

**Queue manager QM2:**

The following queues have been defined on queue manager QM2:

**LQA** A local queue.

**LQB** A local queue.

**DLQ** A local queue that is used as the dead-letter queue.

The following channels have been defined on QM2:

**QM1.TO.QM2.TCP**

A receiver channel definition, with the following attributes:

- CHLTYPE(RCVR)
- TRPTYPE(TCP)
- PUTAUT(CTX)
- MCAUSER(MCATCP)

**QM1.TO.QM2.LU62**

A receiver channel definition, with the following attributes:

- CHLTYPE(RCVR)
- TRPTYPE(LU62)
- PUTAUT(CTX)
- MCAUSER(MCALU62)

(See “Security considerations for distributed queuing” on page 192 for information about setting up APPC security.)

## User IDs used in scenario

The following user IDs are used:

### BATCHID

Batch application (Job or TSO ID)

### MSGUSR

*UserIdentifier* in MQMD (context user ID)

### MOVER1

QM1 channel initiator address space user ID

### MOVER2

QM2 channel initiator address space user ID

### MCATCP

MCAUSER specified on the TCP/IP receiver channel definition

### MCALU62

MCAUSER specified on the LU 6.2 receiver channel definition

## Security profiles and accesses required

Table 75 through Table 78 on page 215 show the security profiles that are required to enable the scenario to work:

*Table 75. Security profiles for the example scenario*

Class	Profile	User ID	Access
MQCONN	QSGA.CHIN	MOVER1 MOVER2	READ
MQADMIN	QSGA.RESLEVEL	BATCHID MOVER1 MOVER2	NONE
MQADMIN	QSGA.CONTEXT.**	MOVER1 MOVER2	CONTROL
MQQUEUE	QSGA.SYSTEM.COMMAND.INPUT	MOVER1 MOVER2	UPDATE
MQQUEUE	QSGA.SYSTEM.CHANNEL.SYNCQ	MOVER1 MOVER	UPDATE
MQQUEUE	QSGA.SYSTEM.CHANNEL.INITQ	MOVER1 MOVER2	UPDATE
MQQUEUE	QSGA.SYSTEM.COMMAND.REPLY.MODEL	MOVER1 MOVER2	UPDATE
MQQUEUE	QSGA.SYSTEM.ADMIN.CHANNEL.EVENT	MOVER1 MOVER2	UPDATE
MQQUEUE	QSGA.SYSTEM.QSG.CHANNEL.SYNCQ	MOVER1 MOVER2	UPDATE
MQQUEUE	QSGA.SYSTEM.QSG.TRANSMIT.QUEUE	MOVER1 MOVER2	UPDATE
MQQUEUE	QSGA.QM1.TO.QM2.TCP	MOVER1	ALTER
MQQUEUE	QSGA.QM1.TO.QM2.LU62	MOVER1	ALTER
MQQUEUE	QSGA.DLQ	MOVER2	UPDATE

### Security profiles required for a batch application:

The batch application runs under user ID BATCHID on QM1. It connects to queue manager QM1 and puts messages to the following queues:

- LQ1
- RQA
- RQB

It uses the MQPMO\_SET\_ALL\_CONTEXT and MQPMO\_ALTERNATE\_USER\_AUTHORITY options. The alternate user ID found in the *UserIdentifier* field of the message descriptor (MQMD) is MSGUSR.

The following profiles are required on queue manager QM1:

*Table 76. Sample security profiles for the batch application on queue manager QM1*

Class	Profile	User ID	Access
MQCONN	QSGA.BATCH	BATCHID	READ
MQADMIN	QSGA.CONTEXT.**	BATCHID	CONTROL
MQQUEUE	QSGA.LQ1	BATCHID	UPDATE
MQQUEUE	QSGA.RQA	BATCHID	UPDATE
MQQUEUE	QSGA.RQB	BATCHID	UPDATE

The following profiles are required on queue manager QM2 for messages put to queue RQA on queue manager QM1 (for the TCP/IP channel):

*Table 77. Sample security profiles for queue manager QM2 using TCP/IP*

Class	Profile	User ID	Access
MQADMIN	QSGA.ALTERNATE.USER.MSGUSR	MCATCP MOVER2	UPDATE
MQADMIN	QSGA.CONTEXT.**	MCATCP MOVER2	CONTROL
MQQUEUE	QSGA.LQA	MOVER2 MSGUSR	UPDATE
MQQUEUE	QSGA.DLQ	MOVER2 MSGUSR	UPDATE

**Notes:**

1. The user ID passed in the MQMD of the message is used as the user ID for the **MQPUT1** on queue manager QM2 because the receiver channel was defined with PUTAUT(CTX) and MCAUSER(MCATCP).
2. The MCAUSER field of the receiver channel definition is set to MCATCP; this user ID is used in addition to the channel initiator address space user ID for the checks carried out against the alternate user ID and context profile.
3. The MOVER2 user ID and the *UserIdentifier* in the message descriptor (MQMD) are used for the resource checks against the queue.
4. The MOVER2 and MSGUSR user IDs both need access to the dead-letter queue so that messages that cannot be put to the destination queue can be sent there.
5. Two user IDs are checked on all three checks performed because RESLEVEL is set to NONE.

The following profiles are required on queue manager QM2 for messages put to queue RQB on queue manager QM1 (for the LU 6.2 channel):



Table 78. Sample security profiles for queue manager QM2 using LU 6.2

Class	Profile	User ID	Access
MQADMIN	QSGA.ALTERNATE.USER.MSGUSR	MCALU62 MOVER1	UPDATE
MQADMIN	QSGA.CONTEXT.**	MCALU62 MOVER1	CONTROL
MQQUEUE	QSGA.LQB	MOVER1 MSGUSR	UPDATE
MQQUEUE	QSGA.DLQ	MOVER1 MSGUSR	UPDATE

**Notes:**

1. The user ID passed in the MQMD of the message is used as the user ID for the **MQPUT1** on queue manager QM2 because the receiver channel was defined with PUTAUT(CTX) and MCAUSER(MCALU62).
2. The MCA user ID is set to the value of the MCAUSER field of the receiver channel definition (MCALU62).
3. Because LU 6.2 supports security on the communications system for the channel, the user ID received from the network is used as the channel user ID (MOVER1).
4. Two user IDs are checked on all three checks performed because RESLEVEL is set to NONE.
5. MCALU62 and MOVER1 are used for the checks performed against the alternate user ID and Context profiles, and MSGUSR and MOVER1 are used for the checks against the queue profile.
6. The MOVER1 and MSGUSR user IDs both need access to the dead-letter queue so that messages that cannot be put to the destination queue can be sent there.

---

## WebSphere MQ security implementation checklist

This chapter gives a step-by-step procedure you can use to work out and define the security implementation for each of your WebSphere MQ queue managers. Refer to other sections for details, in particular “Profiles used to control access to WebSphere MQ resources” on page 132.

If you require security checking, follow this checklist to implement it:

1. Activate the RACF MQADMIN (uppercase profiles) and MXADMIN (mixed case profiles) classes
  - Do you want security at queue-sharing group level, queue-manager level, or a combination of both?  
Refer to “Profiles to control queue-sharing group or queue manager level security” on page 128.
2. Do you need connection security?
  - **Yes:** Activate the MQCONN class. Define appropriate connection profiles at either queue manager level or queue-sharing group level in the MQCONN class and permit the appropriate users or groups access to these profiles.

**Note:** Only users of the **MQCONN** API request or CICS or IMS address space user IDs need to have access to the corresponding connection profile.

- **No:** Define an hlq.NO.CONNECT.CHECKS profile at either queue manager level or queue-sharing group level in the MQADMIN or MXADMIN class.
3. Do you need security checking on commands?
    - **Yes:** Activate the MQCMDS class. Define appropriate command profiles at either queue manager level or queue-sharing group level in the MQCMDS class and permit the appropriate users or groups access to these profiles.  
If you are using a queue-sharing group, you might need to include the user IDs used by the queue manager itself and the channel initiator, see “Setting up WebSphere MQ resource security” on page 186.
    - **No:** Define an hlq.NO.CMD.CHECKS profile for the required queue manager or queue-sharing group in the MQADMIN or MXADMIN class.
  4. Do you need security on the resources used in commands?
    - **Yes:** Ensure the MQADMIN or MXADMIN class is active. Define appropriate profiles for protecting resources on commands at either queue manager level or queue-sharing group level in the MQADMIN or MXADMIN class and permit the appropriate users or groups access to these profiles. Set the CMDUSER parameter in CSQ6SYSP to the default user ID to be used for command security checks.  
If you are using a queue-sharing group, you might need to include the user IDs used by the queue manager itself and the channel initiator, see “Setting up WebSphere MQ resource security” on page 186.
    - **No:** Define an hlq.NO.CMD.RESC.CHECKS profile for the required queue manager or queue-sharing group in the MQADMIN or MXADMIN class.
  5. Do you need queue security?
    - **Yes:** Activate the MQQUEUE or MXQUEUE class. Define appropriate queue profiles for the required queue manager or queue-sharing group in the MQQUEUE or MXQUEUE class and permit the appropriate users or groups access to these profiles.
    - **No:** Define an hlq.NO.QUEUE.CHECKS profile for the required queue manager or queue-sharing group in the MQADMIN or MXADMIN class.
  6. Do you need process security?
    - **Yes:** Activate the MQPROC or MXPROC class. Define appropriate process profiles at either queue manager or queue-sharing group level and permit the appropriate users or groups access to these profiles.
    - **No:** Define an hlq.NO.PROCESS.CHECKS profile for the appropriate queue manager or queue-sharing group in the MQADMIN or MXADMIN class.
  7. Do you need namelist security?
    - **Yes:** Activate the MQNLIST or MXNLIST class. Define appropriate namelist profiles at either queue manager level or queue-sharing group level in the MQNLIST or MXNLIST class and permit the appropriate users or groups access to these profiles.
    - **No:** Define an hlq.NO.NLIST.CHECKS profile for the required queue manager or queue-sharing group in the MQADMIN or MXADMIN class.
  8. Do you need topic security?
    - **Yes:** Activate the MXTOPIC class. Define appropriate topic profiles at either queue manager level or queue-sharing group level in the MXTOPIC class and permit the appropriate users or groups access to these profiles.
    - **No:** Define an hlq.NO.TOPIC.CHECKS profile for the required queue manager or queue-sharing group in the MQADMIN or MXADMIN class.
  9. Do any users need to protect the use of the MQOPEN or MQPUT1 options relating to the use of context?

- **Yes:** Ensure the MQADMIN or MXADMIN class is active. Define hlq.CONTEXT.queueename profiles at the queue, queue manager, or queue-sharing group level in the MQADMIN or MXADMIN class and permit the appropriate users or groups access to these profiles.
  - **No:** Define an hlq.NO.CONTEXT.CHECKS profile for the required queue manager or queue-sharing group in the MQADMIN or MXADMIN class.
10. Do you need to protect the use of alternate user IDs?
- **Yes:** Ensure the MQADMIN or MXADMIN class is active. Define the appropriate hlq.ALTERNATE.USER.*alternateuserid* profiles for the required queue manager or queue-sharing group and permit the required users or groups access to these profiles.
  - **No:** Define the profile hlq.NO.ALTERNATE.USER.CHECKS for the required queue manager or queue-sharing group in the MQADMIN or MXADMIN class.
11. Do you need to tailor which user IDs are to be used for resource security checks through RESLEVEL?
- **Yes:** Ensure the MQADMIN or MXADMIN class is active. Define an hlq.RESLEVEL profile at either queue manager level or queue-sharing group level in the MQADMIN or MXADMIN class and permit the required users or groups access to the profile.
  - **No:** Ensure that no generic profiles exist in the MQADMIN or MXADMIN class that could apply to hlq.RESLEVEL. Define an hlq.RESLEVEL profile for the required queue manager or queue-sharing group and ensure that no users or groups have access to it.
12. Do you need to 'time out' unused user IDs from WebSphere MQ?
- **Yes:** Determine what timeout values you would like to use and issue the MQSC ALTER SECURITY command to change the TIMEOUT and INTERVAL parameters.
  - **No:** Issue the MQSC ALTER SECURITY command to set the INTERVAL value to zero.
- Note:** Update the CSQINP1 initialization input data set used by your subsystem so that the MQSC ALTER SECURITY command is issued automatically at every queue manager start up.
13. Do you use distributed queuing?
- **Yes:** Determine the appropriate MCAUSER attribute value for each channel, and provide suitable channel security exits.
14. Do you want to use the Secure Sockets Layer (SSL)?
- **Yes:** Plan your SSL infrastructure. Install the System SSL feature of z/OS. In RACF, set up your certificate name filters (CNFs), if you are using them, and your digital certificates. Set up your SSL key ring. Ensure that the SSLKEYR queue manager attribute is nonblank and points to your SSL key ring, and ensure that the value of SSLTASKS is at least 2.
  - **No:** Ensure that SSLKEYR is blank, and SSLTASKS is zero.
- For further details about SSL, see WebSphere MQ Security.
15. Do you use clients?
- **Yes:** Determine the appropriate MCAUSER attribute value for each server-connection channel, and provide suitable channel security exits if required.
16. Check your switch settings.

WebSphere MQ issues messages at queue manager startup that display your security settings. Use these messages to determine whether your switches are set correctly. For an example of these messages, see the WebSphere MQ for z/OS System Administration Guide.

## Chapter 8. Upgrading and applying service to Language Environment or z/OS Callable Services

The actions you must take vary according to whether you use CALLLIBS or LINK, and your version of SMP/E.

The following tables show you what you need to do to WebSphere MQ for z/OS if you upgrade your level of, or apply service to, the following products:

- Language Environment
- z/OS Callable Services (APPC and RRS for example)

Table 79. Service has been applied or the product has been upgraded to a new release

Product	Action if using CALLLIBS and SMP/E V3r2 or later <b>Note:</b> You do not need to run three separate jobs for TCP/IP, Language Environment and Callable services. One job will suffice for all products.	Action if using LINK
Language Environment	<ol style="list-style-type: none"> <li>1. Set the Boundary on your SMP/E job to the Target zone.</li> <li>2. On the SMP_CNTL card specify LINK LMODS CALLLIBS. You can also specify other parameters such as CHECK, RETRY(YES) and RC. See <i>SMP/E for z/OS and OS/390: Commands</i> for further information.</li> <li>3. Run the SMP/E job.</li> </ol>	No action required provided that the SMP/E zones were set up for automatic relinking, and the CSQ8LDQM job has been run.
Callable Services	<ol style="list-style-type: none"> <li>1. Set the Boundary on your SMP/E job to the Target zone.</li> <li>2. On the SMP_CNTL card specify LINK LMODS CALLLIBS. You can also specify other parameters such as CHECK, RETRY(YES) and RC. See <i>SMP/E for z/OS and OS/390: Commands</i> for further information.</li> <li>3. Run the SMP/E job.</li> </ol>	No action required provided that the SMP/E zones were set up for automatic relinking, and the CSQ8LDQM job has been run.

Table 80. One of the products has been updated to a new release in a new SMP/E environment and libraries

Product	Action if using CALLLIBS and SMP/E V3r2 or later <b>Note:</b> You do not need to run three separate jobs for Language Environment and Callable services. One job will suffice for both products.	Action if using LINK
Language Environment	<ol style="list-style-type: none"> <li>1. Change the DDDEFs for SCEELKED and SCEESPC to point to the new library.</li> <li>2. Set the Boundary on your SMP/E job to the Target zone.</li> <li>3. On the SMPCNTL card specify LINK LMODS CALLLIBS. You can also specify other parameters such as CHECK, RETRY(YES) and RC. See <i>SMP/E for z/OS and OS/390: Commands</i> for further information.</li> <li>4. Run the SMP/E job.</li> </ol>	<ol style="list-style-type: none"> <li>1. Delete the XZMOD subentries for the following LMOD entries in the WebSphere MQ for z/OS target zone: CMQXDCST, CMQXRCTL, CMQXSUPR, CSQCBE00, CSQCBE30, CSQCBP00, CSQCBP10, CSQCBR00, CSQUCVX, CSQUDLQH, CSQVXPCB, CSQVXSPT, CSQXDCST, CSQXRCTL, CSQXSUPR, CSQXTCMI, CSQXTCP, CSQXTNSV, CSQ7DRPS, IMQB23IC, IMQB23IM, IMQB23IR, IMQS23IC, IMQS23IM, IMQS23IR</li> <li>2. Set up the appropriate ZONEINDEXs between the WebSphere MQ zones and the Language Environment zones.</li> <li>3. Tailor CSQ8LDQM to refer to the new zone on the FROMZONE parameter of the LINK commands. CSQ8LDQM can be found in the SCSQINST library.</li> <li>4. Run CSQ8LDQM.</li> </ol>
Callable services	<ol style="list-style-type: none"> <li>1. Change the DDDEF for CSSLIB to point to the new library</li> <li>2. Set the Boundary on your SMP/E job to the Target zone.</li> <li>3. On the SMPCNTL card specify LINK LMODS CALLLIBS. You can also specify other parameters such as CHECK, RETRY(YES) and RC. See <i>SMP/E for z/OS and OS/390: Commands</i> for further information.</li> <li>4. Run the SMP/E job.</li> </ol>	<ol style="list-style-type: none"> <li>1. Delete the XZMOD subentries for the following LMOD entries in the WebSphere MQ for z/OS target zone: CMQXRCTL, CMQXSUPR, CSQBSRV, CSQILPLM, CSQXJST, CSQXRCTL, CSQXSUPR, CSQ3AMGP, CSQ3EPX, CSQ3REPL</li> <li>2. Set up the appropriate ZONEINDEXs between the WebSphere MQ zones and the Callable Services zones.</li> <li>3. Tailor CSQ8LDQM to refer to the new zone on the FROMZONE parameter of the LINK commands. CSQ8LDQM can be found in the SCSQINST library.</li> <li>4. Run CSQ8LDQM.</li> </ol>

## Running a REPORT CALLLIBS job

When running a REPORT CALLLIBS job there is an option that instructs SMP/E to place a user created job card at the front of the output data set. The option (JOB CARD) requires a DDDEF that points to a PDS containing the job card. This DDDEF can be defined to SMP/E using the example job step shown in Figure 33 on page 221.

```

//*****
//* DEFINE THE LOCATION TO SMP/E OF THE JOBCARD CALLLIBS IS TO
//* USE WHEN CREATING THE LINK-EDIT JOB.
//*****
//DEFJCARD EXEC PGM=GIMSMP,REGION=4096K
//SMPCSI DD DSN=your.csi,
//      DISP=SHR
//SYSPRINT DD SYSOUT=*
//SMPCNTL DD *
//      SET BDY(GLOBAL).
//      UCLIN .
//      REP DDDEF(MQJCARD)
//      DA(job.card.data.set) SHR .
//      ENDUCL .
/*

```

Figure 33. Example SMP/E JOBCARD job step

To run a REPORT CALLLIBS job you need to supply the following:

- The data set names of SMP/E CSI that contains WebSphere MQ for z/OS
- The data set into which the linkedit job is to be placed

In the SMP/E commands you need to provide:

- The DDDEFs of the products that have been upgraded or updated (see Table 79 on page 219 and Table 80 on page 220 for a list of possible DDDEFs)
- The DDDEF of the PDS containing the JOBCARD (MQJCARD in the example below)
- The name of the member containing the JOBCARD

Figure 34 shows an example SMP/E job that can be used as a basis for your own job.

```

//*****
//* RUN REPORT CALLLIBS.
//*****
//CALLLIBS EXEC PGM=GIMSMP,REGION=4096K
//SMPCSI DD DSN=your.csi,
//      DISP=SHR
//SMPPUNCH DD DSN=your.calllibs.punched.output,
//      DISP=(MOD,CATLG),
//      UNIT=SYSDA,
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=8800),
//      SPACE=(8800,(10,2))
//SYSPRINT DD SYSOUT=*
//SMPCNTL DD *
//      SET BDY(GLOBAL).
//      REPORT CALLLIBS(dddef)
//      ZONES(target zone)
//      JOBCARD(MQJCARD,jobcard member name).
/*

```

Figure 34. Example SMP/E REPORT CALLLIBS job

## Running a LINK CALLLIBS job

An example job to relink modules when using CALLLIBS.

The following is an example of the job to relink modules when using CALLLIBS on a SMP/E V3r2 system. You must provide a JOBCARD (see “Running a REPORT CALLLIBS job” on page 220) and the data set name of SMP/E CSI that contains

## WebSphere MQ for z/OS.

```
//*****  
//* RUN LINK CALLLIBS.  
//*****  
//CALLLIBS EXEC PGM=GIMSMP,REGION=4096K  
//SMPCSI DD DSN=your.csi  
// DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SMPCNTL DD *  
SET BDY(TZONE).  
LINK LMODS CALLLIBS .  
/*
```

Figure 35. Example SMP/E LINK CALLLIBS job



---

## Chapter 9. Using OTMA exits in IMS

If you want to send output from an IMS transaction to WebSphere MQ, and that transaction did not originate in WebSphere MQ, you need to code one or more IMS OTMA exits.

Similarly if you want to send output to a non-OTMA destination, and the transaction did originate in WebSphere MQ, you also need to code one or more IMS OTMA exits.

The following exits are available in IMS to enable you to customize processing between IMS and WebSphere MQ:

- An OTMA pre-routing exit
- A destination resolution user (DRU) exit

---

### Exit names

You must name the pre-routing exit DFSYPRX0. You can name the DRU exit anything, as long as it does not conflict with a module name already in IMS.

### Specifying the destination resolution user exit name

You can use the *Druexit* parameter of the OTMACON keyword of the CSQ6SYSP macro to specify the name of the OTMA DRU exit to be run by IMS.

We suggest you adopt a naming convention of DRU0xxxx, where xxxx is the name of your WebSphere MQ queue manager.

If you do not specify the name of a DRU exit in the OTMACON parameter, the default is DFSYDRU0. A sample of this module is supplied by IMS. See the *IMS/ESA® Customization Guide* for information about this.

### Naming convention for IMS destination

You need a naming convention for the destination to which you send the output from your IMS program. This is the destination that is set in the CHNG call of your IMS application, or that is preset in the IMS PSB.

---

### A sample scenario

We suggest the OTMA destination name is synonymous with the WebSphere MQ queue manager name, for example the WebSphere MQ queue manager name repeated. (In this case, if the WebSphere MQ queue manager name is VCPE, the destination set by the CHNG call is VCPEVCPE.)

### The pre-routing exit DFSYPRX0

You must first code a pre-routing exit DFSYPRX0. Parameters passed to this routine by IMS are documented in *IMS/ESA Customization Guide*.

This exit tests whether the message is intended for a known OTMA destination (in our example VCPEVCPE). If it is, the exit must check whether the transaction sending the message originated in OTMA. If so, it will already have an OTMA header, so you should exit from DFSYPRX0 with register 15 set to zero.

- If the transaction sending the message did not originate in OTMA, you must set the client name to be a valid OTMA client. This is the XCF member-name of the WebSphere MQ queue manager to which you want to send the message. The *IMS/ESA Customization Guide* tells you where to set this. We suggest you set your client name (in the OTMACON parameter of the CSQ6SYSP macro) to be the queue manager name. This is the default. You should then exit from DFSYPRX0 setting register 15 to 4.
- If the transaction sending the message originated in OTMA, and the destination is non-OTMA, you should set register 15 to 8 and exit.
- In all other cases, you should set register 15 to zero.

If you set the OTMA client name to one that is not known to IMS, your application CHNG or ISRT call returns an A1 status code.

For an IMS system communicating with more than one WebSphere MQ queue manager, you should repeat the logic above for each WebSphere MQ queue manager.

Sample assembler code to achieve the above is shown in Figure 36 on page 225:

```

        TITLE 'DFSYPRX0: OTMA PRE-ROUTING USER EXIT'
DFSYPRX0 CSECT
DFSYPRX0 AMODE 31
DFSYPRX0 RMODE ANY
*
        SAVE (14,12),,DFSYPRX0&SYSDATE&SYSTIME
        SPACE 2
        LR R12,R15          MODULE ADDRESSABILITY
        USING DFSYPRX0,R12
*
        L R2,12(,R1)       R2 -> OTMA PREROUTE PARMS
*
        LA R3,48(,R2)      R3 AT ORIGINAL OTMA CLIENT (IF ANY)
        CLC 0(16,R3),=XL16'00'  OTMA ORIG?
        BNE OTMAIN        YES, GO TO THAT CODE
*
NOOTMAIN DS 0H           NOT OTMA INPUT
        LA R5,8(,R2)      R5 IS AT THE DESTINATION NAME
        CLC 0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
        BNE EXIT0        NO, NORMAL PROCESSING
*
        L R4,80(,R2)     R4 AT ADDR OF OTMA CLIENT
        MVC 0(16,R4),=CL16'VCPE' CLIENT OVERRIDE
        B EXIT4          AND EXIT
*
OTMAIN DS 0H            OTMA INPUT
        LA R5,8(,R2)     R5 IS AT THE DESTINATION NAME
        CLC 0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
        BNE EXIT8        NO, NORMAL PROCESSING
*
EXIT0 DS 0H
        LA R15,0         RC = 0
        B BYEBYE
*
EXIT4 DS 0H
        LA R15,4         RC = 4
        B BYEBYE
*
EXIT8 DS 0H
        LA R15,8         RC = 8
        B BYEBYE
*
BYEBYE DS 0H
        RETURN (14,12),,RC=(15) RETURN WITH RETURN CODE IN R15
        SPACE 2
        REQUATE
        SPACE 2
        END

```

Figure 36. OTMA pre-routing exit assembler sample

## The destination resolution user exit

If you have set register 15 to 4 in DFSYPRX0, or if the source of the transaction was OTMA *and* you set Register 15 to zero, your DRU exit is invoked. In our example, the DRU exit name is DRU0VCPE.

The DRU exit checks if the destination is VCPEVCPE. If it is, it sets the OTMA user data (in the OTMA prefix) as follows:

**Offset OTMA user data**

**(decimal)**

- 0 OTMA user data length (in this example, 334)
- 2 MQMD
- 326 Reply to format

These offsets are where the WebSphere MQ-IMS bridge expects to find this information.

We suggest that the DRU exit is as simple as possible. Therefore, in this sample, all messages originating in IMS for a particular WebSphere MQ queue manager are put to the same WebSphere MQ queue.

If the message needs to be persistent, IMS must use a synchronized transaction pipe. To do this, the DRU exit must set the OUTPUT flag. For further details, please refer to the *IMS/ESA Customization Guide*.

You should write a WebSphere MQ application to process this queue, and use information from the MQMD structure, the MQIIH structure (if present), or the user data, to route each message to its destination.

A sample assembler DRU exit is shown in Figure 37 on page 227.

```

        TITLE 'DRU0VCPE: OTMA DESTINATION RESOLUTION USER EXIT'
DRU0VCPE CSECT
DRU0VCPE AMODE 31
DRU0VCPE RMODE ANY
*
        SAVE (14,12),,DRU0VCPE&SYSDATE&SYSTIME
        SPACE 2
        LR R12,R15                                MODULE ADDRESSABILITY
        USING DRU0VCPE,R12
*
        L R2,12(,R1)                               R2 -> OTMA DRU PARMS
*
        L R5,88(,R2)                               R5 ADDR OF OTMA USERDATA
        LA R6,2(,R5)                               R6 ADDR OF MQMD
        USING MQMD,R6                             AS A BASE
*
        LA R4,MQMD_LENGTH+10                       SET THE OTMA USERDATA LEN
        STH R4,0(,R5)                              = LL + MQMD + 8
*
        MVI 0(R6),X'00'                            ...NULL FIRST BYTE
        MVC 1(255,R6),0(R6)                        ...AND PROPAGATE IT
        MVC 256(MQMD_LENGTH-256+8,R6),255(R6)    ...AND PROPAGATE IT
*
VCPE    DS 0H
        CLC 44(16,R2),=CL16'VCPE'                 IS DESTINATION VCPE?
        BNE EXIT4                                 NO, THEN DEST IS NON-OTMA
        MVC MQMD_REPLYTOQ,=CL48'IMS.BRIDGE.UNSOLICITED.QUEUE'
        MVC MQMD_REPLYTOQMGR,=CL48'VCPE'          SET QNAME AND QMGRNAME
        MVC MQMD_FORMAT,MQFMT_IMS                 SET MQMD FORMAT NAME
        MVC MQMD_LENGTH(8,R6),MQFMT_IMS_VAR_STRING
*
                                                SET REPLYTO FORMAT NAME
        B EXIT0
*
EXIT0   DS 0H
        LA R15,0                                  SET RC TO OTMA PROCESS
        B BYEBYE                                  AND EXIT
*
EXIT4   DS 0H
        LA R15,4                                  SET RC TO NON-OTMA
        B BYEBYE                                  AND EXIT
*
BYEBYE  DS 0H
        RETURN (14,12),,RC=(15)                  RETURN CODE IN R15
        SPACE 2
        REQUATE
        SPACE 2
        CMQA EQUONLY=NO
        CMQMDA DSECT=YES
        SPACE 2
        END

```

Figure 37. Sample assembler DRU exit



---

## Notices

This information was developed for products and services offered in the United States. IBM may not offer the products, services, or features discussed in this information in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this information. The furnishing of this information does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing,  
IBM Corporation,  
North Castle Drive,  
Armonk, NY 10504-1785,  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation,  
Licensing,  
2-31 Roppongi 3-chome, Minato-k,u  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Laboratories,  
Mail Point 151,  
Hursley Park,  
Winchester,  
Hampshire,  
England  
SO21 2JN.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX	CICS	DB2
IBM	IBMLink	IMS
Language Environment	MQSeries	MVS
OS/390	RACF	RAMAC
REXX	RMF	SecureWay
Tivoli	WebSphere	z/OS

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.



Other company, product, or service names may be trademarks or service marks of others.



---

# Index

## Special characters

% character in RACF profiles 125  
&ZSEL 48

## A

access  
  if incorrect 191  
  restricting by using alias queues 138  
access method services (AMS), defining  
  page sets 25  
accounting  
  data 104  
  introduction 81  
  message manager 109  
  queue level 110  
  rules for data collection 83  
  sample SMF records 109  
  SMF trace 85  
  starting automatically 34  
  thread level 110  
active log  
  input buffer size (INBUFF) 36  
  number of buffers per write 38  
  output buffer  
    number filled (WRTHRSH) 38  
    size (OUTBUFF) 37  
  single or dual (TWOACTV) 37  
adding a structure 19  
address space user ID 167, 170  
administrative structure 19  
age, specifying for OTMA 31  
ALCUNIT parameter of CSQ6ARVP 39  
alias queues  
  command resource checking 162  
  restricting access using 138  
  security 138, 141  
  undelivered messages 141  
ALL attribute of DISPLAY  
  SECURITY 183  
alter queue attributes, security 152  
ALTER SECURITY command 180  
alternate user ID  
  distributed queuing 168  
  intra-group queuing 168  
alternate user security  
  description 147  
  implementing 217  
AMS (access method services), defining  
  page sets 25  
APF authorization of load libraries 7  
API-crossing exit  
  defining 68  
  including the load module 66  
API-resource security  
  quick reference 143  
  RESLEVEL 163  
APPC  
  applying service 219  
  example security scenario 203

APPC (*continued*)  
  security 193  
APPC channels, user IDs used 174  
applId node name 67  
archive initialization parameters,  
  setting 28  
archive log  
  cataloging (CATALOG) 41  
  compacting (COMPACT) 42  
  data set  
    name prefix 39  
    password protection  
      (PROTECT) 43  
    time stamp (TSTAMP) 44  
  deallocate period 36  
  device type (UNIT) 44  
  input buffer size (INBUFF) 36  
  maximum number in BSDS  
    (MAXARCH) 36  
  maximum number of tape units 37  
  mounting, WTOR (ARCWTOR) 40  
  optimize tape handling 36  
  output buffer size (OUTBUFF) 37  
  quiesce time (QUIESCE) 44  
  retention period (ARCRETN) 40  
  route codes (ARCWRTC) 40  
  single or dual (TWOARCH) 38  
  space allocation  
    block size (BLKSIZE) 41  
    primary (PRIQTY) 42  
    secondary (SECQTY) 44  
    units (ALCUNIT) 39  
archive parameter  
  default 38  
  setting 38  
archiving, controlling using OFFLOAD  
  parameter of CSQ6LOGP 37  
ARCPFX1 parameter of CSQ6ARVP 39  
ARCPFX2 parameter of CSQ6ARVP 40  
ARCRETN parameter of CSQ6ARVP 40  
ARCWRTC parameter of CSQ6ARVP 40  
ARCWTOR parameter of CSQ6ARVP 40  
ARM (automatic restart manager)  
  security 194  
audit, security 188  
auditing RESLEVEL 189  
autoinstall, CICS 64  
automatic restart manager (ARM)  
  security 194  
automating starting of CKTI 196

## B

base function 1  
batch  
  example security scenario 203  
  improving application portability 47  
  testing customization 53  
batch assembler, IVP 53  
batch connections  
  definition 133

batch connections (*continued*)  
  RESLEVEL 164  
  security 133  
  security checking 164  
  user IDs, security checking 171  
Batch/TSO adapter  
  installing 46  
  maximum number of connections  
    background (IDBACK) 30  
    foreground (IDFORE) 30  
  z/OS SNAP dump 47  
blank user IDs 178  
BLKSIZE parameter of CSQ6ARVP 41  
BSDS  
  creating 24  
  maximum number of log volumes  
    (MAXARCH) 36  
  preparation 24  
  single or dual (TWOBSDS) 38  
buffer  
  input buffer size (INBUFF) 36  
  number filled before write to log 38  
  output buffer size (OUTBUFF) 37  
buffer manager  
  suppressing console messages 49  
buffer manager statistics 95  
buffer pool  
  defining 22  
  size 95

## C

C and C++, testing customization 61  
callable services, applying service 219  
CATALOG parameter of CSQ6ARVP 41  
catalog, archive log (CATALOG) 41  
CCSID  
  queue manager (QMCCSID) 32  
CF structure  
  adding 19  
  naming 19  
CFRM policy  
  activating 19  
  data set 19  
CFSTRUCT name, what to specify 19  
change log inventory utility (CSQJU003)  
  BSDS preparation 24  
  log data set preparation 24  
Change Security command 180  
channel  
  security 186, 193  
  suppressing console messages 49  
  user ID 168  
channel initiator  
  connection security 135  
  defining the procedure 16  
  installation verification program 58  
  security considerations 192  
  SYSTEM.\* queue security 142  
  tailoring parameters 45  
  user IDs used 173

- channel name
  - meaning 119
- channel user ID, security 192
- checkpoint, number of log records (LOGLOAD) 30
- CHIADAPS
  - setting 45
- CHIDISPS
  - setting 45
- Chinese language feature 4
- CHINIT
  - tailoring parameters 45
- CICS
  - address space user ID, security checking 170
  - autoinstall 64
  - completing the connection 66
  - connection security 134
  - example security scenario 203
  - monitoring facility 87
  - system definition (CSD) data set 63
  - testing customization 61
  - user IDs
    - for security 170
    - security checking 171
- CICS adapter
  - API-crossing exit 68
  - CSQCSTUB stub program 66
  - customizing 67
  - installing 63
  - PLTPI program, writing 67
  - resource definition 63
  - security
    - authorization 195
    - for transactions 195
    - PLTPI 196
    - PLTSD 196
    - terminal user IDs 196
    - user IDs 196
  - statistics 82
  - system definition 65
  - testing customization 61
  - transaction services
    - security support 195
  - user IDs for security 196
- CICS bridge
  - customizing 68
  - prerequisite APARs 68
  - security 197
  - tuning considerations 71
- CICS connection
  - RESLEVEL 165
  - security
    - checking 165
    - user IDs for security 165
- CKBM security 196
- CKCN security 196
- CKDL security 196
- CKDP security 196
- CKQC
  - authorization 195
  - security 196
- CKRS security 196
- CKRT security 196
- CKSD security 196
- CKSG MCA transaction 197
- CKSQ security 196
- CKTI transaction
  - automating starting of 196
  - propagating user IDs 197
  - security 195, 196
- client attachment feature 1
- clients, security 175
- close options, dynamic queues 140
- cluster support
  - security considerations 194
- CMDUSER parameter of CSQ6SYSP 30
- COBOL, testing customization 61
- coded character set identifier, queue manager (QMCCSID) 32
- command prefix string (CPF)
  - defining 12
  - establishing 11
  - in a sysplex environment 13
- command security
  - implementing 216
- commands
  - DISPLAY 81
  - implementing resource security checking 216
  - resource checking summary table 152
  - resource security profiles 161
  - security profiles 151
- COMPACT parameter of CSQ6ARVP 42
- compacting archive logs (COMPACT) 42
- configuring, Secure Sockets Layer (SSL) 186
- connection parameters, setting 28
- connection security
  - implementing 215
  - IMS bridge 200
- connections
  - controlling CICS 67
  - maximum number
    - Batch/TSO, background (IDBACK) 30
    - Batch/TSO, foreground (IDFORE) 30
  - profiles for security 133
  - starting from
    - PLTPI program 64
- console messages, suppressing 49
- console security 170
- context security
  - implementing 216
  - profiles 149
- controlling application connections 67
- Coupling Facility (CF)
  - adding a structure 19
  - customization 19
  - testing 57
- Coupling Facility manager statistics 103
- CPF (command prefix string)
  - defining 12
  - establishing 11
  - in a sysplex environment 13
- cross reference data
  - thread 108
- CSD (CICS system definition data set) 63
- CSQ\_ADMIN structure 19
- CSQ4IVP1 installation verification program
  - overview 53
  - RACF commands 54
- CSQ4IVPX installation verification program
  - example output 60
  - overview 58
  - RACF commands 59
- CSQ4MPFL information message suppression sample 50
- CSQ4MSRR sample started task procedure 15
- CSQ4MSTR sample started task procedure 15
- CSQ4PAGE page set sample 25
- CSQ5PQSG (queue-sharing group utility) 26
- CSQ6ARVP macro 26, 38
- CSQ6LOGP macro 26, 35
- CSQ6SYSP
  - macro 26, 28
  - QINDXBLD parameter 32
- CSQBDEFV 46, 47
- CSQCAPX sample API-crossing exit program 68
- CSQCCODF sample PLTPI program 64
- CSQCSTUB for CICS adapter 66
- CSQDQ5ST 98
- CSQINP1
  - issuing commands from 21
  - sample data set 22
  - security 170
- CSQINP2
  - issuing commands from 21
  - security 170
  - updating 66
  - using 23
- CSQINPV
  - sample data set 22
- CSQINPX
  - security 193
  - using 23
- CSQOREXX 48
- CSQQDEFV, subsystem definition table 73, 78
- CSQQDEFX macro 77
- CSQSNAP 47
- CSQUTIL
  - RESLEVEL 165
  - security checking 165
  - SYSTEM.\* queue security 142
- CSQWDMP 49
- CSQWDPRD 49
- CSQZPARM
  - creating 26
  - displaying settings 26
- customizing
  - before you start 4
  - CICS adapter 67
  - CICS bridge 68
  - coupling facility related sysplex tasks 19
  - DB2 related sysplex tasks 17
  - IMS bridge 79
  - initialization input data sets 21
  - introduction 3

- customizing (*continued*)
  - IPL 4
  - overview 1
  - planning 1
  - security 190
  - summary 5
  - tasks 4
  - testing 53
  - WebSphere MQ 1
  - when migrating from previous versions 51
- D**
- data manager statistics 95
- data sets
  - bootstrap, creating 24
  - initialization 21
  - log, creating 24
  - RACF authorization 185
  - security 185
- data-sharing group, adding WebSphere MQ entries 25
- DB2
  - adding WebSphere MQ entries to the data-sharing group 25
  - customization 17
  - drop a database 18
  - group attachment 33
  - number of BLOB tasks 33
  - number of servers 33
  - testing 57
- DB2 manager statistics 98
- dead-letter queue, security 141
- DEALLCT parameter of CSQ6LOGP 36
- default
  - archive parameters 38
  - CCSID 32
  - CSQ6ARVP macro 38
  - CSQ6LOGP macro 35
  - CSQ6SYSP macro 28
  - logging parameters 35
  - routing code 34
  - user ID 30
- defining
  - CICS resources for WebSphere MQ 63
  - subsystems 10
  - WebSphere MQ to IMS 74
- dependent region user ID, IMS 173
- destination resolution exit
  - sample 225
  - specifying name 31
  - writing 223
- device type for logs (UNIT) 44
- DFSYDRU0 sample module 223
- DFSYPRX0 223
- display
  - system settings 26
- distributed queuing
  - defining the data sets 16
  - installation verification program 58
  - LE runtime library 10
  - MCA user ID 173
  - RESLEVEL 167
  - SCEERUN 10
  - security 20

- distributed queuing (*continued*)
  - security checking 167
  - security considerations 192
  - setting the CCSID 32
  - supplied samples 23
  - testing customization 58
  - user IDs used 173
- distributed queuing using CICS ISC SYSTEM.\* queue security 142
- DRU exit
  - sample 225
  - specifying name 31
  - writing 223
- druexit name, specifying for OTMA 31
- dual BSDS (TWOBSDS) 38
- dual logging
  - specifying for active log (TWOACTV) 37
  - specifying for archive log (TWOARCH) 38
- dump formatting member 49
- dynamic calls, IMS 74
- dynamic queues
  - close options 140
  - security 139

## E

- early code
  - library 8
- English language feature 4
- example
  - output from CSQ4IVPX 60
  - queue manager security scenario 203
  - queue-sharing group security scenario 210
  - SMF statistics records 91
  - SMP/E JOBCARD job step 220
  - SMP/E LINK CALLLIBS job 221
  - SMP/E REPORT CALLLIBS job 221
- EXEC CICS LINK
  - COMMAREA option 68
  - linking to the CICS adapter 68
- exit program
  - CICS adapter 68
  - number of TCBS 30
  - time allowed per invocation 30
- EXITLIM parameter of CSQ6SYSP 30
- EXITTCB parameter of CSQ6SYSP 30
- eye-catcher strings, statistics trace 90

## F

- features, installable 1
- format
  - type 115 SMF records 89
  - type 116 SMF records 104
- formatting dumps 49
- function keys, updating 49

## G

- global accounting interval 86
- global trace
  - initial setting 34
  - start automatically 34

- GMQADMIN security class 123, 127
- GMQNLIST security class 123
- GMQPROC security class 123
- GMQQUEUE security class 123
- GMXADMIN security class 127
- group class, security 123
- group name, specifying for OTMA 31
- GRPLIST system initialization
  - parameter 66

## H

- header
  - SMF type 115 record 89
  - SMF type 116 record 104

## I

- ICHRIN03, started-task procedure table 185
- IDBACK parameter of CSQ6SYSP 30
- IDFORE parameter of CSQ6SYSP 30
- IEFSSNss SYS1.PARMLIB member 11
- IFASMFDP reporting program for SMF 86
- IGAUT attribute 177
- IGQUSER attribute 177
- importance
  - workload management 17
- IMS
  - connection security 134
  - dynamic call stub, linking 74
  - OPERCMD5 security class 199
  - second user ID, determining 173
  - security 199
  - user IDs, security checking 172
- IMS adapter
  - CSQQDEFV, subsystem definition table 77
  - CSQQDEFX, macro 77, 78
  - defining WebSphere MQ to it 77
  - installing 73
  - language interface token (LIT) 78
  - SSM EXEC parameter 76
  - SSM specification options 76
  - subsystem member entry in IMS.PROCLIB 74
- IMS bridge
  - age, specifying for OTMA 31
  - application access control 200
  - connection security 200
  - customizing 79
  - druexit name, specifying for OTMA 31
  - group name, specifying for OTMA 31
  - member name, specifying for OTMA 31
  - OTMA parameters 31
  - persistent messages 226
  - RACF profiles 200
  - security 199
  - security checking 202
  - storage class 80
  - suppressing console messages 49
  - Tpipe name 32

- IMS connection
  - RESLEVEL 166
  - second user ID 167, 172
  - security checking 166
- IMS.PROCLIB library 74
- INBUFF parameter of CSQ6LOGP 36
- incorrect access 191
- information messages, suppressing 49
- initialization input data sets
  - customizing 21
  - formats 22
- INITPARM system initialization
  - parameter 65
- input buffer size (INBUFF) 36
- installable features 1
- installation
  - Batch/TSO adapter 46
  - bootstrap data set 24
  - CICS adapter 63
  - creating the logging environment 26
  - CSQ6ARVP macro 38
  - CSQ6LOGP macro 35
  - CSQ6SYSP macro 28
  - defining page sets 25
  - IMS adapter 73
  - log data set 24
  - operations and control panels 47
  - security tasks 184
  - updating the ISPF menu 48
  - using CSQ6LOGP macro 35
- installation verification program (IVP)
  - distributed queuing 58
  - queue manager 53
- installing
  - CICS adapter 63
  - IMS adapter 73
- INTERVAL attribute of ALTER
  - SECURITY 22, 180
- intra-group queuing
  - RESLEVEL 168
  - security checking 168
  - user IDs for security checking 177
- investigating performance 88
- IPCS job, formatting dumps 49
- IPCS list, updating 49
- IPCS VERBEXIT 49
- IRC and the CICS adapter 65
- ISPF
  - effect of split screen 30
  - installing panels permanently 48
  - menu, updating 48
  - operations and control panels, setting up 47
- ISPLLIB concatenation 48
- ISPMLIB concatenation 48
- ISPPLIB concatenation 48
- ISPTLIB concatenation 48
- IVP (installation verification program)
  - distributed queuing 58
  - queue manager 53
- IXCMIAPU, sample statements 19

## J

- Japanese language feature 4
- Japanese language letter 1
- Java Support feature 2

## L

- Language Environment, applying
  - service 219
- language interface token (LIT) 75
- language letter 1
- language, national 4
- layout
  - type 115 SMF records 89
  - type 116 SMF records 104
- libraries, after installation 2
- LINK CALLLIBS 221
- link list, updating 8
- LIT (language interface token) 75
- load libraries, APF authorization of 7
- lock manager statistics 98
- log data sets
  - creating 24
  - single or dual 35
- log initialization parameters, setting 28
- log manager statistics 94
- log records, number between
  - checkpoints 30
- logging parameters
  - default 35
  - setting 35
- logging, single and dual 35
- LOGLOAD parameter of CSQ6SYSP 30
- logs
  - number of buffers per write 38
- LU 6.2
  - channels, user IDs used 174
  - example security scenario 203

## M

- macros
  - CSQ6ARVP 26, 38
  - CSQ6LOGP 26, 35
  - CSQ6SYSP 26, 28
- MAXARCH parameter of
  - CSQ6LOGP 36
- MAXCHL
  - setting 45
- MAXRTU parameter of CSQ6LOGP 37
- MCA user ID, security 173, 192
- member class, security 123
- member name, specifying for OTMA 31
- message manager
  - accounting 109
  - statistics 95
- message routing code (ROUTCDE) 33
- messages
  - information, suppressing 49
  - suppressing 49
  - undelivered, security 141
  - violation, security 190
- MGCRE 171
- migration, testing 53
- model queues, security 139
- monitoring
  - DISPLAY commands 81
  - performance 81
  - resource usage 81
  - tools 81
- mounting, archive log (ARCWTOR) 40
- MQADMIN security class 123, 127

- MQCLOSE options, security 143
- MQCMD5 security class 123
- MQCONN security class 123, 133
- MQGET security 138
- MQI client security 175
- MQNLIST security class 123
- MQOPEN
  - security options 143
  - user IDs used 171
- MQOPEN/MQPUT1 options,
  - security 136
- MQPROC security class 123
- MQPUT security 138
- MQPUT1
  - security options 143
  - user IDs used 171
- MQQUEUE security class 123
- MQSC commands
  - resource security profiles 161
  - security 152
- MXADMIN security class 127

## N

- namelist security
  - implementing 216
  - profile 146
- naming a CF structure 19
- national language feature 1
- network-received user ID, security 192
- NOWAIT option on QINDXBLD 32

## O

- objects
  - supplied samples 23
- OFFLOAD parameter of CSQ6LOGP 37
- OpenEdition sockets
  - security 20
- operations and control panels
  - changing function keys 49
  - installing permanently 48
  - libraries 48
  - RESLEVEL 165
  - security 170
  - security checking 165
  - setting up 47
  - SYSTEM.\* queue security 142
- OPERCMD5 class 199
- OTMA
  - DRU exit sample 225
  - pre-routing exit sample 223
- OTMACON parameter of CSQ6SYSP 31
- OUTBUFF parameter of CSQ6LOGP 37
- output buffer, logs (OUTBUFF) 37

## P

- page sets
  - adding 25
  - defining 25
  - dynamic expansion 25
  - initialization input data sets 22
  - sample 25
- panels
  - changing function keys 49

- panels (*continued*)
  - installing 48
  - installing permanently 48
  - libraries 48
  - RESLEVEL 165
  - security 170
  - security checking 165
  - setting up 47
  - SYSTEM.\* queue security 142
- parmlibs, updating 10
- PCF commands
  - security 155
- pending events, termination
  - notification 67
- performance
  - compacting archive log 42
  - DISPLAY commands 81
  - effect of WebSphere MQ trace 84
  - example SMF records 91
  - investigating individual tasks 88
  - monitoring 81
  - problems 87, 88
  - SMF trace 85
  - snapshots 81
  - symptoms of reduced 87
  - workload management 17
- Performance Reporter 87
- performance statistics 88
- PL/I, testing customization 61
- plans (DB2), customization 17
- PLTPI (program list table post initialization)
  - starting a connection 64
  - starting the CICS adapter 67
- PPT (program properties table)
  - example 10
  - updating 10
- pre-routing exit 223
- PRIQTY parameter of CSQ6ARVP 42
- problem determination
  - performance 87
  - security 190
- procedures
  - channel initiator 16
  - queue manager 14
- process security
  - implementing 216
  - profile 145
- proclibs 14
- profile, RACF 124
  - for alternate user security 147
  - for command resources 161
  - for command security 151
  - for connection security
    - batch connections 133
    - channel initiator 135
    - CICS 134
    - IMS 134
  - for context security 149
  - for namelists 146
  - for process security 145
  - for queue security 135
  - for topic security 158
  - RESLEVEL 163
  - switch 126
  - used to protect WebSphere MQ resources 132

- program autoinstall, CICS 64
- program list table (PLT) 64
- PROTECT parameter of CSQ6ARVP 43
- PUTAUT channel attribute 173

## Q

- Q5ST 98
- QINDEXBLD
  - NOWAIT option 32
  - WAIT option 32
- QINDEXBLD parameter of CSQ6SYSP 32
- QMCCSID (queue manager coded character set identifier) 32
- QMCCSID parameter of CSQ6SYSP 32
- QSGDATA (queue-sharing group data parameter) 32
- QSGDATA parameter of CSQ6SYSP 32
- queue manager
  - coded character set identifier (QMCCSID) 32
  - data-sharing group name 33
  - DB2 name 33
  - installation verification program 53
  - queue-sharing group name 32
  - testing 53
- queue security
  - alter attributes 135, 152
  - implementing 216
  - profiles 135
- queue-level accounting 110
- queue-sharing group data parameter (QSGDATA) 32
- queue-sharing group security
  - alternate user 147
  - command 151
  - command resource 161
  - connection 133
  - context 149
  - incorrect access 191
  - intra-group queuing 177
  - namelist 146
  - process 145
  - queue 135
  - RESLEVEL profile 163
  - violation messages 191
- queue-sharing groups
  - adding WebSphere MQ entries to the data-sharing group 25
  - CF structures required 19
  - controlling security 128
  - customizing DB2 17
  - customizing the Coupling Facility 19
  - data-sharing group name 33
  - DB2 name 33
  - example security scenario 210
  - name 32
  - overriding security settings 127
  - QSGDATA parameter 32
  - testing 57
- queues
  - alter attributes, security 135, 152
  - implementing security 216
  - security 135
  - supplied samples 23
- QUIESCE parameter of CSQ6ARVP 44
- QWHCCV 108

## R

- RACF
  - authority, dead-letter queue 142
  - authorization
    - ICHRIN03 185
    - STARTED class 185
    - started-task procedure table 185
    - to WebSphere MQ data sets 185
  - commands for CSQ4IVP1 54, 59
  - profiles 124
  - security classes 123
- RACF audit records written during connection processing 33
- RACF profiles, IMS bridge 200
- RDO (resource definition online) 63
- record layouts
  - CSQDQ5ST 98
- reduced performance, symptoms of 87
- REFRESH SECURITY command 132, 180
  - for SSL 183
- refreshing
  - security 180
- region error options (REO) 75
- remote queues
  - command resource checking 162
  - security 140
- REO (region error options) 75
- REPORT CALLLIBS 220
- RESAUDIT parameter of CSQ6SYSP 33
- RESLEVEL
  - auditing 163, 189
  - checking CICS user IDs 165
  - distributed queuing 167
  - implementing 217
  - IMS connection 166
  - intra-group queuing 168
  - usage notes 163
  - user IDs associated with 163
  - using 163
- resource definition online (RDO) 63
- Resource Measurement Facility (RMF) 87
- Resource Recovery Services (RRS)
  - applying service 219
- resource security
  - alias queues 162
  - API 143
  - commands 161
  - remote queues 162
- resource-level security checks
  - CICS adapter 195
  - switch profiles 130
- restricting access using alias queues 138, 141
- retention period, archive logs (ARCRETN) 40
- RMF (Resource Measurement Facility) 87
- ROUTCODE parameter of CSQ6SYSP 33
- route codes, archive log (ARCWRTC) 40
- routing code, message (ROUTCODE) 33
- RRS (Resource Recovery Services)
  - applying service 219
- RRS adapter, installing 46
- RVERIFY SECURITY command 179

## S

### sample

- CSQINP1 22
- CSQINP2 23
- CSQINPV 22
- CSQINPX 23
- data set members 3
- defining page sets 25
- destination resolution exit 225
- IXCMIAPU statements 19
- linking the IMS dynamic call stub 74
- OTMA pre-routing exit 223
- output from CSQ4IVPX 60
- queue manager security scenario 203
- queue-sharing group security scenario 210
- SMF accounting record 109
- SMF type 116 record
  - subtype 1 120
  - subtype 2 120
- SMP/E JOBCARD job step 220
- SMP/E LINK CALLLIBS job 221
- SMP/E REPORT CALLLIBS job 221
- started task procedure 15
- SCSQxxxx contents 2
- second user ID, IMS connection 167, 172
- SECQTY parameter of CSQ6ARVP 44
- Secure Sockets Layer (SSL),
  - configuration 186
- security
  - activating 126
  - API quick-reference table 143
  - archive log 43
  - auditing considerations 188
  - automating starting of CKTI 196
  - batch connections 133
  - blank user IDs 178
  - channel 186, 193
  - channel initiator 192
  - CICS adapter 195
    - transactions 195
    - user IDs 196
  - CICS bridge 197
  - CKSG user IDs 197
  - CKTI 196
  - clustering 194
  - command summary table 152
  - connection 133, 170
  - customizing 190
  - data sets 185
  - default user ID 30, 170
  - displaying status 183
  - distributed queuing 192
  - example queue manager scenario 203
  - example queue-sharing group scenario 210
  - implementation checklist 215
  - IMS 199
  - incorrect access 191
  - installation tasks 19, 184
  - installation verification program
    - distributed queuing 59
    - queue manager 54
  - INTERVAL attribute 22, 180
  - intra-group queuing 177
  - management 178

### security (continued)

- MQCLOSE/MQOPEN/MQPUT1
  - options 143
- operations and control panels 170
- OTMA 199
- problem determination 190, 215
- profile, RESLEVEL 163
- propagating CKTI user IDs 197
- queue-sharing group 127
- queues
  - alias 138, 162
  - dead-letter 141
  - dynamic 140
  - model 139
  - profiles 135, 152
  - remote 140, 162
  - transmission 144, 150, 162
- refreshes
  - SSL 183
- refreshing 180
- setting checks off 126
- terminal user IDs 196
- TIMEOUT attribute 22, 180
- topics
  - profiles 158
- undelivered messages 141
- universal access (UACC) levels 178
- user ID timeouts 179
- user IDs 170, 193
- user IDs for MQI clients 175
- using RACF classes 123
- utilities 171

### security switches

- example 131
- valid combinations 129

### self-defining section

- SMF type 115 records 90
- SMF type 116 records 105

### service class

- workload management 17

### service considerations 219

### SETSSI command 11

### shared queues

- adding WebSphere MQ entries to the data-sharing group 25
- CF structures required 19
- customizing DB2 17
- customizing the Coupling Facility 19
- data-sharing group name 33
- DB2 name 33
- example security scenario 210
- profiles for security 135
- QSGDATA parameter 32
- queue-sharing group name 32
- security 135
- testing 57

### Simplified Chinese

- language feature 4
- language letter 1

### single BSDS (TWOBSDS) 38

### single logging

- specifying for active log (TWOACTV) 37
- specifying for archive log (TWOARCH) 38

### SIT (system initialization table)

- connection ID 67

### SIT (system initialization table)

- (continued)
  - GRPLIST parameter 66
  - INITPARM parameter 65
  - PLTPI parameter 66
- SMF (System Management Facility)
  - accounting record sample 109
  - buffers 86
  - common header 107
  - CSQ6SYSP, specifying parameters 28
  - gathering (STATIME) 34
  - introduction 85
  - processing type 115 records 93
  - processing type 116 records 107
  - recording trace data for 85
  - reporting data in (IFASMFDP) 86
  - self-defining section
    - type 115 90
    - type 116 records 105
  - starting automatically (SMFSTAT) 34
  - statistics records example 91
  - type 115
    - header 89
    - record layout 89
    - record subtypes 89
    - self-defining section 90
  - type 116
    - header 104
    - record layout 104
- SMF global accounting interval 86
- SMFACCT parameter of CSQ6SYSP 34
- SMFSTAT parameter of CSQ6SYSP 34
- SNAP dump, Batch/TSO adapter 47
- snapshots, performance 81
- sockets
  - security 20
- space allocation
  - archive logs, block size (BLKSIZE) 41
  - logs, primary (PRIQTY) 42
  - logs, secondary (SECQTY) 44
  - units, logs (ALCUNIT) 39
- SPT (started-task procedure table) 185
- SSL
  - refreshing security 183
- SSL (Secure Sockets Layer),
  - configuration 186
- SSM (subsystem member)
  - entry in IMS.PROCLIB 74
  - EXEC parameter 76
  - specification options 76
- START TRACE command 83
- STARTED RACF class
  - authorization to data sets 185
  - authorizing procedures 185
- started task procedure
  - authorization 185
  - creating for channel initiator 16
  - creating for queue manager 14
  - security 14
- started task procedure, CSQ4MSRR 15
- started task procedure, CSQ4MSTR 15
- starting WebSphere MQ trace 83
- startup procedure, CSQ4MSRR 15
- startup procedure, CSQ4MSTR 15
- STATIME parameter of CSQ6SYSP 34
- statistics
  - buffer manager 95



- statistics (*continued*)
  - buffer pool 95
  - CICS adapter 82
  - Coupling Facility manager 103
  - data manager 95
  - DB2 manager 98
  - example SMF records 91
  - eye-catcher strings 90
  - gathering time interval 34
  - lock manager 98
  - log manager 94
  - message manager 95
  - security 189
  - starting automatically 34
  - storage manager 93
- STOP TRACE command 83
- storage class
  - IMS bridge 80
- storage group (DB2), drop 18
- storage manager statistics 93
- storage requirement, trace table 35
- structure layout
  - WQ 113
  - WQSTAT 113
  - WTAS 111
  - WTICDORI 119
  - WTID 118
- stub programs
  - CSQCSTUB for CICS adapter 66
- subsystem name table, updating 11
- subsystem security 127
- suppressing information messages 49
- switches, security 126
- SYS1.PARMLIB members
  - IEFSSNss 11
- SYS1.PARMLIB, updating 21
- SYSEXEC concatenation 48
- sysplex
  - command prefix string (CPF) 13
  - scope 13
- SYSPROC, concatenation 48
- system monitoring, DISPLAY commands 81
- system parameters
  - displaying settings 26
  - invoking 27
  - setting 28
  - tailoring 26
  - z/OS 7
- system security 185
- SYSTEM.\* queues
  - security 142

## T

- table space (DB2), drop 18
- tables and index (DB2), drop 18
- tailoring
  - channel initiator parameters 45
- TCB, number for exit programs 30
- TCP/IP
  - example security scenario 203
- TCP/IP channels, user IDs used 173
- terminal user IDs, CICS adapter 196
- testing your queue manager 53
- thlqual.SCSQxxxx, contents 2
- thread cross reference data 108

- thread-level accounting 110
- threads
  - ID for CICS transactions 67
- time stamp, archive log (TSTAMP) 44
- TIMEOUT security attribute 22, 180
- topic
  - security 158
- topic security
  - implementing 216
- Tpipe, name 32
- trace
  - controlling WebSphere MQ 83
  - effect on performance 84
  - specifying destinations 83
  - starting automatically (TRACSTR) 34
  - trace table size (TRACTBL) 35
- tracing parameters, setting 28
- TRACSTR parameter of CSQ6SYSP 34
- TRACTBL parameter of CSQ6SYSP 35
- TSO
  - formatting dumps 49
  - improving application portability 47
- TSTAMP parameter of CSQ6ARVP 44
- tuning the CICS bridge 71
- tuning WebSphere MQ 81
- TWOACTV parameter of CSQ6LOGP 37
- TWOARCH parameter of CSQ6LOGP 38
- TWOBSDS parameter of CSQ6LOGP 38

## U

- undelivered messages, security 141
- UNIT parameter of CSQ6ARVP 44
- UNIT2 parameter of CSQ6ARVP 45
- universal access (UACC) levels 178
- updating
  - CSQINP2 66
  - z/OS link list 8
  - z/OS parmlibs 10
  - z/OS subsystem name table 11
- upgrade considerations 219
- US English language letter 1
- user exits 68
- user ID
  - implementing timeouts 217
  - intra-group queuing 177
  - maximum age in OTMA 31
  - MQI clients 175
- user ID security
  - batch connections
    - checks 170
    - introduction 133
    - RESLEVEL 163, 164
  - blank 178
  - channel initiator, connection 135
  - checking 171
  - CICS
    - connection 134, 170
    - RESLEVEL 163
  - CICS adapter
    - transactions 195
  - CICS connection
    - address space 172
    - connection 165
    - RESLEVEL 165
    - task 172

- user ID security (*continued*)
  - CICS connection (*continued*)
    - transactions 172
  - CKTI 196
  - CSQUTIL 165
  - default 30
  - distributed queuing
    - connection 167
    - RESLEVEL 167, 168
  - IMS
    - connection 134, 170
    - RESLEVEL 163
  - IMS connection
    - address space 172
    - connection 166
    - RESLEVEL 166
    - second user ID 167, 172
  - intra-group queuing, RESLEVEL 168
  - number checked 163
  - operations and control panels 165
  - RESLEVEL profile 163
  - reverification 179
  - timeouts 179
- Utoken 171

## V

- velocity goal
  - workload management 17
- VERBEXIT, IPCS 49
- violation messages, security 190
- VTAM node name, connection ID in system initialization tables 67

## W

- WAIT, option on QINDXBLD 32
- WebSphere MQ commands
  - DISPLAY SECURITY 132
  - REFRESH SECURITY 132
  - security profiles 151
  - START TRACE 83
  - STOP TRACE 83
- WebSphere MQ utility program (CSQUTIL), security 171
- WLMTIME parameter of CSQ6SYSP 35
- WLMTIMU parameter of CSQ6SYSP 35
- workload management, service class 17
- workload manager, queue scan interval 35
- workload manager, queue scan interval units 35
- WQ layout 113
- WQSTAT layout 113
- WRTHRS parameter of CSQ6LOGP 38
- WTAS layout 111
- WTICDORI layout 119
- WTID layout 118

## X

- XCF
  - group name, specifying for OTMA 31
  - member name, specifying for OTMA 31

## Z

### z/OS

- APF authorized libraries 7
- link list, updating 8
- parmlibs, updating 10
- program properties table,  
updating 10
- SNAP dump 47
- subsystem name table, updating 11
- system parameters 7

---

## Sending your comments to IBM

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

**To make comments about the functions of IBM products or systems, talk to your IBM representative or to your IBM authorized remarketer.**

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail, to this address:

User Technologies Department (MP095)  
IBM United Kingdom Laboratories  
Hursley Park  
WINCHESTER,  
Hampshire  
SO21 2JN  
United Kingdom

- By fax:
  - From outside the U.K., after your international access code use 44-1962-816151
  - From within the U.K., use 01962-816151
- Electronically, use the appropriate network ID:
  - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
  - IBMLink™: HURSLEY(IDRCF)
  - Internet: idrcf@hursley.ibm.com

Whichever method you use, ensure that you include:

- The publication title and order number
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.







SC34-6927-00



Spine information:



WebSphere MQ for z/OS

System Setup Guide

Version 7.0