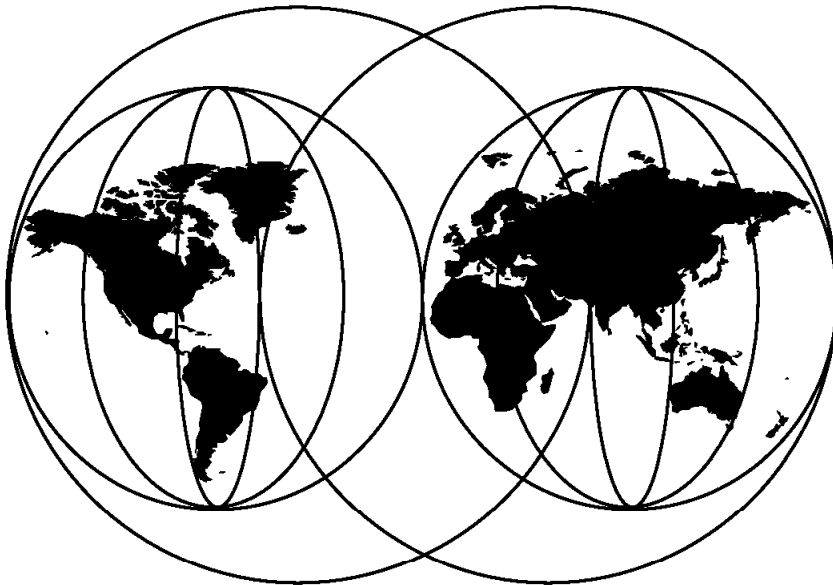




MQSeries

Backup and Recovery

*Dieter Wackerow, Juan Arevalo, Fernando Garcia, Ericka Hernandez,
Marc Luong, Hubert T. Lye, Hector Sanchez, Paul Sehorne*



International Technical Support Organization

<http://www.redbooks.ibm.com>

This book was printed at 240 dpi (dots per inch). The final production redbook with the RED cover will be printed at 1200 dpi and will provide superior graphics resolution. Please see "How to Get ITSO Redbooks" at the back of this book for ordering instructions.

SG24-5222-00



SG24-5222-00

International Technical Support Organization

MQSeries
Backup and Recovery

September 1998

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix D, "Special Notices" on page 141.

First Edition (September 1998)

This edition applies to the following products:

- MQSeries for AIX Version 5
- MQSeries for AS/400 Version 4 Release 2
- MQSeries for MVS/ESA Version 1 Release 2
- MQSeries for OS/2 Warp Version 5
- MQSeries for Windows NT Version 5

Comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 678
P.O. Box 12195
Research Triangle Park, NC 27709-2195

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1998. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	vii
Preface	ix
The Team That Wrote This Redbook	ix
Comments Welcome	xi
Chapter 1. Introduction to Backup and Recovery	1
1.1 Terminology Used in This Book	1
1.2 Restore and Recovery	1
1.3 Circular versus Linear Logging	2
1.3.1 UNIX, OS/2, and Windows NT	3
1.3.2 AS/400	3
1.3.3 MVS	3
1.4 Protecting Your MQSeries Object Definitions	4
1.4.1 UNIX, OS/2, and Windows NT	4
1.4.2 AS/400	4
1.4.3 MVS	4
1.5 MQSeries Informational Messages Regarding Logs	5
1.5.1 Informational Message Numbers	5
1.5.2 Location of Informational Messages Regarding Logs	5
1.5.3 Client Messages, More Error Messages, and FDC Files	6
1.5.4 Other Messages Not Specific to an Individual Queue Manager	6
1.6 Manually Recording/Recovering Media Images	6
1.6.1 UNIX, OS/2, Windows NT	6
1.6.2 AS/400	7
1.6.3 MVS	7
1.7 Queue Manager Resources to Protect/Backup	7
1.7.1 UNIX, OS/2, Windows NT	7
1.7.2 AS/400	7
1.7.3 MVS	7
Chapter 2. MQSeries for MVS/ESA	9
2.1 Background Information	9
2.1.1 Data Sets Used by MQSeries for MVS/ESA	9
2.1.2 Backup	12
2.1.3 Restart and Recovery	13
2.2 Tips for Backup and Recovery	14
2.2.1 Periodically Take Backup Copies of Page Sets	14
2.2.2 Periodically Take Backup Copies of Queue Manager Objects	15
2.2.3 Use Dual Logging for Your Active Log, Archive Log, and BSDS	15
2.2.4 Keep Archive Logs You Might Need	15

2.2.5 Retain the DD Name or Page Set Association	16
2.3 Backup and Recovery Procedures	16
2.3.2 Backing Up the Page Sets	17
2.3.3 Recovering a Page Set	19
2.3.4 Backing Up Your Object Definitions	21
2.3.5 Backing Up and Restoring Queues	22
2.4 Recovery Scenarios	22
2.4.1 Recovering Lost or Corrupted Logs	23
2.4.2 Recovering a Damaged Page Set	26
2.4.3 Recovering a Lost or Damaged BSDS	26
2.4.4 Disaster Recovery	28
2.4.5 Channel Re-synchronization	31
Chapter 3. MQSeries for AS/400	37
3.1 Journaling	37
3.1.1 OS/400 Journaling	37
3.1.2 MQSeries and OS/400 Journaling	39
3.1.3 Journal Management	39
3.1.4 Tips for MQSeries Journaling	43
3.2 MQSeries for AS/400 Backup Procedures	47
3.2.1 Where Things Are Located	47
3.2.2 Backing Up MQSeries for OS/400 Objects	47
3.3 MQSeries for AS/400 Recovery Scenarios	49
3.3.1 Scenario 1: Disk Crash	50
3.3.2 Scenario 2: QMQMDATA Contains Only Journal Receivers	53
3.3.3 Scenario 3: Queue Manager Is Damaged	54
3.3.4 Scenario 4: Queue Manager Is Deleted	56
3.3.5 Scenario 5: System Failure, No Data Loss	59
3.3.6 Scenario 6: Restoring MQSeries into Another AS/400 System	60
3.3.7 Scenario 7: Object Damage	64
3.4 Useful Hints for MQSeries Backup and Recovery	64
3.4.1 RCRMQMIMG	64
3.4.2 Size of Journal Receivers	66
3.4.3 Sign Off and Sign On before STRMQM	66
3.4.4 Quiescing the AS/400	66
3.4.5 Shared Memory	66
3.4.6 Journal Receiver Chain	67
Chapter 4. MQSeries for UNIX Systems	71
4.1 Parameters for Creating the Queue Manager	72
4.2 MQSeries Logging and Logs	74
4.2.1 What Are the Necessary Logs for Recovery?	74
4.3 What to Backup	74
4.3.1 Directories and Files to Back Up	74

4.3.2	System Checkpoint Files to Back Up	75
4.3.3	Queue Manager and Queues to Back Up	76
4.4	How to Re-synchronize the Channel	76
4.4.1	Verify Channel and Transmission Queue	77
4.4.2	Backout/Commit and Resolve Channel	79
4.4.3	Reset the Sequence Number in the Channel	79
4.4.4	Restart the Channel	80
4.5	How to Re-synchronize the Channel with Media Command	80
4.6	How to Stop the Queue Manager	80
4.6.1	Normal Shutdown	80
4.6.2	Immediate Shutdown	81
4.6.3	Preemptive Shutdown	81
4.7	MQSeries for UNIX Systems Backup Procedures	81
4.7.1	Two Backup Options	81
4.7.2	How to Back Up All Queue Managers (System Backup)	83
4.7.3	How to Back Up One Queue Manager (System Backup)	85
4.7.4	How to Back Up One Queue Manager (Media Backup)	86
4.7.5	How to Back Up a Synchronization File (Media Backup)	87
4.7.6	How to Back Up a Queue (Media Backup)	88
4.8	MQSeries for UNIX Systems Recovery Scenarios	88
4.8.1	Reasons for a Recovery	88
4.8.2	Two Recovery Options	89
4.8.3	How to Restore a Queue Manager (System Recovery)	90
4.8.4	How to Recover a Queue Manager (Media Recovery)	93
4.8.5	How to Recover the Synchronization File (Media Recovery)	95
4.8.6	How to Recover a Queue (Media Recovery)	96
Chapter 5.	MQSeries for Windows NT and MQSeries for OS/2	99
5.1	Parameters for Creating the Queue Manager	100
5.2	MQSeries Logging and Logs	102
5.2.1	Which Are the Necessary Logs for Recovery?	103
5.2.2	Log Problems and Queues Containing Very Old Messages	105
5.3	What to Back Up	107
5.3.1	Directories and Files to Back Up	107
5.3.2	System Checkpoint Files to Back Up	108
5.3.3	Queue Manager and Queues to Back Up	108
5.4	How to Re-synchronize the Channel	108
5.4.1	Verify Channel and Transmission Queue	110
5.4.2	Backout/Commit and Resolve Channel	111
5.4.3	Reset the Sequence Number in the Channel	112
5.4.4	Restart the Channel	113
5.5	How to Re-synchronize the Channel with Media Command	113
5.6	How to Stop the Queue Manager	113
5.6.1	Normal Shutdown	113

5.6.2	Immediate Shutdown	113
5.6.3	Preemptive Shutdown	114
5.6.4	Using the Windows NT Task Manager	114
5.7	MQSeries for Windows NT and OS/2 Backup Procedures	116
5.7.1	Two Backup Options	116
5.7.2	How to Back Up All Queue Managers (System Backup)	117
5.7.3	How to Back Up One Queue Manager (System Backup)	119
5.7.4	How to Back Up One Queue Manager (Media Backup)	122
5.7.5	How to Back Up the Synchronization File (Media Backup)	123
5.7.6	How to Back Up a Queue (Media Backup)	123
5.8	MQSeries for Windows NT and OS/2 Recovery Scenarios	124
5.8.1	Two Recovery Options	124
5.8.2	How to Restore a Queue Manager (System Recovery)	125
5.8.3	How to Recover a Queue Manager (Media Recovery)	129
5.8.4	How to Recover the Synchronization File (Media Recovery)	131
5.8.5	How to Recover a Queue (Media Recovery)	131
Appendix A. Guide to Unplanned Outages on the AS/400 (DASD Loss)		135
Appendix B. AS/400 DASD Storage Management: ASPs (Auxillary Storage Pools)		137
B.1	Additional Disk Protection for Large AS/400s	137
B.2	Partial DASD Protection and User ASPs	138
Appendix C. RAID		139
Appendix D. Special Notices		141
Appendix E. Related Publications		145
E.1	International Technical Support Organization Publications	145
E.2	Redbooks on CD-ROMs	145
E.3	Other Publications	145
How to Get ITSO Redbooks		147
	How IBM Employees Can Get ITSO Redbooks	147
	How Customers Can Get ITSO Redbooks	148
	IBM Redbook Order Form	150
List of Abbreviations		151
Index		153
ITSO Redbook Evaluation		155

Figures

1.	JCL to Invoke the Print Log Map Utility	16
2.	JCL to Define New Page Sets	18
3.	JCL to Back Up Page Sets	19
4.	JCL to Recover a Page Set	20
5.	JCL Using the MAKEDEF Command of CSQUTIL	21
6.	JCL to Take a Backup of a Queue	22
7.	JCL to Restore a Queue	22
8.	JCL to Re-define and Restore an Active Log	24
9.	JCL Using the RESETPAGE Function of CSQUTIL	25
10.	JCL Using the AMS VERIFY Function	25
11.	JCL to Restore from a Backup Page Set	26
12.	JCL Used to Define a New BSDS	27
13.	JCL Used to Recover a BSDS	28
14.	List Channels Output	33
15.	Illustration of OS/400 Journaling Concept	38
16.	MQSeries for AS/400 Journal and Receiver Management	40
17.	MQSeries for AS/400 Startup Journal Information AMQ7460	41
18.	MQSeries for AS/400 Media Recovery Journal Information AMQ7462	42
19.	MQSeries for AS/400 Display Current Journal Receiver	43
20.	MQSeries for AS/400 Display Journal Receiver Attributes	44
21.	MQSeries for AS/400 Changing to System Managed Journal Receivers	46
22.	MQSeries for AS/400 Damaged Object	65
23.	MQSeries for AS/400 Display Journal Receiver Chain	67
24.	Work with Journal Attributes Menu	68
25.	Display the Journal Receiver Attributes	68
26.	Display the Journal Receiver Attributes for AMQA000042	69
27.	Message AMQ7467	104
28.	Message AMQ7468	105
29.	Dealing with Old Messages	106
30.	Logs Required for Startup and Media Recovery	107
31.	Windows NT Task Manager	115
32.	Windows NT Services	115

Preface

Backup and recovery is very important in every production environment. It is essential to have well-defined procedures that ensure a smooth recovery in case any of the components in the system fails, for example, disk failures, electrical black-outs, or accidentally deleted files.

This redbook helps you define backup, restore and recovery procedures for MQSeries objects on MVS/ESA, AS/400, UNIX, OS/2 and Windows NT systems. Such procedures enable system operators to restore lost data, restart the system, and make applications available to end users again as fast as possible.

This redbook contains concise facts about logs and archives, the data that has to be protected, where to save it and how long to keep it. Practical sample scenarios for backing up essential data, recovering destroyed objects, and restoring a corrupted queue manager are included.

The scenarios in this book have been validated using MQSeries for Windows NT Version 5, MQSeries for AIX Version 5, MQSeries for AS/400 Version 4 Release 2, and MQSeries for MVS/ESA Version 1 Release 2.

The first chapter contains an introduction to backup and recovery including topics such as circular and linear logging, messages and message logs, recording and recovering media images, and what resources to back up. The other chapters contain guidelines and scenarios specific to the platforms MVS/ESA, AS/400, AIX, OS/2 and Windows NT.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the Systems Management and Networking ITSO Center, Raleigh.

Juan Arevalo Carranza is the project leader for MQSeries applications at Elektra S.A. de C.V. in Mexico. He has eight years of experience with AS/400 systems. He holds a degree in Information Technology from the Instituto Politecnico Nacional in Mexico City. His areas of expertise include Programming in RPG and C++ on Windows NT and AS/400 platforms.

Fernando Garcia is an I/T Specialist in Product Support Services, IBM Germany. He has 20 years of experience with IBM and has provided 2nd level and product support for MQSeries for 5 years. His main areas of expertise are MVS, CICS/ESA, CICSplex/SM and Cobol. He plays an active role as an MQSeries specialist for services and field support operations.

Ericka Hernandez Lara is an MQSeries Certified Specialist in Mexico. She is experienced in installing and administering MQSeries on several platforms, and also in programming MQSeries applications. She joined IBM two years ago and holds a degree in Electronic and Communication Engineering from the Instituto Politecnico Nacional in Mexico City.

Marc Luong is an IBM MQSeries Specialist in the Software Group in IBM France. For six years he worked very closely with the Hursley laboratory on MQSeries multi-platform products. He has 23 years experience in database data communication (DB/DC), IMS, DB2, CA-Datcom, CICS and MQSeries. Marc joined IBM in 1986. He holds degrees in Computer and Electronic Engineering from Ecole Superieure d'Informatique and the University of Paris.

Hubert T. K. Lye is a Software Services Specialist with the IBM Support Centre in IBM Australia. He has been with IBM for 9 years. Hubert holds a B.Sc degree in Combined Studies (Economics and Computing Science) from Manchester Polytechnic, UK. He contributed to a redbook on AS/400 availability and recovery planning and is actively providing support for MQSeries in IBM Australia.

Hector Sanchez Palacios is a Certified Specialist for MQSeries in Mexico. He has many years of experience in MQSeries on several UNIX platforms, Windows NT and AS/400. He has worked at IBM for three years and holds a degree in Electronic Engineering from the Universidad Autonoma Metropolitana in Mexico City.

Paul F. Sehorne is an IBM Certified Solutions Expert - MQSeries. He is team leader of the MQSeries Advanced Technical Support Team of the IBM Dallas Systems Center. He does technical marketing support of MQSeries including consulting, architecture design, and application design. He has helped many customers get off to a SmoothStart with MQSeries. He has contributed to an MQSeries whitepaper and now a Redbook and is a speaker at the 1998 MQSeries Technical Conference.

Dieter Wackerow is the MQSeries expert at the Systems Management and Networking ITSO Center, Raleigh. His areas of expertise include application design and development for various industries, performance evaluations, capacity planning, and modelling of computer systems and networks. He also wrote a simulator for banking hardware and software. He teaches classes and writes on performance issues, application development and about MQSeries.

Thanks to the following people for their invaluable contributions to this project:

Satish Babu
IBM India

Klaus Haaber-Bernth
IBM Sweden

Jane Porter
IBM Hursley, England

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “ITSO Redbook Evaluation” on page 155 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

For Internet users <http://www.redbooks.ibm.com/>

For IBM Intranet users <http://w3.itso.ibm.com/>

- Send us a note at the following address:

redbook@us.ibm.com

Chapter 1. Introduction to Backup and Recovery

In a production environment, it is very important to have a well-defined strategy to recover or restore the MQSeries environment and data in case of a disaster. Think about situations such as electrical failures, disk failures or accidentally deleted files. All these can result in situations where there are system failures but no data loss, system failures and total data loss and also damaged objects.

In this chapter, you will find some general but important information about backup, restore and recovery. The other chapters provide various platform-dependent backup and recovery scenarios you may use on MVS/ESA, AS/400, AIX, OS/2 and Windows NT.

1.1 Terminology Used in This Book

The following terms are very important to the discussions that follow. You should make sure you are familiar with the exact meaning they have in this context.

<i>Log</i>	A file where transactions (MQPUT, MQGET) are recorded.
<i>Queue</i>	A file where messages are stored.
<i>Archiving</i>	The act of "moving" log data no longer needed on disk to an offline media for the purpose of freeing disk space.
<i>Backing up</i>	The act of "copying" files from disk to an another location for the purpose of creating a point-in-time image of a system.
<i>Restore</i>	The act of copying a point-in-time image or some part of that image (a single log file, or page set) back to disk.
<i>Recovery</i>	The act of applying log records to the recoverable resources managed by a queue manager to arrive at a point of consistency.

1.2 Restore and Recovery

An archive (of transaction logs) may be used to recover damaged/lost objects by restoring the archive and applying the media images and transactions in the logs to the damaged queue manager objects. A backup can be used to restore a complete queue manager to the state that existed at the time the backup was taken. Use of an archive or a backup assumes that log files that are required to recover objects but that are not in the archive or backup are intact on disk. If the log files that have been created since a backup was taken exist they can be applied to the restored

point-in-time copy of the queue manager to bring the queue manager back to the state it was in when a failure occurred. If log files on disk are corrupted or otherwise unavailable a backup can only be used to restore a queue manager to a previous point in time.

If only a point-in-time restore can be done, that is, the log files created since the backup was taken cannot be applied to the point-in-time copy, purge messages from all your queues before using the queue manager. Any messages that were on queues when the backup was taken are likely to have been processed between the time the backup was taken and when the failure occurred. You probably do not want to process these messages again (unless they were deposits to your personal bank account).

On UNIX, OS/2, and Windows NT, a restore from backup will re-create the directory structure if done properly.

Non-persistent messages on a queue are always lost if the queue manager is restarted.

Persistent messages survive a restart after a normal or abnormal termination. This does not depend on the setting of the DEFPSIST (default persistence) attribute in the local queue definition.

The default persistence flag of a queue is only used to specify the persistence attribute for messages that use the default value for the queue. This is set in the message descriptor (MQMD) of the message.

It has no affect if a queue has been defined with default persistence yes or no. The queue always survives a restart.

1.3 Circular versus Linear Logging

Circular logging keeps data in a ring of log files. When the last log is full it overwrites the first one. Linear logging keeps the log data in a continuous sequence of files.

With circular logging, you only get stop/restart capability for a "healthy" queue manager. MQ objects are not written to circular logs. To restart a queue manager if objects have been damaged or lost, you also need media recovery. In order to perform media recovery, the queue manager *must* have written media images of MQ objects in the logs. MQSeries stores media images in logs only if you are using linear logging. Images are written to logs at the following times:

- When an object is created

- Process objects and non-local queues at shutdown
- Local queues when the queue becomes empty

We recommend that log files reside on mirrored or parity protected hard drives.

1.3.1 UNIX, OS/2, and Windows NT

For UNIX, OS/2, and Windows NT, logging type *must* be set at the time the queue manager is *created*. The default for the "crtmqm" (create queue manager) command is circular logging. Use the "-ll" parameter to specify linear logs.

1.3.2 AS/400

MQSeries for AS/400 uses the OS/400 file journaling. Media images are therefore written to journal receivers. There is no option for circular logging. For additional protection, we recommend that the journal receivers reside in a user ASP (Auxiliary Storage Pool). User ASP provides additional protection by eliminating the need to recover the entire system in the event of a failure in one ASP. When used without some form of DASD protection, recovery time is shortened by the time required to reload the failed disk. When used with DASD mirroring or RAID 5, it provides a much higher level of protection. For systems with highly available, high availability and continuous availability requirements, all DASD in ASPs should be protected with some form of DASD protection. Further information can be found in Appendix A, "Guide to Unplanned Outages on the AS/400 (DASD Loss)" on page 135 and Appendix B, "AS/400 DASD Storage Management: ASPs (Auxillary Storage Pools)" on page 137.

1.3.3 MVS

MQSeries for MVS/ESA uses its own logging system. The object definitions are available in a special type of MQSeries data set which is called page set zero. An MVS storage class maps one or more queues to a page set. Since page set zero is used to store all the object definitions required by the queue manager, it is essential that page set zero does not become full. It is best not to have object definitions and messages on the same page set. Therefore, we recommend that you do not define queues with storage classes that map to page set zero. If you do, you run the risk of filling page set zero if too many messages accumulate there.

MQSeries for MVS/ESA does archiving automatically and will recall logs from archive media automatically when needed. You can use the RESETPAGE function to generate a set of consistent page sets that can be used with a set of new (clean) boot strap data set (BSDS) and log data sets to start MQSeries. You would only have to do this if both copies of the log

have been lost or damaged for some reason, and you need to restart from backup page set copies (and accept the resulting loss of data from the time the copies were made). In this situation, you should use the RESETPAGE function on all the page sets of the affected queue manager. You must also create new BSDS and log data sets.

1.4 Protecting Your MQSeries Object Definitions

It is a good idea to keep MQSeries object definition files in a safe place. These are text files you create that contain the MQSeries definition statements that you used to create your MQSeries objects. Some third party tools keep object definitions in a database. If you are using one of these tools, it should not be necessary to keep a text file of MQ object definitions. If you create MQ objects interactively you will not have a permanent record from which those objects can be re-created if necessary (except for the media images in the logs). This is particularly important if you are using circular logs, since there are no media images in circular logs.

1.4.1 UNIX, OS/2, and Windows NT

We recommend that you only create MQ objects by putting the definitions in text files and passing these text files to "runmqsc". If you create object definitions by using "runmqsc" interactively it will be less convenient to re-create your objects.

1.4.2 AS/400

The AS/400 command STRMQMMQSC can accept a text file of object definitions as input. In the AS/400, MQSC commands are not interactive. The user can create MQ objects by coding them into a text member (script file) and then executing them through the AS/400 command STRMQMMQSC, for example,

```
STRMQMMQSC SCRMBR(some member) OPTION(*RUN)
```

. The results of this command will be in the spool file.

1.4.3 MVS

MQSeries uses page set zero (PSID=00) to store object definitions and other important information relevant to the queue manager subsystem. The objects are separate from the data on MVS. On MVS, you can create definition statements for your objects using the MAKEDEF parameter of the COMMAND function of CSQUTIL while the queue manager is running or by using the SDEFS function (V1.2) while the queue manager is stopped. This will create a text file that can be used as input to CSQUTIL to re-create the objects.

1.5 MQSeries Informational Messages Regarding Logs

There are two messages issued by MQSeries that help you manage the logs. One message indicates the oldest log that must remain on disk for normal operations of queue manager. The other indicates the oldest log file that is needed to restore a damaged object. Log files older than the oldest one that must remain on disk, back to and including the oldest one needed for media recovery, may be archived and deleted from disk.

1.5.1 Informational Message Numbers

1.5.1.1 UNIX, OS/2, and Windows NT

AMQ7467 Indicates the oldest log needed to restart a healthy queue manager.

AMQ7468 Indicates the oldest log needed to recover media images.

1.5.1.2 OS/400

AMQ7460 Startup recovery point.

AMQ7462 Oldest media recovery entry.

1.5.1.3 MVS/ESA

CSQI024I System restart RBA.

CSQI025I System restart RBA including offline page sets.

1.5.2 Location of Informational Messages Regarding Logs

The location of these messages vary by platform.

1.5.2.1 UNIX, OS/2, and Windows NT

Informational messages are written to AMQERR01.LOG in the queue manager's "error" directory. In the list below "xxxx" is the queue manager's, possibly mangled, name and "x:" is the drive letter. The queue manager name may be mangled in order to create a name that is valid for directories on the specific platform.

UNIX /var/mqm/qmgrs/xxxx/errors

OS/2, Windows NT x:\mqm\qmgrs\xxxx\errors

Messages are also written to the Windows NT Event Log.

Error messages issued by a queue manager are also written to the AMQERR01.LOG in the same directory.

1.5.2.2 AS/400

Messages are written to the system operator queue. Messages may be viewed by following these steps:

1. Execute DSPMGS.
2. Press F6=Display system operator messages.
3. Find the following two messages:
 - MQM media recovery journal information.
 - MQM startup journal information.
4. Use option 5 to display details.

1.5.2.3 MVS

Messages are posted to the system operator console.

1.5.3 Client Messages, More Error Messages, and FDC Files

Messages issued by client applications are written to:

UNIX /var/mqm/errors

OS/2, Windows NT x:\mqm\errors

1.5.4 Other Messages Not Specific to an Individual Queue Manager

Messages of various types not specific to queue manager are written to the following directories:

UNIX /var/mqm/qmgrs/@SYSTEM/errors

OS/2, Windows NT x:\mqm\qmgrs\@SYSTEM\errors

1.6 Manually Recording/Recovering Media Images

Queues that do not go empty will require very old log files if a recovery is necessary.

1.6.1 UNIX, OS/2, Windows NT

"rcdmqimg" (record media image) may be used to force a media image to disk.

"rcrmqobj" (re-create object) may be used to re-create an object from its media image contained in the log.

1.6.2 AS/400

RCDMQMIMG (record media image) may be used to force a media image to disk.

RCRMQMOBJ (re-create object) may be used to re-create an object from its media image contained in the log.

1.6.3 MVS

MQSeries for MVS does not have functions to record/re-create media images. You must have backup copies of the page sets and all of the logs that were required at the time the backup was taken. The required logs are indicated by these messages:

- CSQI024I - System restart RBA.
- CSQI025I - System restart RBA including offline page sets.

1.7 Queue Manager Resources to Protect/Backup

The following lists the queue manager resources you should back up.

1.7.1 UNIX, OS/2, Windows NT

- Logs - /var/mqm/log/xxxx/active/*.LOG
- Log control file - /var/mqm/log/xxxx/amqhlctl.lfh
- System wide "ini" file - /var/mqm/mqs.ini
- The queue manager "ini" file and objects - /var/mqm/qmgrs/xxxx/* (recurse subdirectories)
- The conversion/ccsid tables - /var/mqm/conv/table/*

1.7.2 AS/400

- Journal receivers (AS400)
- Library QMQMDATA

1.7.3 MVS

- Active logs
- Archive logs
- BSDS (bootstrap data set)
- Page sets
- Queue manager objects

Chapter 2. MQSeries for MVS/ESA

This chapter is divided into the following sections:

- 2.1, Background Information about:
 - 2.1.1, Data Sets Used by MQSeries for MVS/ESA
 - 2.1.2, Backup
 - 2.1.3, Restart and Recovery
- 2.2, Tips for Backup and Recovery
- 2.3, Backup and Recovery Procedures
- 2.4, Recovery Scenarios

2.1 Background Information

This section discusses the different types of data sets used by MQSeries for MVS/ESA, the management of data, backup and recovery considerations.

2.1.1 Data Sets Used by MQSeries for MVS/ESA

MQSeries for MVS/ESA uses the following data sets:

- Page sets
- Log data sets
- Bootstrap data set (BSDS)

2.1.1.1 Page Sets

Page sets are VSAM linear data sets (LDS) and are used to store messages and object definitions. They are identified by a page set identifier in the range 00 through 99. MQSeries uses page set 00 to store object definitions.

A *storage class* maps one or more queues to a page set. More than one queue can use the same storage class and you can define as many storage classes as you like, each mapped to a different page set (or the same page set).

The following example shows how to map the local queue TARGET.QUEUE to page set 07 through storage class APPL.

```
DEFINE STGCLASS(APPL) PSID(07)
DEFINE QLOCAL(TARGET.QUEUE) STGCLASS(APPL)
```

MVS/ESA

Note: Since page set zero is used to store all the object definitions (queues, processes, channels, etc), it is essential that page set zero does not become full. Therefore, we recommend that you do not define queues with storage classes that map to page set zero. If you do, you run the risk of filling page set zero if too many messages accumulate there.

2.1.1.2 Log Data Sets

MQSeries for MVS/ESA records all persistent messages in the *active log* as they are put onto queues by applications. The log contains the information needed to recover messages, queues, and the queue manager.

When the active log is full, MQSeries switches to the next available log data set and, if archiving has been switched on during customization, copies the contents of this log to an *archive log*, which can be a data set on DASD or magnetic tape. If there is a problem, MQSeries could require data from archived logs at startup time if it is necessary to recover damaged objects. For greater assurance against, for example, DASD failure, MQSeries supports dual logging for both the active and the archive logs. This means, the same information is recorded into two data sets. If possible, when using dual logging, the log data sets should be on separate volumes.

Recommendation

To minimize the risk of losing data, for example, because of DASD failures, we recommend in a production subsystem to establish both of the following:

1. Archiving (OFFLOAD=YES in macro CSQ6LOGP)
2. Dual logging for:
 - Active logs (TWOACTV=YES in macro CSQ6LOGP)
 - Archive logs (TWOARCH=YES in macro CSQ6LOGP)
 - Bootstrap data sets (TWOBSDS=YES in macro CSQ6LOGP)

These options are specified in the subsystem initialization parameter module CSQZPARM.

When active logs get full, MQSeries copies the data in the active log to an archive log. It is not guaranteed that the current transaction will be completed before the active log fills up. Therefore a transaction could be stored partially in the current active log and partially in an archive log. You should never turn off archiving in a production environment. With only a finite number of active logs, MQSeries must be allowed to archive as

needed. If you turn off archiving and the active logs become full, the queue manager will stop.

Log records

The log is made up of log records, each of which is a set of log data treated as a single unit. A log record is identified by the RBA (relative byte address) of the first byte of its header. That RBA is the offset from the beginning of the log.

There are three types of log records being written to the log:

- Unit-of-recovery log records
- Checkpoint records
- Page set control records

Any change to a queue is made within a unit of recovery. These are called *unit-of-recovery log records*.

MQSeries takes *checkpoints* at the end of a successful restart, at normal termination and during normal operation when a predefined number of log records has been written. This number is defined by the checkpoint frequency operand called LOGLOAD of the installation macro CSQ6SYSP.

Page set control records register the page sets known to MQSeries at each checkpoint.

2.1.1.3 Bootstrap Data Set (BSDS)

The BSDS is a VSAM keyed-sequence data set (KSDS) that stores information about the logs. This information allows the queue manager to locate log records so that it can handle restart processing and can satisfy log read requests during normal processing. For active logs the information shows which logs are full and which are available for reuse. MQSeries supports dual BSDSs, each recording the same information. As for dual active log data sets, copies should, if possible, be on separate volumes.

MQSeries always writes a copy of the BSDS in all of the archived logs. In fact, the archive log is actually two data sets in one: the first data set is a copy of the most recent image of the BSDS, and the second data set is the archive itself. If the archive log is on DASD, the BSDS copy is a separate file on the same volume.

Archive log name	MQM.ARC1CSQ1.A0000005
BSDS copy name	MQM.ARC1CSQ1.B0000005

MVS/ESA

The names are the same, except that the lowest-level qualifier of the archive log name begins with A and the BSDS copy begins with B.

2.1.2 Backup

Messages and MQ objects are stored in queues. Transactions (MQPUT, MQGET, etc.) are stored in logs. When an MQPUT function call is issued, a transaction is recorded in a log and a message is put on a queue. When a subsequent MQGET function call is issued a second transaction is written to the log and a message is removed from a queue. If these two transactions were issued against the same queue and the queue was empty before the transactions, the queue would still be empty after the transactions, but the log would have two transactions in it.

In a more complex scenario assume that a queue has some messages on it and over a period of time several other MQGETs and MQPUTs were issued. The queue would have some number of messages remaining. These could be some of the original messages and some of the new messages or all of the original messages could have been consumed by the MQGETs and all of the remaining messages are from the MQPUTs. We don't have any way of knowing. If there were a failure of the queue, we could restore the queue to the state it was in before the failure by going back to some point in time when the contents of the queue were known (possibly a backup) and replaying all of the subsequent MQPUTs and MQGETs. MQSeries will usually be able to do this for us automatically by going back into the logs and replaying the transactions forward from that point.

In order to be able to do this manually, for instance in the case of hard disk failure or a disaster, we must have a backup plan. We must know what to back up, what to protect with redundancy, and the procedure to use to restore. The pieces that play a part in this puzzle are:

- Queues
- Active logs/archive logs
- Bootstrap data set (BSDS)
- Page set zero

When queues are created they are indirectly mapped to page sets.

Messages that are put on queues end up on page sets, theoretically. Physically, messages are stored temporarily in buffers before they are written to disk. The goal is for messages to remain in buffers for a very short time and then pass through the system. In a utopic situation messages would never actually get written to the page set.

Before messages are put on queues or taken off queues, the transactions MQGET/MQPUT are written to transaction logs if the messages are persistent. Page zero of each page set contains the minimum log RBA needed to recover the page set in the event of a failure. For this reason it is necessary to have backup copies of page sets. If you do not have a backup copy of a page set that needs to be recovered, at restart time the queue manager will ask for the archive log with the minimum log RBA since the queue manager was created. In other words, you will be asked for the first log that the queue manager ever used.

2.1.3 Restart and Recovery

MQSeries uses *units of recovery* to keep data consistent. A unit of recovery (UR) begins with the first change to MQSeries data after the beginning of the program or following the previous point of consistency (also called syncpoint or commit point); it ends with a later point of consistency. A normal termination of an application program automatically causes a point of consistency.

A system crash may occur and leave a UR in an incomplete state, such that neither ABORT END nor FINAL COMMIT records were written to the log. During restart the state of the UR is determined and appropriate backout and recovery activity takes place. MQSeries writes one of three record types to the log to mark the end of the UR:

END REDO - UR WAS COMMITTED OR RESOLVED TO A COMMITTED STATE.
END UNDO - UR WAS ABORTED OR RESOLVED TO ABORTED STATE.
END TODO - UR IS IN DOUBT.

MQSeries uses its recovery log and the bootstrap data set to determine what to recover when it restarts. The BSDS identifies the active and archive log data sets, and the location of the most recent MQSeries checkpoint in the log.

To recover a page set, MQSeries needs to know how far back in the log to go. During each checkpoint, the log RBA of the checkpoint is recorded in each page set's page 0. This "page set restart RBA" is used to help position the log so that all relevant log records are processed during the REDO pass of recovery.

When a page set copy, taken some time ago, is restored and the system is restarted, the log RBA in the page set's page 0 is used to position the log. The REDO pass log position is the lowest value determined by the values extracted from page 0 of all page sets, the beginning of the last complete

MVS/ESA

checkpoint, or the Begin-UR log record (the first in any unit of work) for all incomplete transactions (this information is in the checkpoint).

Now the REDO pass reads the log sequentially from this RBA to the end of the log. For each REDO log record, the log record is "applied". This means that the appropriate page set record is fetched, the log RBA is compared with the "page set RBA", and if the page set RBA is less than the log RBA, the change implied by the REDO action is made in the page set buffer, and the page set RBA is changed to the RBA of the REDO log record.

Once the REDO pass is completed, there's an UNDO pass to undo the "loser" units of recovery (those that were in flight or in backout when the crash occurred) and then there's a checkpoint.

2.2 Tips for Backup and Recovery

The MQSeries restart process recovers your data to a consistent state by applying the log information to the page sets. If your page sets are damaged, you can resolve the problem using your backup copies of your page sets. If your log data sets are damaged or unavailable, you cannot completely recover. It is recommended that you do the following:

- Periodically take backup copies of page sets.
- Periodically take backup copies of your queue manager objects.
- Use dual logging for your active log, archive log, and bootstrap data sets.
- Keep archive logs you might need.
- Retain the DD name or page set association.

2.2.1 Periodically Take Backup Copies of Page Sets

Having a set of backup copies of page sets and all the logs from the point that these copies were taken, MQSeries can be recovered to the point of failure. The more recent the backup copies are, the quicker MQSeries can recover the data in the page sets and the shorter log data sets have to be available.

You should have a backup strategy for your page sets. The log range kept should be at least enough so that a page set can be recovered from its previous backup, by applying all the appropriate logged updates between that backup and now.

The following two factors tell you what log data to keep:

- MAXARCH parm of CSQ6LOGP macro specifies how many archive logs are known in the BSDS. Once this number is exceeded, only the most recent ones are kept.
- SMS MANAGEMENT class and retention period will determine when/if archive logs are deleted.

Important

If you do not have backups of the page sets then the queue manager will not be able to recover any data from the page set.

2.2.2 Periodically Take Backup Copies of Queue Manager Objects

Take regular backups of your queue manager objects into a sequential data set which can then be put into a queue manager via CSQINP2 if it is necessary to clean out page sets and logs and restart from scratch.

2.2.3 Use Dual Logging for Your Active Log, Archive Log, and BSDS

This increases the chances of recovering from all problems. On the other hand, it will affect the performance of the system.

Keep the dual logs on separate volumes, if possible.

2.2.4 Keep Archive Logs You Might Need

You must keep sufficient archive logs to recover a page set from a restored backup copy. Otherwise, MQSeries will not be able to restore the page set to its current state.

You should also maintain sufficient archive logs to allow restart of the system; that is, any uncommitted URs (unit-of-recoveries) should be entirely within this log range.

If you have active (maybe inflight, or in doubt) URs with a start RBA on tape, then at restart time you will have to mount all tapes from that earliest RBA forward to be able to restart the system.

You can determine the earliest RBA required by a potential restart at any time (the system is active) by doing a DISPLAY USAGE command. Armed with this information, and a BSDS print, you'll know how many tapes would need to be mounted to get the entire log read during a restart. This log RBA should be within the range of the archive logs known in the BSDS.

Figure 1 on page 16 shows a sample JCL of the print log map utility to get the contents of the BSDS.

MVS/ESA

```
//JU004    EXEC PGM=CSQJU004
//STEPLIB DD DISP=SHR,DSN=MQM.V1R2.SCSQANLE
//         DD DISP=SHR,DSN=MQM.V1R2.SCSQAUTH
//SYSPRINT DD SYSOUT=*,DCB=BLKSIZE=629
//SYSUDUMP DD SYSOUT=*
//SYSUT1  DD DISP=SHR,DSN=MQM.SA25.CSQ3.BSDS01
```

Figure 1. JCL to Invoke the Print Log Map Utility

This utility can be run regardless of whether MQSeries is running.

2.2.5 Retain the DD Name or Page Set Association

MQSeries associates the page set number with the DD name, for example 01 with CSQP0001, and so on. The DD name is used by MQSeries to write recovery log records for a page set. For this reason, you must not move or rename page sets. Otherwise, MQSeries tries to recover the page sets with the wrong log data.

Note: If you intend archive logging to tape, you might want to consider making the active logs of a size which optimizes tape use when they are archived.

2.3 Backup and Recovery Procedures

This section describes procedures you should run periodically to avoid costly and time-consuming losses of data.

MQSeries can recover objects and persistent messages to their current state only if there is:

1. A copy of all page sets from an earlier point
2. All the MQSeries logs since that point

So it is very important to make backup copies of the page sets. But, before you do this, you should create a so called *point of recovery*. There are two methods to do this.

2.3.1.1 Method 1

This method requires you to shut down MQSeries to force all updates on to the page sets so that the page sets are in a consistent state.

Use the following procedure:

- Step 1. Issue the DISPLAY USAGE command and record the lowest RBA value in the CSQI024I or CSQI025I messages.

- Step 2. Issue the ARCHIVE LOG command to take a copy of the current active log.
- Step 3. Issue the STOP QMGR MODE(QUIESCE) command to stop the queue manager.
- Step 4. Take backup copies of the page sets using IDCAMS REPRO (as shown in the following section).
- Step 5. Restart the queue manager.

2.3.1.2 Method 2

This method should be used if running a 7 x 24 hours shop. The backup is taken while the system is running. The page sets need to be allocated in a shared way, using SHAREOPTIONS(2,3), so that a copy utility can read the data set while the system is reading and writing it.

If you use this kind of backup, a so called *fuzzy backup*, be aware that updates may be in virtual storage buffers during the backup process. This means page sets are not in a consistent state and if your logs subsequently become damaged or lost you will not be able to use the fuzzy page set backup copies to recover.

Use the following procedure:

- Step 1. Issue the DISPLAY USAGE command and record the lowest RBA value in message CSQI024I or CSQI025I.
- Step 2. Take backup copies of the page sets using IDCAMS REPRO (as shown in the following section).
- Step 3. Issue the ARCHIVE LOG command to take a copy of the current active log.

2.3.2 Backing Up the Page Sets

To be able to recover a page set you periodically need to make backup copies of your page sets.

Therefore, you need to do the following:

- 1. Define new page sets as shown in Figure 2 on page 18.

MVS/ESA

```
//DELDEF EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE (MQM.SA25.CSQ3.PSID00.BACKUP1)
DELETE (MQM.SA25.CSQ3.PSID01.BACKUP1)
DELETE (MQM.SA25.CSQ3.PSID02.BACKUP1)
DELETE (MQM.SA25.CSQ3.PSID03.BACKUP1)

DEFINE CLUSTER -
(NAME(MQM.SA25.CSQ3.PSID00.BACKUP1) -
RECORDS(1000 100) -
LINEAR -
VOLUMES(WTL242) -
SHAREOPTIONS(2 3) ) -
DATA -
(NAME(MQM.SA25.CSQ3.PSID00.BACKUP1.DATA) )
DEFINE CLUSTER -
(NAME(MQM.SA25.CSQ3.PSID01.BACKUP1) -
RECORDS(1000 100) -
LINEAR -
VOLUMES(WTL242) -
SHAREOPTIONS(2 3) ) -
DATA -
(NAME(MQM.SA25.CSQ3.PSID01.BACKUP1.DATA) )
DEFINE CLUSTER -
(NAME(MQM.SA25.CSQ3.PSID02.BACKUP1) -
RECORDS(1000 100) -
LINEAR -
VOLUMES(WTL242) -
SHAREOPTIONS(2 3) ) -
DATA -
(NAME(MQM.SA25.CSQ3.PSID02.BACKUP1.DATA) )
DEFINE CLUSTER -
(NAME(MQM.SA25.CSQ3.PSID03.BACKUP1) -
RECORDS(1000 100) -
LINEAR -
VOLUMES(WTL242) -
SHAREOPTIONS(2 3) ) -
DATA -
(NAME(MQM.SA25.CSQ3.PSID03.BACKUP1.DATA) )
/*
/**
```

Figure 2. JCL to Define New Page Sets

2. Copy the original page sets into the newly created page sets using AMS REPRO, as shown in Figure 3 on page 19.


```

//REPRO    EXEC PGM=IDCAMS,REGION=4M
//DD1     DD DISP=SHR,DSN=MQM.SA25.CSQ3.PSID00
//DD2     DD DISP=SHR,DSN=MQM.SA25.CSQ3.PSID01
//DD3     DD DISP=SHR,DSN=MQM.SA25.CSQ3.PSID02
//DD4     DD DISP=SHR,DSN=MQM.SA25.CSQ3.PSID03
//DD5     DD DISP=SHR,DSN=MQM.SA25.CSQ3.PSID00.BACKUP1
//DD6     DD DISP=SHR,DSN=MQM.SA25.CSQ3.PSID01.BACKUP1
//DD7     DD DISP=SHR,DSN=MQM.SA25.CSQ3.PSID02.BACKUP1
//DD8     DD DISP=SHR,DSN=MQM.SA25.CSQ3.PSID03.BACKUP1
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
          REPRO INFILE(DD1) OUTFILE(DD5)
          REPRO INFILE(DD2) OUTFILE(DD6)
          REPRO INFILE(DD3) OUTFILE(DD7)
          REPRO INFILE(DD4) OUTFILE(DD8)
/*
//*

```

Figure 3. JCL to Back Up Page Sets

Note: If you want to REPRO to tape, you have to set the recfm to F and lrecl to 4096 for your OUT DD.

You should have more than one back up data set and alternate them. If you back up to data set BACKUP2, and the system fails, then you can use data set BACKUP1. If you have just one data set, then if the backup fails while you are copying the data, you have lost your backup copy.

2.3.3 Recovering a Page Set

During restart the queue manager will read from the log and recover the persistent data for the page set. If all the logs are available since the backup was taken, MQSeries will apply all changes made to the page set that are registered in the log.

There are two methods you can use to recover a page set after a page set failure.

2.3.3.1 Method 1

- Step 1. Delete and redefine the page set which got damaged **1** **2**.
- Step 2. Use AMS REPRO to copy the backup data into the page set **3**.
- Step 3. Restart the queue manager as normal:
- ```
-START QMGR PARM(xxxxxx)
```

## MVS/ESA

Step 4. When the queue manager has started successfully, you can run your applications.

Step 5. Back up the page set again.

```
//DELDEF EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE (MQM.SA25.CSQ3.PSID01)
DEFINE CLUSTER -
 (NAME(MQM.SA25.CSQ3.PSID01) -
 RECORDS(1000 100) -
 LINEAR -
 VOLUMES(WTL242) -
 SHAREOPTIONS(2 3))
DATA
 (NAME(MQM.SA25.CSQ3.PSID01.DATA))
/*
//REPRO EXEC PGM=IDCAMS,REGION=4M
//DD1 DD DISP=SHR,DSN=MQM.SA25.CSQ3.PSID01.BACKUP1
//DD2 DD DISP=SHR,DSN=MQM.SA25.CSQ3.PSID01
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
REPRO INFILE(DD1) OUTFILE(DD2)
/*
//
```

Figure 4. JCL to Recover a Page Set

### 2.3.3.2 Method 2

Step 1. Rename the damaged page set ID in your startup procedure (xxxxMSTR) with the name of the backup page set.

```
//*****
/* PAGE SET DATA SETS *
/* *
//*****
//CSQP000 DD DISP=SHR,DSN=MQM.SA25.CSQ3.PSID00
//CSQP001 DD DISP=SHR,DSN=MQM.SA25.CSQ3.PSID01.BACKUP1
//CSQP002 DD DISP=SHR,DSN=MQM.SA25.CSQ3.PSID02
//CSQP003 DD DISP=SHR,DSN=MQM.SA25.CSQ3.PSID03
```

Step 2. Restart the queue manager:  
START QMGR PARM(xxxxxx)

Step 3. Back up the page set again.

**Recommendation**

We recommend using method 1, because the backup version may not be on the optimum DASD volume for performance or if you got an error with the backup page set during restart, your data is lost if you had only one backup.

**2.3.4 Backing Up Your Object Definitions**

Whenever you take backup copies of your queue manager data sets, you should also back up copies of your object definitions. You take the backups into a sequential data set which can then be put in the CSQINP2 initialization input data set if it is necessary to clean out page sets and logs and restart MQSeries from scratch. This output could also be used in case of disaster recovery to create all the definitions of the queue manager.

You can use the MAKEDEF feature of the CSQUTIL COMMAND function as shown in the following sample JCL:

```
//SDEFS EXEC PGM=CSQUTIL,PARM='CSQ1'
//STEPLIB DD DISP=SHR,DSN=MQM.V1R2.SCSQANLE
// DD DISP=SHR,DSN=MQM.V1R2.SCSQAUTH
//CSQP0000 DD DISP=SHR,DSN=MQM.SA25.NEW.PSID00
//OUTPUT1 DD DISP=SHR,DSN=MQM.USER.SOURCE(MAKEOUT)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(CMDINP) MAKEDEF(OUTPUT1)
/*
//CMDINP DD *
DISPLAY STGCLASS(*)
DISPLAY QUEUE(*)
DISPLAY NAMELIST(*)
DISPLAY PROCESS(*)
DISPLAY CHANNEL(*)
/*
```

Figure 5. JCL Using the MAKEDEF Command of CSQUTIL

The MAKEDEF keyword causes the list of DISPLAY commands to be converted into a corresponding set of DEFINE commands. These commands are put into the data set referenced by the ddname parameter of the MAKEDEF keyword, that is, OUTPUT1.

## MVS/ESA

### 2.3.5 Backing Up and Restoring Queues

If you need to take a backup of a queue, use the following JCL:

```
//COPY EXEC PGM=CSQUTIL,PARM=' CSQ3'
//STEPLIB DD DSN=MQM.V1R2.SCSQANLE,DISP=SHR
// DD DSN=MQM.V1R2.SCSQAUTH,DISP=SHR
//OUTPUT DD DISP=(NEW,CATLG),DSN=MQM.USER.QUEUE,
// SPACE=(CYL,(5,1),RLSE),UNIT=SYSDA,
// DCB=(RECFM=VBS,BLKSIZE=23200)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
* COPY ONE QUEUE TO 'OUTPUT'
COPY QUEUE(PHD.CSQ3.REQUESTQ) DDNAME(OUTPUT)
/*
```

Figure 6. JCL to Take a Backup of a Queue

To restore the queue you can use the following JCL:

```
//LOAD EXEC PGM=CSQUTIL,PARM=' CSQ3'
//STEPLIB DD DSN=MQM.V1R2.SCSQANLE,DISP=SHR
// DD DSN=MQM.V1R2.SCSQAUTH,DISP=SHR
//OUTPUT DD DISP=SHR,DSN=MQM.USER.QUEUE
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
* LOAD MESSAGES FROM OUTPUT DATA SET INTO QUEUE' PHD.CSQ3.REQUESTQ'
LOAD QUEUE(PHD.CSQ3.REQUESTQ) DDNAME(OUTPUT)
/*
```

Figure 7. JCL to Restore a Queue

---

## 2.4 Recovery Scenarios

This topic discusses several scenarios and how to recover the queue manager after the following situations:

- Lost or corrupted logs
- Damaged page set
- Lost or damaged BSDS
- Disaster recovery
- Channel re-synchronization

## 2.4.1 Recovering Lost or Corrupted Logs

Since you need your logs for recovery to a consistent point in time, you will run into a problem if your MQSeries subsystem crashes, and there are "in-flight" tasks that have not been committed. Without the logs, MQSeries can not back out any unit-of-work that has not been committed, since it has no copy of the "before" images that would have been written to the logs.

The following scenarios show how to restart MQSeries in one of those situations.

### 2.4.1.1 Scenario 1: One Active Log Deleted or Corrupted

If an active log has been corrupted or deleted you will receive the following errors when starting MQSeries:

```
CSQJ103E - LOG ALLOCATION ERROR 319
DSNAME=MQM.SA25.CSQ3.LOGCOPY1.DS02, ERROR STATUS=17080002
CSQJ232E - OUTPUT DATA SET CONTROL INITIALIZATION PROCESSING FAILED
IEA794I SVC DUMP HAS CAPTURED: 321
DUMPID=038 REQUESTED BY JOB (CSQ3MSTR)
DUMP TITLE=CSQ3,ABN=5C6-00E80084,U=SYSOPR ,C=13700.120.IPC -CS
 QYSIRM,M=CSQYECTE,PSW=077C20008A9086A0,ASID=0059
*CSQV086E - MQSERIES ABNORMAL TERMINATION REASON=00E80084
```

#### Important

You can only recover a damaged active log data set, if DUAL logging was used.

In such a situation you have to perform the following tasks (see Figure 8 on page 24):

- Step 1. Redefine a new active log data set.
- Step 2. Use AMS REPRO to copy the LOGCOPY of the damaged log to the new log data set.
- Step 3. Restart the queue manager as normal with the command:  
START QMGR PARM(xxxxxx)

## MVS/ESA

```
//REPROLG EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//DD1 DD DISP=SHR,DSN=MQM.SA25.CSQ3.LOGCOPY2.DS02
//DD2 DD DISP=SHR,DSN=MQM.SA25.CSQ3.LOGCOPY1.DS02
//SYSIN DD *
DELETE (MQM.SA25.CSQ3.LOGCOPY1.DS02)
SET MAXCC = 0

DEFINE CLUSTER -
 (NAME (MQM.SA25.CSQ3.LOGCOPY1.DS02) -
 LINEAR -
 VOLUMES(WTL242) -
 UNIQUE -
 RECORDS(1000))
 DATA -
 (NAME (MQM.SA25.CSQ3.LOGCOPY1.DS02.DATA))

REPRO -
 INFILE(DD1) -
 OUTFILE(DD2)
/*
//
```

Figure 8. JCL to Re-define and Restore an Active Log

### 2.4.1.2 Scenario 2: Both Active Logs Deleted or Corrupted

If both logs (supposing dual logging was active) are corrupted or have been deleted, it is possible to restart MQSeries provided you have a consistent set of page sets.

Use the following procedure if you run into that situation:

- Step 1. Define and format new page sets. You can use the JCL in `thlqual.SCSQPROC(CSQ4PAGE)` to do this.
- Step 2. Use the `RESETPAGE` function of `CSQUTIL` to copy the "original" contents of the page set in question to the newly-formatted page set. The extra step that the reset page function will do is to also reset the log information in the page set itself, which it will obtain from the `BSDS` and `LOG` data set that was just created.

```

//RESET EXEC PGM=CSQUTIL
//STEPLIB DD DISP=SHR,DSN=MQM.V1R2.SCSQANLE
// DD DISP=SHR,DSN=MQM.V1R2.SCSQAUTH
//CSQS0000 DD DISP=OLD,DSN=MQM.SA25.CSQ3.PSID00 SOURCE
//CSQS0001 DD DISP=OLD,DSN=MQM.SA25.CSQ3.PSID01 SOURCE
//CSQS0002 DD DISP=OLD,DSN=MQM.SA25.CSQ3.PSID02 SOURCE
//CSQS0003 DD DISP=OLD,DSN=MQM.SA25.CSQ3.PSID03 SOURCE
//CSQT0000 DD DISP=OLD,DSN=MQM.SA25.CSQ3.NEW.PSID00 TARGET
//CSQT0001 DD DISP=OLD,DSN=MQM.SA25.CSQ3.NEW.PSID01 TARGET
//CSQT0002 DD DISP=OLD,DSN=MQM.SA25.CSQ3.NEW.PSID02 TARGET
//CSQT0003 DD DISP=OLD,DSN=MQM.SA25.CSQ3.NEW.PSID03 TARGET
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
RESETPAGE
/*

```

Figure 9. JCL Using the RESETPAGE Function of CSQUTIL

- Step 3. Define new log data sets and BSDS using the utility CSQJU003. You can use the sample JCL thlqual.SCSQPROC(CSQ4BSDS).
- Step 4. Restart the queue manager using new log data sets and BSDSs and with the new copy of the page sets.

Effectively, what this does is create a "new" starting point for an MQ restart, with newly-defined BSDS and logs, and a copy of the page sets which reflect this BSDS and log information.

**Note:** If the queue manager stopped abnormally, you have to run an AMS VERIFY job before using the RESETPAGE function. Otherwise you will get a VSAM OPEN error, because the page sets were not closed properly.

```

//IDCAMS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
 VERIFY DATASET(MQM.SA25.CSQ3.PSID00)
 VERIFY DATASET(MQM.SA25.CSQ3.PSID01)
 VERIFY DATASET(MQM.SA25.CSQ3.PSID02)
 VERIFY DATASET(MQM.SA25.CSQ3.PSID03)

```

Figure 10. JCL Using the AMS VERIFY Function

## MVS/ESA

### 2.4.2 Recovering a Damaged Page Set

If a page set is damaged and you get an I/O error, restore it from a backup copy and restart MQSeries. MQSeries automatically applies any updates from the logs that are necessary.

- Step 1. Restore the page set from a backup copy using the JCL shown in Figure 11.

```
//IDCAMS EXEC PGM=IDCAMS
//DD1 DD DSN=MQM.SA25.CSQ3.BACKUP.PSID00,DISP=SHR
//DD2 DD DSN=MQM.SA25.CSQ3.PSID00,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
 REPRO -
 INFILE(DD1) -
 OUTFILE(DD2)
/*
```

Figure 11. JCL to Restore from a Backup Page Set

- Step 2. Restart the queue manager with the command:

```
START QMGR PARM(xxxxxx)
```

- Step 3. Take new backup copies of the page sets.

### 2.4.3 Recovering a Lost or Damaged BSDS

If MQSeries is using dual BSDS and one BSDS becomes damaged, MQSeries continues to operate with one BSDS in a single BSDS mode. To return to dual BSDS mode you have to:

- Step 1. Delete/define a new BSDS with the same name as the damaged BSDS (Figure 12 on page 27).

- Step 2. Issue the MQSeries command RECOVER BSDS.

This will make a copy of the valid BSDS in the newly allocated data set.



```

//BSDS EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
DELETE (MQM.SA25.CSQ3.BSDS01) ERASE
SET MAXCC = 0
DEFINE CLUSTER -
 (NAME(MQM.SA25.CSQ3.BSDS01) -
 VOLUMES(WTL242) -
 UNIQUE -
 SHAREOPTIONS(2 3)) -
DATA -
 (NAME(MQM.SA25.CSQ3.BSDS01.DATA) -
 RECORDS(60 60) -
 RECORDSIZE(4089 4089) -
 CONTROLINTERVALSIZE(4096) -
 FREESPACE(0 20) -
 KEYS(4 0)) -
INDEX -
 (NAME(MQM.SA25.CSQ3.BSDS01.INDEX) -
 RECORDS(5 5) -
 CONTROLINTERVALSIZE(1024))
/*

```

Figure 12. JCL Used to Define a New BSDS

If MQSeries terminated before you recovered the damaged BSDS, the next restart will fail:

```

CSQJ107E - READ ERROR ON BSDS 889
DSNAME=MQM.SA25.CSQ3.BSDS01, ERROR STATUS=0874
CSQJ117E - INITIALIZATION ERROR READING BSDS 890
DSNAME=MQM.SA25.CSQ3.BSDS01, ERROR STATUS=0874
CSQJ119E - BOOTSTRAP ACCESS INITIALIZATION PROCESSING FAILED
IEA794I SVC DUMP HAS CAPTURED: 892
DUMPID=039 REQUESTED BY JOB (CSQ3MSTR)
DUMP TITLE=CSQ3,ABN=5C6-00E80084,U=SYSOPR ,C=13700.120.IPC -CS
QYSIRM,M=CSQYECTE,PSW=077C20008A9086A0,ASID=005A
*CSQV086E - MQSERIES ABNORMAL TERMINATION REASON=00E80084

```

To get MQSeries restarted again do the following:

1. Redefine the BSDS as shown in Figure 12.
2. Use AMS REPRO to make a copy of the valid BSDS in the newly allocated data set.

## MVS/ESA

```
//IDCAMS EXEC PGM=IDCAMS
//DD1 DD DISP=SHR,DSN=MQM.SA25.CSQ3.BSDS02
//DD2 DD DISP=SHR,DSN=MQM.SA25.CSQ3.BSDS01
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
 REPRO -
 INFILE(DD1) -
 OUTFILE(DD2)
/*
```

Figure 13. JCL Used to Recover a BSDS

3. Restart MQSeries using the following command:

```
START QMGR PARM(xxxxxxxx)
```

### Important

You can only recover a damaged BSDS if you are operating in dual BSDS mode.

## 2.4.4 Disaster Recovery

If you have any kind of hardware errors on your system, it might be necessary to recover the MQSeries system at a remote site.

The following backups are required in order to perform disaster recovery at a remote system where no MQSeries system is installed:

- A backup copy of the page sets
- A copy of the archive logs and BSDSs
- A copy of the MQSeries target libraries
- A copy of the MQSeries subsystem name table entry  
SYS1.PARMLIB(IEFSSNxx)
- A copy of the program entry CSQYASCP in the Program Properties Table (PPT) in SYS1.PARMLIB(SCHEDxx)
- A copy of the SYS1.PROCLIB xxxxMSTR and xxxxCHIN procedures (xxxx is the subsystem name)
- If using APPC/MVS:
  - A copy of the LUADD entry in SYS1.PARMLIB(APPCPMxx)
  - A copy of the LUs defined in SYS1.VTAMLST
  - A copy of the CPI-C entries

- If using TCP/IP, verify that the parameter TCPNAME in the channel initiator parameter module CSQ4XPRM contains the correct name of the TCP/IP address space. If not, edit and recompile it.

The following steps should be performed before you start the queue manager at a recovery site:

- Step 1. Copy the subsystem name table entry into your active IEFSSNxx member.

**Important**

You must IPL the system before this change takes effect.

**Note:** If you start a new queue manager on a different system using the backup copy and logs from another subsystem, the subsystem name must be the same. This is required, because the subsystem name is recorded in the log during checkpoint.

- Step 2. Add the thlqual.SCSQLINK data set to the link list in SYS1.PARMLIB(LNKLSTxx).
- Step 3. The load libraries thlqual.SCSQAUTH and thlqual.SCSQLINK must be APF authorized.
- Step 4. Copy of the xxxxMSTR and xxxxCHIN procedures into the SYS1.PROCLIB of the new MVS.
- Step 5. Copy the MQSeries target libraries using IEBCOPY.
- Step 6. Define new page sets and load them with the backup copy using AMS REPRO.
- Step 7. Define new active logs.
- Step 8. Define a new BSDS and REPRO into it the *most recent* archived BSDS.
- Step 9. Use the print log map utility CSQJU004 to print information from this most recent BSDS. At the time this BSDS was archived, the most recent archived log you have, would have just been truncated as an active log, and will not appear as an archived log. Record the STARTRBA and ENDRBA of this log.
- Step 10. Use AMS REPRO to copy the most recent archived log into one of the active logs.
- Step 11. Use the CSQJU003 utility to remove all active log information from the BSDS:

## MVS/ESA

```
DELETE DSNAME=MQM.SA25.CSQ3.LOGCOPY1.DS01
DELETE DSNAME=MQM.SA25.CSQ3.LOGCOPY1.DS02
DELETE DSNAME=MQM.SA25.CSQ3.LOGCOPY2.DS01
DELETE DSNAME=MQM.SA25.CSQ3.LOGCOPY2.DS02
```

- Step 12. Use CSQJU003 to add active logs to the BSDS, including RBA range of logs used in Step 8 as found in Step 7.

```
NEWLOG DSNAME=MQM.SA25.CSQ3.LOGCOPY1.DS01,COPY1
 STARTRBA=000000197000,ENDRBA=0000005CEFFF
NEWLOG DSNAME=MQM.SA25.CSQ3.LOGCOPY1.DS02,COPY1
NEWLOG DSNAME=MQM.SA25.CSQ3.LOGCOPY2.DS01,COPY2
 STARTRBA=000000197000,ENDRBA=0000005CEFFF
NEWLOG DSNAME=MQM.SA25.CSQ3.LOGCOPY2.DS02,COPY2
```

- Step 13. Add a conditional restart record to the BSDS. Specify the following statement in CSQJU003:

```
CRESTART CREATE,ENDRBA=05CF000
```

ENDRBA is the high RBA of the most recent archive log available (as found in step 7).

The BSDS now describes: 1 active log with an RBA range, and the other active logs as being empty, all the archived logs you have available and no checkpoints beyond the end of your logs.

- Step 14. If APPC/MVS is being used, you have to:

- Copy the LUADD entry in SYS1.PARMLIB(APPCPMxx).
- Copy the LUs into the SYS1.VTAMLST.
- Copy the CPI-C entries into your side information data set.

- Step 15. For TCP/IP, edit the parm TCPNAME of the channel initiator parameter module CSQ4XPRM to point to the TCP/IP address space in the new MVS system and recompile it.

- Step 16. After the sessions fail (because you had a disaster), you can change the DNS (domain name server) tables, so that the hostname now points at the IP address of the recovery site. The next time your distributed server tries to start a channel to MVS, it will start that channel with the new queue manager.

- Step 17. The queue manager can now be restarted as normal:

```
START QMGR PARM(yyyyyyyyy)
```

During initialization, an operator reply message will be issued:

```
CSQJ245E +cpf RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.
 REPLY Y TO CONTINUE, N TO CANCEL
```

Reply Y to start the queue manager. MQSeries will start and recover data up to ENDRBA specified in the CRESTART statement.

**Note:** When you define the SAME queue manager on two MVS systems, only one should be active at a given time.

For the APPC/MVS definitions, it is not necessary to IPL the system, or even to stop and restart APPC/MVS itself. You can have identical LUADD entries on both systems provided that only one set of LUs is active at a time. The side information can also be the same in two sideinfo KSDS data sets.

For TCP/IP you have to change the TCP/IP hostname in the CONNAME attribute of the SENDER channel definitions on the queue managers that you want to communicate with this one.

**Recommendation**

Always use the hostname in the MQSeries channel definition, CONNAME attribute; never use the IP address in that parameter.

## 2.4.5 Channel Re-synchronization

This section discusses different scenarios and provides useful hints if channels are not synchronized automatically after a restart of the queue manager or channel initiator.

If you are communicating with any other queue manager, either using TCP/IP or LU 6.2, you must ensure that a queue is available with the name SYSTEM.CHANNEL.SYNCQ. This is used to keep sequence numbers and logical unit of work identifiers (LUWID).

The following describes scenarios for these events:

- TCP/IP has to be restarted.
- The channel initiator must be restarted.
- Re-synchronize the channels manually.
- The channel may have stopped.
- Error messages for channels are reported.
- Message sequence number errors occurred.

### 2.4.5.1 TCP/IP Restart

After TCP/IP is stopped and restarted, you will have to restart the CHANNEL INITIATOR ADDRESS SPACE to allow it to recognize the new TCP/IP.

## MVS/ESA

### 2.4.5.2 Channel Initiator Restart

If you have restarted the queue manager or the channel initiator on MVS, you must ensure that *runmqchi* is started on any distributed platform you are communicating with. The channel initiator will manage the channel in RETRY to restart successfully.

### 2.4.5.3 Re-synchronizing the Channels Manually

A channel can be in-doubt with its partner channel about which messages have been sent and received. This would be a situation which is handled (resolved) automatically by the queue manager.

There might be situations where you manually have to re-synchronize the channels.

This is done using the following commands:

- For the sending side of the channel:  
DISPLAY CHSTATUS(channel-name) SAVED CURLUWID
- For the receiving side of the channel:  
DISPLAY CHSTATUS(channel-name) SAVED LSTLUWID

If the two LUWIDs are the same, the sending side can remove the in-doubt message. This is done by the following command:

```
RESOLVE CHANNEL(channel-name) COMMIT
```

If the two LUWIDs are different, the sending side cannot remove its in-doubt messages on the transmission queue and resend them. To resolve this problem issue the following command:

```
RESOLVE CHANNEL(channel-name) BACKOUT
```

### 2.4.5.4 Channel Stopped

If for any reason a sender channel is in status RETRYING, verify that the receiver channel is not stopped as shown in Figure 14 on page 33:

```

 List Channels

Type action codes. Then press Enter.
 1=Display 2=Define like 3=Alter 4=Delete 5=Reset
 6=Start 7=Stop 8=Ping 9=Resolve

 Name Type Status
- WTR05054.TC.CSQ3 CHLRECEIVER INACTIVE
- WTR05110.TC.CSQ3 CHLRECEIVER STOP
- WTR05133.TC.RACSQ3M1 CHLRECEIVER INACTIVE
- WTR05133.62.RACSQ3M1 CHLRECEIVER INACTIVE
 ***** End of list *****

```

Figure 14. List Channels Output

If it is stopped, you have to start it manually. You can use the ISPF panel shown in Figure 14 or use the following command from the console:

```
START CHANNEL(channel-name)
```

#### 2.4.5.5 Messages for Channels

The channel initiator reports when a channel was started or stopped. This means that the help desk people can see what's going on in the system even if no error has occurred. The messages are:

```
+CSQX500I - CSQXRESP Channel WTR05110.TC.CSQ3 started
```

```
+CSQX501I - CSQXRESP Channel WTR05110.TC.CSQ3 is no longer active
```

If you stop the queue manager while some channels are running, you will receive the following error messages.

- Sender channels:

```
+CSQX599E - CSQXRCTL Channel CSQ3.TC.WTR05110 ended abnormally
```

## MVS/ESA

- Receiver channels:
  - +CSQX208E - CSQXRESP Error receiving data, 500 connection 9.24.106.122, TRPTYPE=TCP RC=000003F1
  - +CSQX599E - CSQXRESP Channel WTR05110.TC.CSQ3 ended abnormally

You can ignore these messages. After restart the channels are restarted automatically. This is indicated by the message:

+CSQX500I - CSQXRESP Channel WTR05110.TC.CSQ3 started

### Recommendation

If you are using TCP/IP we recommend stopping the sender channels before stopping the queue manager or channel initiator address space, just to end things cleanly.

If a channel is not started or stopped due to a problem, MQSeries will always report:

+CSQX599E - CSQXRCTL Channel CSQ3.TC.WTR05110 ended abnormally

If messages are sent to the dead-letter queue local or remote, MQSeries sends:

+CSQX548E - csect-name Messages for CSQ3.TC.WTR05110 sent to local dead-letter queue  
+CSQX544E - csect-name Messages for CSQ3.TC.WTR05110 sent to remote dead-letter queue

### 2.4.5.6 Message Sequence Number Errors

Message sequence number errors (MSN) indicate a mismatch in the sequence number for the channels. The message you will get is:

+CSQX526E - CSQXRCTL Message sequence error for CSQ3.TC.WTR05110, sent=1 expected=9

This situation can be resolved at the sending end of the channel with the command:

```
RESET CHANNEL(channel-name) SEQNUM(xx)
```

where xx is the sequence number shown in message CSQX526E.

**Note:** It is recommended that this command be issued when the channel is inactive.



## MVS/ESA

This problem is most likely to occur in the following scenarios:

- If the queue manager to which MVS is talking has two queue managers with the same name and identically named channels. MQSeries does not support multiple identically named channels from the same connection name.
- If the IP address has changed between connections the MSN on the receiver channel is set to 1. You will then have to perform the RESET CHANNEL command to re-synchronize the channel's MSN.

**MVS/ESA**

---

## Chapter 3. MQSeries for AS/400

This chapter touches on various issues pertaining to MQSeries' use of journaling and its management, backups of the pertinent objects and libraries, and various recovery scenarios. In addition, tips and concepts are given and explained further.

This chapter contains sections about:

- 3.1, Journaling
- 3.2, MQSeries for AS/400 Backup Procedures
- 3.3, MQSeries for AS/400 Recovery Scenarios
- 3.4, Useful Hints for MQSeries Backup and Recovery

---

### 3.1 Journaling

MQSeries uses the OS/400 journaling support to recover objects and data when they become damaged. Journals and receivers consume disk space and should be managed to optimize this resource.

#### 3.1.1 OS/400 Journaling

OS/400 file journaling is a facility for capturing changes of database (DB) records to a separate repository. It captures images of changed records, along with who did it, at what time, and with what program. Optionally, OS/400 journaling can hold the "before" image (forward recovery) allowing file changes to be rolled forward or backward. OS/400 journaling should be used anytime there are critical transactions that require up to the minute recovery capabilities and when nightly save windows do not provide enough time to save all physical files. Some of the benefits are listed below:

- No application changes are required to implement or use.
- Journal operations are managed via CL commands or menus.
- Operations include CL commands for recovery using journal receiver entries.
- It is very easy to implement.

Two objects are involved in the journaling operation:

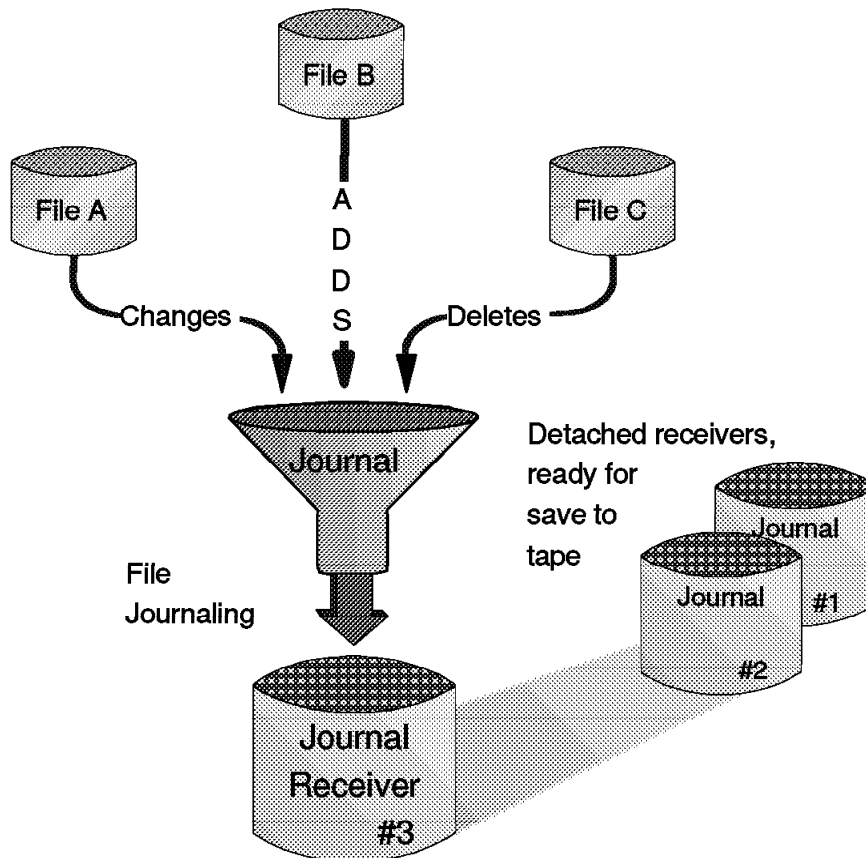
- The Journal (\*JRN) object used to associate journaling with DB files.
- The Journal Receiver (\*JRNRCV), which contains the actual file change (journal) entries. The journal receiver is eventually detached and saved

to tape while a replacement receiver is created. Journal receiver changes may occur while files are in use.

**Recommendations**

- Consider user ASPs for receiver. This provides better protection against disk failures.
- May require additional DASD space. The receivers present on the system will incur additional storage.

Figure 15 illustrates OS/400 journaling concepts.



5222522201

Figure 15. Illustration of OS/400 Journaling Concept

### 3.1.2 MQSeries and OS/400 Journaling

MQSeries for OS/400 uses two journals that are in the QUSRSYS library:

|                |                                                                                       |
|----------------|---------------------------------------------------------------------------------------|
| <i>AMQAJRN</i> | Controls updates to local objects.                                                    |
| <i>AMQRJRN</i> | Controls the flow of messages over the communications link. It details channel usage. |

Each journal has journal receivers attached. MQ writes information about all MQSeries operations to the receivers.

The MQSeries journal receivers are located in the QMQMDATA library. The journal receiver names associated with the MQSeries local objects are in the form of AMQAnnnnnn where A stand for "local" and nnnnnn is a sequence number.

The remote journal receiver names associated with the MQSeries remote objects are in the form of AMQRnnnnnn where R stands for "remote" and nnnnnn is a sequence number.

For more information refer to the *MQSeries for AS/400 Administration Guide*, GC33-1956.

There is no automatic management of remote journal receivers by MQSeries. This is done either by the user or by OS/400 via MGRCV(\*SYSTEM) parameter. For a more detailed discussion on journals and journal management, refer to 3.1.3, "Journal Management."

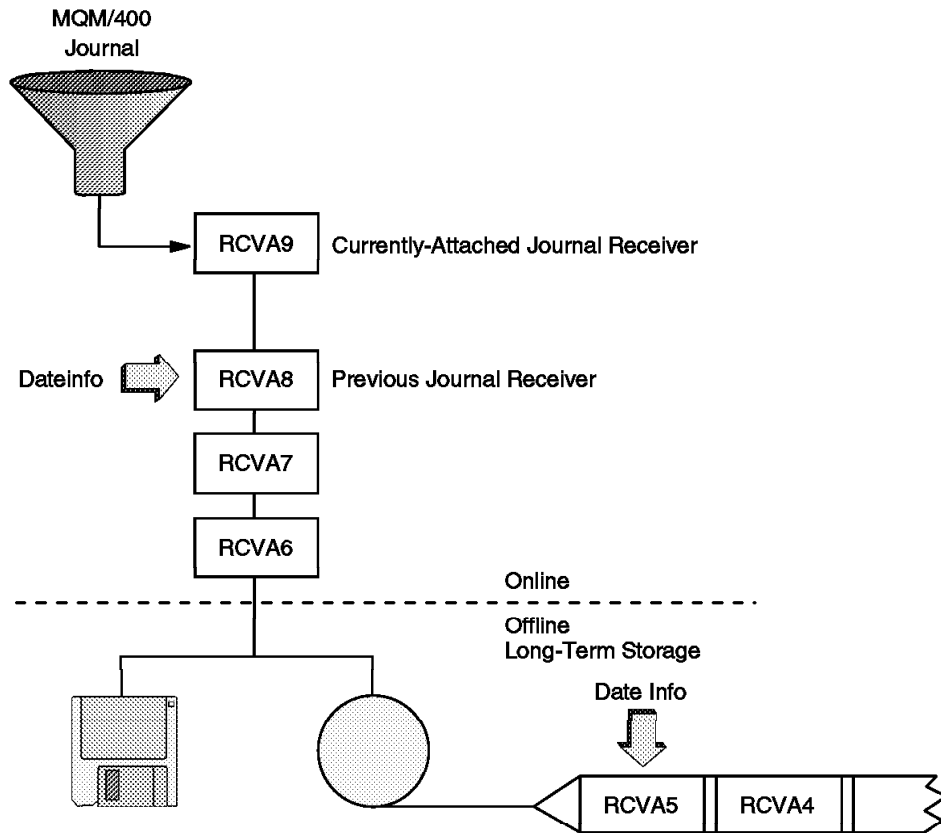
### 3.1.3 Journal Management

Journal management is important as MQSeries uses the journal and its receivers to start and restart MQ and also assists in media recovery and persistent message retention. Because journal receivers can use up valuable disk space, we need to manage them to ensure that only the required receivers are present on the system. Figure 16 on page 40 illustrates this concept.

Before deleting journal receivers you need ensure that you have made a backup copy and that these journal receivers are not required for either MQ startup or media recovery.

In the above diagram, the MQM/400 journal has a currently attached journal receiver, RCVA9. The system detaches the current receiver when the threshold is reached and creates a new one based on a similar naming convention.

## AS/400



5222\522202

Figure 16. MQSeries for AS/400 Journal and Receiver Management

To summarize, the three salient points in journal and receiver management are:

- Ensure that the required receivers are present.
- Advance the media and startup recovery points via RCDMQMIMG.
- Let the system manage the receivers.

These are discussed further below.

### 3.1.3.1 Ensuring the Required Receivers Are Present

As journal receivers record the necessary changes to objects, their presence is vital to help recover from damage to local objects and for restart of MQSeries for AS/400. The replay of these receivers must have a known starting point.

There are two starting points for the local journal or AMQAJRN. These are given by two messages in the *QSYSOPR* message queue:

**AMQ7460** This is the syncpoint for the queue manager startup recovery. Syncpoints are recorded dynamically by the AS/400 system job AMQALMP4. This is shown in Figure 17.

```

Additional Message Information

Message ID : AMQ7460
Date sent : 06/22/98 Time sent : 12:47:13

Message : MQM startup journal information.

Cause : This message is issued periodically by MQM to help you
 identify which journal receivers can be removed from the system because they
 are no longer required for startup recovery.
Recovery : None
Technical Description : The date of the oldest journal entry
 required for startup recovery is 06/22/98 12:46:22. This entry is contained
 in journal receiver AMQA000037. Journal receivers older than this are not
 required by MQM for startup recovery. If the journal receiver name is
 *****, MQM was unable to determine the name of the receiver. The format
 of the date string is *MDY.

 Bottom

Press Enter to continue.

F1=Help F3=Exit F6=Print F9=Display message details F12=Cancel
F21=Select assistance level

```

Figure 17. MQSeries for AS/400 Startup Journal Information AMQ7460

**AMQ7462** This is the media image for object recovery. Media images are recorded manually with RCDMQMIMG .

In the case of the remote journal, AMQRJRN, there is only one file being journaled. This file is the AMQRSYNA file. This file has only one starting point. This starting point is the time when the last *SAVOBJ* or *SAVLIB* was done. Here remote journaling is independent of the local queue manager journaling. RCDMQMIMG does not have any affect on the AMQRSYNA file.

An example of this message from the *QSYSOPR* message queue is shown in Figure 18 on page 42.

```

Additional Message Information
Message ID : AMQ7462
Date sent : 06/22/98 Time sent : 12:47:13

Message : MQM media recovery journal information.

Cause : This message is issued periodically by MQM to help you
 identify which journal receivers can be removed from the system because they
 are no longer required for media recovery.
Recovery : None
Technical Description : The date of the oldest journal entry
 required for media recovery is 06/22/98 12:19:02. This entry is contained in
 journal receiver AMQA000033. Journal receivers older than this are not
 required by MQM for media recovery. If the journal receiver name is
 *****, MQM was unable to determine the name of the receiver. The
 format of the date string is *MDY.

 Bottom

Press Enter to continue.

F1=Help F3=Exit F6=Print F9=Display message details F12=Cancel
F21=Select assistance level

```

Figure 18. MQSeries for AS/400 Media Recovery Journal Information AMQ7462

### 3.1.3.2 Using Record Media Image Command

It is recommended to run the "Record MQM Object Image" command to reduce the number of journal receivers needed on the system:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL)
```

For more information refer to 3.4.1, "RCRMQMIMG" on page 64.

### 3.1.3.3 Specify Delete Receivers Parameter \*NO for AMQAJRN

The current default for the "Delete Receivers" (DLTRCV) parameter for the journal AMQAJRN is \*NO, which means the journal receivers are not deleted by the system.

*Do not change this parameter!*

It is used in conjunction with the "Manage Receivers" (MNGRCV) parameter \*SYSTEM for the local journal. When an attached journal receiver reaches its threshold, the system detaches the current AMQAJRN journal receiver and creates and attaches a new AMQAJRN journal receiver. The existing (old) journal receivers that are present on the system are not deleted. The detached journal receivers are required for startup and media recovery.



### 3.1.4 Tips for MQSeries Journaling

The tips given below are relevant to our discussion on journaling and provide useful additional information.

#### 3.1.4.1 Finding Current Journal Receiver and Attributes

The command to find the current journal receiver is:

- On the AS/400 command line, issue:

```
WRKJRNA JRN(QUSRSYS/AMQAJRN)
```

You will next get a panel similar to that shown in Figure 19.

```

Work with Journal Attributes

Journal : AMQAJRN Library : QUSRSYS

Auxiliary storage
 pool : 1
Message queue . . . : AMQAJRNMSG
 Library : QMQMDATA
Manage receivers . . : *SYSTEM
Delete receivers . . : *NO
Text : *BLANK

Type options, press Enter.
 8=Display attributes

Attached
Option Receiver Library
 AMQA000021 QMQMDATA

Bottom

F3=Exit F5=Refresh F12=Cancel F13=Display journaled files
F14=Display journaled access paths F24=More keys

```

Figure 19. MQSeries for AS/400 Display Current Journal Receiver

The steps to find the current journal attributes are:

- Type option 8 in the above panel and press Enter.
- You will next get the panel shown in Figure 20 on page 44. The parameters to note here are Threshold and Size. The Threshold parameter indicates when the system creates a new journal receiver. This value is in KB. The Size parameter indicates the current size of the journal receiver in bytes.

```

 Display Journal Receiver Attributes
Receiver : AMQA000021 Library : QMQMDATA
Journal : AMQAJRN Library : QUSRSYS
Threshold : 17000 Size : 262144
Attach date : 06/12/98 Attach time : 09:01:28
Detach date : 00/00/00 Detach time : 00:00:00
Save date : 00/00/00 Save time : 00:00:00
Text : MQM LOCAL JOURNAL

Auxiliary storage pool : 1
Status : ATTACHED
Number of entries : 156
Minimized fixed length : NO
Maximum entry specific data length : 32000
Display Journal Receiver Attr. : 0
First sequence number : 12968
Last sequence number : 13123
Pending transactions : NO

More...

F3=Exit F5=Refresh F6=Display associated receivers
F10=Work with journal attributes F12=Cancel

```

Figure 20. MQSeries for AS/400 Display Journal Receiver Attributes

### 3.1.4.2 Deleting Redundant MQSeries Checkpoint Journal Receivers

There are three tasks involved here:

1. Tidying up shared storage in QMQMDATA.
2. Force an MQSeries checkpoint.
3. Restart the queue manager.

Below are the procedures to accomplish the above tasks.

#### Tidying up shared storage:

- Step 1. Shut down all MQSeries channels.
- Step 2. Warn users who have the library QMQM in their \*libl that they are about to be signed-off.
- Step 3. Call the quiesce program with CALL QMQM/AMQIQES4.
- Step 4. If the message AMQ6906 indicates that not all MQ jobs have ended repeat CALL QMQM/AMQIQES4 .
- Step 5. Tidy up cache storage with CALL QMQM/AMQIQEM4.

Step 6. Delete user spaces with the following commands:

```
DLTUSRSPC *ALL/QXCSGLOBAL
DLTUSRSPC *ALL/QMQMLOCAL
```

**Note:** Both should not exist.

Step 7. Sign-off and sign-on again.

Step 8. Start the queue manager with STRMQM.

#### Forcing MQSeries checkpoint:

Step 1. Record media image with the command:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL)
```

Step 2. End the queue manager:

```
ENDMQM MQMNAME(Queue Manager name) OPTION(*IMMED)
```

Step 3. Start the queue manager again with STRMQM .

Step 4. Display the system operator messages with DSPMSG QSYSOPR .

Check for the most recent messages. We are looking for two messages:

- AMQ7460 MQM startup journal information
- AMQ7462 MQM media recovery information

**Note:** Using PF1 for help we should be able to display the technical description. Suffice to say that these two messages will give the name of the oldest receiver. Both messages will probably reference the same receiver. If not take the oldest or lowest receiver number.

Step 5. Save the backup copy of QMQMDATA: SAVLIB LIB(QMQMDATA).

Step 6. Work with Objects in QMQMDATA library: WRKOBJPDM QMQMDATA.

Step 7. Use option 4 (delete) against all \*JRNRCVs which are older than the receiver identified in the earlier steps. You can reply I (ignore) to any messages about receivers not fully saved.

#### Restarting the queue manager:

Step 1. End the queue manager with the command:

```
ENDMQM MQMNAME(queue manager name) OPTION(*IMMED)
```

Step 2. Start the queue manager with STRMQM.

Step 3. Display the QSYSOPR message queue with DSPMSGQ QSYSOPR.

**Note:** You will see new AMQ7460 and AMQ7462 messages!

### 3.1.4.3 Changing to System Managed Journal Receivers

In most installations system managed receivers is the default. If not, you may change the way receivers are managed with the following steps:

Step 1. Type the command CHGJRN(QUSRSYS/AMQAJRN) and press Enter.

This displays the panel shown in Figure 21.

```

Change Journal (CHGJRN)

Type choices, press Enter.

Journal > AMQAJRN Name
Library > QUSRSYS Name, *LIBL, *CURLIB
Journal receiver:
 Journal receiver *SAME Name, *SAME, *GEN
 Library Name, *LIBL, *CURLIB
 Journal receiver Name, *GEN
 Library Name, *LIBL, *CURLIB
Sequence option *CONT *RESET, *CONT
Journal message queue AMQAJRNMSG Name, *SAME
 Library QMQMDATA Name, *LIBL, *CURLIB
Manage receivers *USER *SAME, *USER, *SYSTEM
Delete receivers *NO *SAME, *NO, *YES
Receiver size options *NONE *SAME, *NONE, *RMVINTENT...

Journal state *SAME *SAME, *ACTIVE, *INACTIVE

More...
```

Figure 21. MQSeries for AS/400 Changing to System Managed Journal Receivers

Step 2. Change the "Manage Receivers" parameter from \*USER to \*SYSTEM.

Step 3. Also, make sure that the "Delete Receivers" parameter is set to \*NO. This is the system default.

Step 4. Press Enter to accept the new changes for the parameters.

### 3.1.4.4 Deleting Journal Receivers

We recommended not to let the system delete any receivers (by specifying \*NO as DLTRCV parameter in AMQAJRN) and let the operator do it.

Journal receivers must be deleted in the same order in which they were attached. An easy way to see the receiver chain is to print it out. Issue the command:

```
WRKJRNA JRN(QUSRSYS/AMQAJRN) OUTPUT(*PRINT)
```

Next, use WRKJOB from the "Work with Job" menu. Choose option 4 and select the spooled output. The printer file created is QPDSPJNA. Displaying

this spooled file, you should be able to see a section headed "Receiver Directory". You will see the order of the chain as displayed sequentially in the column headed "NUMBER". Another option is given in 3.4.6, "Journal Receiver Chain" on page 67. Here you can view the receiver chains.

**Note:** You cannot delete a journal receiver that is attached to a local journal.

---

## 3.2 MQSeries for AS/400 Backup Procedures

Backing up MQSeries enables us to restore the system to a known point. We all know the importance of a good backup strategy. This topic discusses what objects have to be backed up and how we can perform the backup.

### 3.2.1 Where Things Are Located

All objects in MQSeries are located in the following libraries:

|                 |                                                   |
|-----------------|---------------------------------------------------|
| <i>QMQM</i>     | Base product library                              |
| <i>QMQMADM</i>  | Product library, admin application (optional)     |
| <i>QMQMSAMP</i> | Product library, samples (optional)               |
| <i>QMQMPROC</i> | Data library, processes                           |
| <i>QMQMDATA</i> | Data library                                      |
| <i>QUSRSYS</i>  | OS/400 library, contains local and remote journal |

### 3.2.2 Backing Up MQSeries for OS/400 Objects

This involves saving the following libraries or objects:

1. AMQAJRN and AMQRJRN journal objects in QUSRSYS library
2. QMQMPROC
3. QMQMDATA

The following are product libraries:

- QMQM
- QMQMADM
- QMQMSAMP

If you have not created any user objects in these libraries you can restore them by reinstalling the MQSeries product. The install is done using an OS/400 install process invoked by GO LICPGM and selecting option 11 (install).

## AS/400

Objects in QMQMDATA library are grouped into two broad categories:

1. Channel objects, which consist of:
  - Channel definitions (AMQRFCD4 file)
  - TCP/IP configuration (QMINI file)
  - Channel synchronization (AMQRSYNA file)
2. Queue objects, which consist of:
  - Queue definitions (QMCMOBYCAT OBJTYPE(\*USRIDX) and all the user spaces that have queue names in their definitions)
  - Persistent message data (AMQAJRN and its corresponding receivers)

Objects in QMQMPROC library consist of PROCESS objects.

**Note:** You can view the files in QMQMDATA and QMQMPROC using:

```
WRKOBJPDM LIB(QMQMDATA)
WRKOBJPDM LIB(QMQMPROC)
```

### 3.2.2.1 Procedure to Back Up Data Libraries

The idea behind these procedures is to ensure consistency during the backup process. The steps are:

- Step 1. End the MQ channels with one of the following commands:

```
WRKMQMCHL (then choose option 15)
ENDMQMCHL CHLNAME(some name) OPTION(*IMMED)
```
- Step 2. Exit the Administrator Utility if running.
- Step 3. End the Command Server with the command:

```
ENDMQMCSVR MQMNAME(QMNAME) Option(*IMMED)
```
- Step 4. Reset the media recovery point with the command:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL)
```
- Step 5. End the queue manager:

```
ENDMQM MQMNAME(queue manager name) OPTION(*IMMED)
```
- Step 6. Sign off and sign on again.
- Step 7. Restart the queue manager with STRMQM.
- Step 8. Display the QSYSOPR message queue with DSPMSG QSYSOPR and look for AMQ7460 and AMQ7462 messages. Write down the receivers' numbers in these messages.

Step 9. Issue WRKOBJPDM LIB(QMQMDATA). Delete the unwanted journal receivers with the help of AMQ7462 and AMQ7460 messages. For example, if AMQ7462 indicates receiver, AMQA000050 and AMQ7460 indicate receiver AMQA000054, then delete any receiver before AMQA000050. This will help reduce the number of unwanted journal receivers present in QMQMDATA prior to backing up QMQMDATA.

Further information on this aspect of journal receiver management can be found in 3.1.3.1, "Ensuring the Required Receivers Are Present" on page 40.

**Note:** During deletion, you may get the following message:

```
CPA7025 Receiver AMQA000042 in QMQMDATA never fully saved. (I C)
Enter "I" to ignore it.
```

Also, journal receivers must be deleted in the same order in which they were attached. Additional information is given in 3.1.4.4, "Deleting Journal Receivers" on page 46.

Step 10. Quiesce MQSeries: CALL QMQM/AMQIQES4.

Step 11. Tidy up the cache storage: CALL QMQM/AMQIQEM4.

Step 12. Save the local and remote journals:

```
SAVOBJ OBJ(AMQAJRN AMQRJRN) LIB(QUSRSYS) OBJTYPE(*JRN)
```

Step 13. Save the QMQMPROC library: SAVLIB LIB(QMQMPROC).

Step 14. Save the QMQMDATA library: SAVLIB LIB(QMQMDATA).

**Note:** It is a good practice to display the joblog of the save to make sure that all the objects are saved.

Step 15. This ends the backup procedure.

---

### 3.3 MQSeries for AS/400 Recovery Scenarios

In this topic there are several scenarios for recovering MQSeries. The scenarios are listed below:

- System failure, total system loss
  - Scenario 1: Disk crash
  - Scenario 2: All objects lost except journal receivers
  - Scenario 3: Queue manager is damaged
  - Scenario 4: Queue manager is deleted

## AS/400

It is worth noting that in these scenarios we can recover up to the point of the last backup. It is likely to mean that some messages will be lost as the journal receivers will not contain any record of transactions between last backup and crash.

- System failure, no data loss
  - Scenario 5: MQSeries system jobs ended abnormally, cannot restart the queue manager
- Restoring MQSeries for AS/400 into a new or backup system
  - Scenario 6: Moving MQSeries to another system on the same OS/400 release
- Object damage
  - Scenario 7: Damage to a queue object

The procedures to recover from the above situations are listed in the sections below:

### 3.3.1 Scenario 1: Disk Crash

Here we have a situation where we have lost everything on the system. This is akin to a disk crash where for some reason we have to install the system from scratch.

In this situation, the recovery steps are:

- Step 1. Re-install the MQSeries for AS/400 licensed program products. The steps to do this are:
- a. From G0 LICPGM, option 12, locate the MQSeries for AS/400 licensed program and delete it. To ensure a clean deletion, the following conditions must exist:
    - The user must be enrolled in the AS/400 system directory. This is done using the WRKDIRE command.
    - The queue manager has ended and quiesced and cache storage purged:

```
ENDMQM MQMNAME(queue manager name) Option(*IMMED)
CALL QMQM/AMQIQES4
CALL QMQM/AMQIQEM4
```
  - b. Use G0 LICPGM and select option 10 (Display Installed Licensed Programs) to make sure that the product is properly deleted. It is prudent to keep checking the job log to make sure that there are no problems.



- c. Use GO LICPGM and select option 11 (Install Licensed Programs). Locate the MQSeries for AS/400 products and proceed with installation.
  - d. Apply the latest cum tapes or any other PTFs for this product if you are using the distribution media.
- Step 2. Verify the currently available journal receivers that the system requires for restart and media recovery.

**Note:** This could be obtained through some form of manual documentation or through displaying the saved media. In our situation, we found that our journal receivers AMQA000054 and AMQA000055 represent our incremental backup. In the system backup (SAVLIB) we have the receivers AMQA000050 through 54, where AMQA000054 is only partially saved.

- Step 3. Issue CLRLIB LIB(QMQMDATA) to clear the QMQMDATA library.

**Note:** Failure to do so may result in objects such as AMQRFCD4, AMQRYSNA, QMINI, AMQRJRNMSG not being restored as a result of ownership or level check problems with these files.

- Step 4. Restore the local and remote journals from QUSRSYS with the command:

```
RSTOBJ OBJ(AMQAJRN AMQRJRN) SAVLIB(QUSRSYS)
```

**Note:** In our scenario, the system generates a new receiver, AMQA100054, during the restore of the local journal. It also generates AMQR100017 during the restore of the remote journal.

#### Restore Order

The order of the restore is important:

- Journals
- Based-on physical files
- Dependent logical files
- Journal receivers

- Step 5. Issuing WRKJRNA JRN(QUSRSYS/AMQAJRN) shows that the attached receiver is AMQA100054. Pressing F15 gives us the receiver directory. Also, for AMQRJRN, the current attached receiver is AMQR100018. Further information on receiver chains can be found in 3.4.6, "Journal Receiver Chain" on page 67.

## AS/400

- Step 6. Before restoring the data libraries, restore any incremental journal receivers since the last SAVLIB of QMQMDATA. You may have these backed up on a different save media.

**Note:** This will ensure that the most current receivers are restored first. This is helpful in situations where the same receiver exists in both your system save (SAVLIB) and in the incremental (and separate) save of the receivers. This could arise because the journal receiver is attached at the time of the system save. This same receiver is subsequently detached when the incremental save is performed.

- Step 7. Restore the data libraries using:

```
RSTLIB SAVLIB(QMQMDATA)
RSTLIB SAVLIB(QMQMPROC)
```

- Step 8. To keep to our naming convention for receivers, we decided to create a new local receiver, here AMQA000055. This is in preference to going along with the system created one, AMQA100054. The following commands will do that:

```
CRTJRNRCV JRNRCV(QMQMDATA/AMQA000055) THRESHOLD(17000)
CHGOBJOWN OBJ(QMQMDATA/AMQA000055) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
CHGJRN JRN(QUSRSYS/AMQAJRN) JRNRCV(QMQMDATA/AMQA000055)
MSGQ(QMQMDATA/AMQAJRNMSG) MNGRCV(*SYSTEM) DLTRCV(*NO)
```

The result of this is that journal receiver AMQA100054 is now detached.

- Step 9. Repeat the above steps for the remote receivers:

```
CRTJRNRCV JRNRCV(QMQMDATA/AMQR000019) THRESHOLD(17000)
CHGOBJOWN OBJ(QMQMDATA/AMQR000019) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
CHGJRN JRN(QUSRSYS/AMQRJRN) JRNRCV(QMQMDATA/AMQR000019)
MSGQ(QMQMDATA/AMQRJRNMSG) MNGRCV(*SYSTEM) DLTRCV(*NO)
```

The result of this is that journal receiver AMQR100018 is now detached.

- Step 10. DO NOT delete the detached receivers from the system.

**Note:** This is because they are now part of the receiver chain. MQSeries uses these receivers at startup.

- Step 11. Associate the receivers with the restored journals. Use WRKJRN and select option 9 to create the association.

- Step 12. Check for attached receivers and journaled physical file (AMQRSYNA). Issue commands:

```
WRKJRNA JRN(QUSRSYS/AMQAJRN)
WRKJRNA JRN(QUSRSYS/AMQAJRN)
```

Use F13 to show a list of journaled files in the case of AMQRJRN. If AMQRSYNA is not in that list then invoke it using:

```
STRJRNP FILE(QMQMDATA/AMQRSYNA) JRN(AMQRJRN)
```

You can also see the attributes (option 8) and the list of attached receivers (F15).

- Step 13. Sign off and sign on again.
- Step 14. Start the queue manager with STRMQM.
- Step 15. Verify the success of the operation. This includes queues, channels, authorizations, start listener and start channel initiator.
- Step 16. Reset the channel at the sender end to overcome message sequence number errors.
- Step 17. Execute DSPMSG QSYSOPR and note the AMQ7460 startup and AMQ7462 media recovery information.
- Step 18. To advance the startup and recovery points for the local receivers issue this command:
- ```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL)
```
- Step 19. This ends our recovery procedure.

3.3.2 Scenario 2: QMQMDATA Contains Only Journal Receivers

This is a situation where the journal receivers are the only objects present on the system at the time of loss or inadvertent deletion. The recovery steps are:

- Step 1. Display the contents of the QMQMDATA and QMQMPROC libraries:

```
WRKOBJPDM LIB(QMQMDATA)
WRKOBJPDM LIB(QMQMPROC)
```

- Step 2. Clear all the objects in QMQMPROC: `CLRLIB LIB(QMQMPROC)`.

Note: Our testing indicates that we need to also restore all objects from the save of library QMQMPROC. This will minimize damage to these objects upon completion of this particular recovery procedure.

- Step 3. Restore all the objects from the QMQMDATA and QMQMPROC libraries:

```
RSTOBJ OBJ(*ALL) SAVLIB(QMQMPROC)
RSTOBJ OBJ(*ALL) SAVLIB(QMQMDATA)
```

AS/400

Step 4. Next, issue the following commands:

```
WRKJRNA JRN(QUSRSYS/AMQAJRN) OUTPUT(*PRINT)
WRKJRNA JRN(QUSRSYS/AMQRJRN) OUTPUT(*PRINT)
```

Then check in the listing if the receiver chain is still intact. Also, verify that the AMQRSYNA file is currently being journaled to remote journal AMQRJRN.

Step 5. End the queue manager with:

```
ENDMQM MQMNAME(RALYAS4C) OPTION(*IMMED)
```

Step 6. Quiesce MQSeries with CALL QMQM/AMQIQES4.

Step 7. Purge the cache storage with CALL QMQM/AMQIQEM4.

Step 8. Sign off and sign on.

Step 9. Restart the queue manager with STRMQM.

Step 10. Verify the successful operation of MQSeries. This includes queues, channels, authorizations, starting the listener, and channel initiator, etc.

Step 11. Record the media image with the following command:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL)
```

Step 12. This completes the recovery procedure for this scenario.

3.3.3 Scenario 3: Queue Manager Is Damaged

In this scenario, the queue manager becomes damaged. The result is that we are unable to start the queue manager. The recovery steps are:

Step 1. Restore the journals from library QUSRSYS:

```
RSTOBJ OBJ(AMQAJRN AMQRJRN) SAVLIB(QUSRSYS)
```

Note: During this restore of journals, the system will generate a new receiver number for both the local and remote journals. For example, it may generate AMQA100041 and AMQR1000018 for local and remote channels respectively.

Restore Order

The order of restore is important:

1. Journals
2. Based-on physical files
3. Dependent logical files
4. Journal receivers

- Step 2. Restore the data libraries for MQSeries:
RSTLIB SAVLIB(QMQMDATA) & RSTLIB SAVLIB(QMQMPROC)
- Step 3. Issue WRKJRNA JRN(QUSRSYS/AMQAJRN), and you can see that the attached receiver in our example is AMQA100041. As expected, the system has attached this new receiver to the journal.
- Step 4. To keep to our naming convention for receivers, we decided to create a new local receiver, AMQA000042. This is in preference to going along with the system created one, namely AMQA100041:
CRTJRNRCV JRNRCV(QMQMDATA/AMQA000042) THRESHOLD(17000)
CHGOBJOWN OBJ(QMQMDATA/AMQA000042) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
CHGJRN JRN(QUSRSYS/AMQAJRN) JRNRCV(QMQMDATA/AMQA000042)
- The result of this is that journal receiver AMQA100041 is detached.
- Step 5. We repeat the above steps for the remote receiver:
CRTJRNRCV JRNRCV(QMQMDATA/AMQR000018) THRESHOLD(17000)
CHGOBJOWN OBJ(QMQMDATA/AMQR000018) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
CHGJRN JRN(QUSRSYS/AMQAJRN) JRNRCV(QMQMDATA/AMQR000018)
MSGQ(QMQMDATA/AMQRJRNMSG)
- The result of this is that journal receiver AMQR100017 is detached.
- Step 6. Check for attached receivers and journaled physical files (AMQRSYNA) with the following commands:
WRKJRNA JRN(QUSRSYS/AMQAJRN)
WRKJRNA JRN(QUSRSYS/AMQRJRN)
- Use F13 to view a list of journaled files in the case of AMQRJRN. If AMQRSYNA is not in that list then execute:
STRJRNPF FILE(QMQMDATA/AMQRSYNA) JRN(AMQRJRN)
- You can also see the attributes (option 8) and the list of attached receivers (F15).
- Step 7. Associate the receivers with the restored journals with WRKJRN, option 9, to perform the association.
- Step 8. Sign off and sign on.
- Step 9. Start the queue manager with STRMQM.
- Step 10. Verify the success of the operation. This includes queues, channels, authorizations, start listener, start channel initiator, etc.
- Step 11. Issue RCDMQMIMG to advance the startup and recovery points for the receivers.

Note: You will see in WRKJRNA, F15, that there are more receivers in the chain after the RCDMQMIMG command. Look at DSPMSG

AS/400

QSYSOPR to check the new MQ startup journal information AMQ7460 and MQM media recovery journal information AMQ7462.

Step 12. This ends the procedure.

3.3.4 Scenario 4: Queue Manager Is Deleted

In this scenario, we assume that someone has inadvertently deleted the queue manager. During the process of deleting the queue manager, the local receivers and journal are automatically deleted. To fix the problem you have two options:

3.3.4.1 Option 1: Restoring All Objects from QMQMDATA

The easiest way is generally to restore the QMQMDATA library. We will describe this procedure in detail. The steps are:

Step 1. Delete the remote journal from QUSRSYS library:

```
DLTJRN JRN(QUSRSYS/AMQRJRN)
```

Step 2. Delete all the objects in the QMQMDATA library:

```
CLRLIB LIB(QMQMDATA)
```

Step 3. Restore both the local and remote journals:

```
RSTOBJ OBJ(AMQAJRN AMQRJRN) SAVLIB(QUSRSYS)
```

Note: The system generates new receivers during the restore. In our case they are AMQAI00065 (local) and AMQRI00019 (remote).

Step 4. Restore any incremental journal receivers since the last system backup of QMQMDATA.

Step 5. Restore QMQMDATA library: RSTLIB LIB(QMQMDATA).

Note: If you have initially restored, say, AMQA000064 as a result of restoring the incremental receiver first, the restore of QMQMDATA will not restore the same receiver. This is expected.

Step 6. To keep to our own naming convention, we decided to create a new local receiver, AMQA000066, and new remote receiver, AMQR000019.

```
CRTJRNRCV JRNRCV(QMQMDATA/AMQA000066) THRESHOLD(17000)
CHGOBJOWN OBJ(QMQMDATA/AMQA000066) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
CHGJRN JRN(QUSRSYS/AMQAJRN) JRNRCV(QMQMDATA/AMQA000066)
MSGQ(QMQMDATA/AMQAJRNMSG)
```

Step 7. We repeat the above steps for the remote receiver:

```

CRTJRNRCV JRNRCV(QMQMDATA/AMQI000019) THRESHOLD(17000)
CHGOBJOWN OBJ(QMQMDATA/AMQI000019) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
CHGJRN JRN(QUSRSYS/AMQRJRN) JRNRCV(QMQMDATA/AMQA000066)
MSGQ(QMQMDATA/AMQRJRNMSG)

```

- Step 8. Associate the receivers with the restored journals. Use WRKJRN, option 9, to perform the association.
- Step 9. Check for attached receivers and journaled physical file (AMQRSYNA). Issue commands:
- ```

WRKJRNA JRN(QUSRSYS/AMQAJRN)
WRKJRNA JRN(QUSRSYS/AMQRJRN)

```
- Use F13 to show a list of journaled files in the case of AMQRJRN, and if AMQRSYNA is not in that list then invoke it using:
- ```

STRJRNP FILE(QMQMDATA/AMQRSYNA) JRN(AMQRJRN)

```
- Step 10. Sign off and sign on.
- Step 11. Start the queue manager with STRMQM.
- Step 12. Verify the successful operation of MQSeries. This includes queues, channels, authorizations, starting the listener, starting the channel initiator, etc.
- Step 13. Issue RCDQMIMG to advance the startup and recovery points for the local journal receiver (AMQA*****).
- Step 14. This ends the procedure.

3.3.4.2 Option 2: Restoring Specific Objects

Another restore option for a deleted queue manager is to restore specific objects back to the QMQMDATA library. We recommend using 3.3.4.1, "Option 1: Restoring All Objects from QMQMDATA" on page 56. We show a second option to provide another alternative for the reader. The steps are:

- Step 1. End the queue manager with the command:
- ```

ENDMQM MQMNAME(RALYAS4A) OPTION(*IMMED)

```
- Step 2. Quiesce MQSeries with CALL QMQM/AMQIQES4.
- Step 3. Purge the cache storage with CALL QMQM/AMQIQEM4.
- Step 4. Sign off and sign on.
- Step 5. Delete QMQMDATA of all objects except:
- QMINI
  - AMQRJRNMSG
  - AMQRxxxxxx

## AS/400

- AMQRSYNA
- AMQRFCD4

Step 6. Sign off and sign on.

Step 7. Create the queue manager manually with CRTMQM and press Enter.

**Note:** This process will create a local journal and local receiver, AMQAJRN and AMQA000000. However, we want to restore our saved journal and journal receivers from the backup. This leads to the next step.

Step 8. Delete the system created AMQAJRN and the AMQA000000 receiver.

```
DLTJRN JRN(QUSRSYS/AMQAJRN)
DLTJRNRCV JRNRCV(QMQMDATA/AMQA000000)
```

Step 9. To get the persistent messages, restore AMQAJRN journals and all the incremental receivers and all prior receivers from your saves of QMQMDATA. In our example, we have receivers AMQA000060 to AMQA000066:

```
RSTOBJ OBJ(AMQAJRN) SAVLIB(QUSRSYS)
RSTOBJ OBJ(AMQA00006*) SAVLIB(QMQMDATA)
```

**Note:** You need to know the range of receivers to restore. We assume that you have noted the AMQ7460 and AMQ7462 messages specifying the receivers required for startup and media recovery here.

Step 10. Restore the queue definitions and user space index object:

```
RSTOBJ OBJ(*ALL) SAVLIB(QMQMDATA) OBJTYPE(*USRSPC *USRIDX)
```

Step 11. To keep to our own naming convention, we decided to create a new local receiver AMQA000066:

```
CRTJRNRCV JRNRCV(QMQMDATA/AMQA000066) THRESHOLD(17000)
CHGOBJOWN OBJ(QMQMDATA/AMQA000066) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
CHGJRN JRN(QUSRSYS/AMQAJRN) JRNRCV(QMQMDATA/AMQA000066)
MSGQ(QMQMDATA/AMQAJRNMSG)
```

Step 12. Associate the receivers with the restored local journal (AMQAJRN). Use option 9 of WRKJRN to perform the association.

Step 13. Check for attached receivers in the local AMQAJRN journal:

```
WRKJRNA JRN(QUSRSYS/AMQAJRN)
```

Use F15 to view the receiver chains.

Step 14. End the queue manager with the command:

```
ENDMQM MQMNAME(RALYAS4C) OPTION(*IMMED)
```



- Step 15. Quiese MQSeries with CALL QMQM/AMQIQES4.
- Step 16. Tidy up the shared storage with CALL QMQM/AMQIQEM4.
- Step 17. Sign off and sign on.
- Step 18. Restart the queue manager with STRMQM.
- Step 19. Verify the successful operation of MQSeries. This includes queues, channels, authorisations, starting the listener, starting the channel initiator, etc.
- Step 20. Issue RCDMQMIMG to advance the startup and recovery points for AMQA\* receivers.
- Step 21. This ends the procedure

### 3.3.5 Scenario 5: System Failure, No Data Loss

This scenario arises in a situation where someone inadvertently ends the AMQXIHK4 job or for some reason it may have ended due to a system failure. The symptom is the inability to start or end the queue manager.

To test this scenario, we had to create an environment where one is unable to start or end the queue manager. To simulate this situation, we crashed the AMQXIHK4 job by ending it via ENDJOB \*IMMED.

#### 3.3.5.1 Crash the System

The environment can be created the following way:

1. End the AMQXIHK4 jobs from WRKACTJOB SBS(QSYSWRK).

The joblog indicates: ENDJOB started for job 004659/QMQM/QMQM

2. Now end the queue manager with the command:

```
ENDMQM MQMNAME(RALYAS4C) OPTION(*IMMED)
```

The joblog indicates: CPF0001 Error found on ENDMQM command.

3. Start the queue manager with STRMQM .

The joblog indicates: AMQ8101 Unexpected error.

Cause . . . . . : An unexpected reason code with hexadecimal value X'00000893' was received from the message queue manager during command processing.

This is equated to MQI return code 2195 MQRC\_UNEXPECTED\_ERROR.

4. If you issue WRKMQMQ, you get this error in the joblog:

```
CPF8146 Message Queue Manager not available.
```

### 3.3.5.2 How to Recover from System Failure

To recover from such a situation take the following steps:

Step 1. Run the quiesce program `CALL QMQM/AMQIQES4`.

**Note:** While running this quiesce program we get the error AMQ6910. The quiesce of the queue manager failed because the current job has locks on the library QMQM. We then use the recovery for this message which is to sign off the current job and sign on again. Repeating the same command again, it seemed to work correctly. This is shown in the following two steps.

Step 2. Sign off the current job and sign on again.

Step 3. Re-run the same program: `CALL QMQM/AMQIQES4`. This works correctly.

Step 4. Run the program to tidy up the cache storage: `CALL QMQM/AMQIQEM4`.

Step 5. Sign off the current job and sign on again after running AMQIQEM4 as the user space for your current job needs to be purged as well.

Step 6. Restart the queue manager with `STRMQM`.

Step 7. Verify the successful operation of MQSeries. This includes queues, channels, authorizations, starting the listener, starting the channel initiator, etc.

Step 8. This ends the procedure.

### 3.3.6 Scenario 6: Restoring MQSeries into Another AS/400 System

This could occur if for some reason you want to move your MQSeries objects into a new or backup AS/400 system. In our scenario, the same queue manager name is used for the new or backup system. We have tested this scenario on the same release of the OS/400 in both machines.

#### 3.3.6.1 How to Save MQSeries Objects

For completeness, we will reiterate the steps to save the MQSeries objects:

Step 1. End all channels: `ENDMQMCHL`

Step 2. End the Administration Utility: `ENDMQMADM`

Step 3. End the Command Server: `ENDMQMCSVR`

Step 4. Issue Record Media Image:  
`RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL)`

Message AMQ8034, Media Image Recorded will be seen in the joblog.

Step 5. End the queue manager:

```
ENDMQM MQMNAME(RALYAS4C) OPTION(*IMMED)
```

Step 6. Quiese MQSeries: CALL QMQM/AMQIQEJ4

Step 7. Tidy up the shared storage: CALL QMQM/AMQIQEM4

**Note:** When we first issued this command we got AMQ6909, saying user space recovery failed, MQM is running. We suspect that this is a timing problem. After a few minutes we issued the command again and it was successful.

Step 8. Sign off and sign on again.

**Note:** When we attempted an STRMQM without signing off and on, we got AMQ8101, unexpected error. Signing off and on has the effect of purging pointers from QMQMDATA and cleaning up the user space for the job issuing the AMQIQEM4 program. The idea here is to sign off and on after running AMQIQEM4.

Step 9. Start the queue manager: STRMQM

Step 10. Display the QSYSOPR messages: DSPMSG QSYSOPR.

We need to take note of the startup journal information AMQ7460 and media recovery journal information AMQ7462.

Step 11. From information gathered via the AMQ7462 and AMQ7460 messages, we delete unwanted journal receivers. In our case, AMQ7462 points to receiver AQMA000037 and AMQ7460 points to receiver AMQA000041. Based on this information, we delete receivers AMQA000028 to AMQA000036 from the system.

**Note:** While deleting these receivers you may get messages such as "CPA7025 Receiver AMQA000028 in QMQMDATA never fully saved". Enter an I to ignore.

Step 12. Save the QMQMDATA and QMQMPROC libraries:

```
SAVLIB LIB(QMQMDATA QMQMPROC)
```

Step 13. Save the local and remote journals:

```
SAVOBJ OBJ(QUSRSYS/AMQAJRN)
SAVOBJ OBJ(QUSRSYS/AMQRJRN)
```

Step 14. This brings us to the end of the SAVE procedure.

### 3.3.6.2 How to Restore MQSeries Objects

We will now proceed to restore to new (different) system. In this scenario, we keep the same queue manager name. This will at least ensure not having to change queue manager names of remote queue definition in other machines communicating with this new or backup machine. The steps to do this are:

- Step 1. Sign on to the new system using the QSECOFR profile.
- Step 2. From GO LICPGM, option 11 (Install Licensed Programs), locate MQSeries for AS/400, and proceed with the installation.
- Step 3. Apply the latest cum tapes or any other PTFs for this product if you are using the distribution media.
- Step 4. Create the queue manager manually:  
CRTMQM MQMNAME(RALYAS4A) UDLMMSGQ(RALYAS4C.DEADQ)
- Step 5. Start the queue manager: STRMQM  
**Note:** This is to test that all is well before we go ahead with the restore.
- Step 6. Delete journals AMQAJRN and AMQRJRN from library QUSRSYS:  
DLTJRN JRN(QUSRSYS/AMQ\*)
- Step 7. Clear QMQMDATA and QMQMPROC libraries:  
CLRLIB LIB(QMQMDATA)  
CLRLIB LIB(QMQMPROC)
- Step 8. Restore the journal objects first:  
RSTOBJ OBJ(AMQAJRN AMQRJRN) SAVLIB(QUSRSYS)  
**Note:** In our scenario, the system generates a new receiver, AMQA100041, during the restore of the local journal. It also generates AMQR100017 during the restore of the remote journal.

#### Restore Order

The order of the restore is important:

1. Journals
2. Based-on physical files
3. Dependent logical files
4. Journal receivers

- Step 9. Issue WRKJRNA JRN(QUSRSYS/AMQAJRN)

We can see that the attached receiver is AMQ100041. The system has attached this new receiver to the journal.

- Step 10. To keep to our naming convention for receivers, we decided to create a new receiver, AMQA00042, and to detach the system created one, in our case AMQA100041.

```
CRTJRNRCV JRNRCV(QMQMDATA/AMQA000042) THRESHOLD(17000)
CHGOBJOWN OBJ(QMQMDATA/AMQA000042) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
CHGJRN JRN(QUSRSYS/AMQAJRN) JRNRCV(QMQMDATA/AMQA000042)
MSGQ(QMQMDATA/AMQAJRNMSG)
```

The result of this is that the journal receiver AMQA100041 is detached.

- Step 11. We repeat the above steps for the remote journal receiver:

```
CRTJRNRCV JRNRCV(QMQMDATA/AMQR000018) THRESHOLD(17000)
CHGOBJOWN OBJ(QMQMDATA/AMQR000018) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
CHGJRN JRN(QUSRSYS/AMQRJRN) JRNRCV(QMQMDATA/AMQR000018)
MSGQ(QMQMDATA/AMQRJRNMSG)
```

The result of this is that the journal receiver AMQR100017 is detached.

- Step 12. DO NOT delete the detached receivers AMQA100041 and AMQR100017 on the system.

**Note:** This is because they now form part of the receiver chain. MQSeries startup uses this chain to replay the receivers.

- Step 13. We restore the QMQMDATA and QMQMPROC data libraries:

```
RSTLIB LIB(QMQMDATA)
RSTLIB LIB(QMQMPROC)
```

- Step 14. Associate the receivers with the restored local and remote journals. Use option 9 of WRKJRN to perform the association.

- Step 15. Check for attached receivers and journaled physical files (AMQRSYNA). Issue commands:

```
WRKJRNA JRN(QUSRSYS/AMQAJRN)
WRKJRNA JRN(QUSRSYS/AMQRJRN)
```

Use F13 to show a list of journaled files in the case of AMQRJRN; and if AMQRSYNA is not in that list then invoke it using:

```
STRJRNP FILE(QMQMDATA/AMQRSYNA) JRN(AMQRJRN)
```

F15 will show the receiver directory entries.

- Step 16. Sign off and sign on again to purge the pointers from QMQMDATA.

- Step 17. Start the queue manager with STRMQM.

## AS/400

Step 18. Verify the successful restore operation of MQSeries. This includes queues, channels, authorizations, starting the listener, starting the channel initiator, etc.

Step 19. Issue RCDMQMIMG to advance the startup and recovery points for the receivers.

**Note:** You will see via the WRKJRNA F15 key, that there are more receivers on the chain after the RCDMQMIMG command. Issue a DSPMSG QSYSOPR to check the the new MQM Startup journal information AMQ7460 and MQM media recovery journal information AMQ7462.

Step 20. This ends the procedure.

### 3.3.7 Scenario 7: Object Damage

This situation is more commonly presented in the form of queue damage.

A typical message that you will get when a queue is damaged is illustrated in Figure 22 on page 65.

In this situation the recovery steps are:

Step 1. Use the Re-create MQM Object command:

```
RCRMQMOBJ OBJ(OTHER) OBJTYPE(*Q)
```

If successful, you will get a new AMQ8033- MQM object.

Step 2. Sign off and sign on again.

**Note:** This will only work if the receiver chains are intact and all receivers required for playback are either on the system or backed up in offline storage. If the receiver cannot be found, then the only choice is to re-create the object manually from your planning sheet or restore from a previous backup copy.

---

## 3.4 Useful Hints for MQSeries Backup and Recovery

In the context of MQSeries backup and recovery, it will be useful to understand some of the commands used.

### 3.4.1 RCRMQMIMG

The Record MQM Object Image (RCDMQMIMG) command provides a marker for the selected set of MQM objects, so that the Re-create MQM Object (RCRMQMOBJ) command can recover this set of objects from journal data recorded earlier.

```

Additional Message Information
Message ID : AMQ7472 Severity : 10
Message type : Information
Date sent : 06/25/98 Time sent : 13:52:02

Message : Object OTHER, type queue damaged.
Cause : Object OTHER, type queue has been marked as damaged. This
indicates that the queue manager was either unable to access the object in
the file system, or that some kind of inconsistency with the data in the
object was detected.
Recovery : If a damaged object is detected, the action performed
depends on whether the queue manager supports media recovery and when the
damage was detected. If the queue manager does not support media recovery,
you must delete the object as no recovery is possible. If the queue manager
does support media recovery and the damage is detected during the processing
performed when the queue manager is being started, the queue manager will
automatically initiate media recovery of the object. If the queue manager
More...

Press Enter to continue.

F3=Exit F6=Print F9=Display message details F12=Cancel
F21=Select assistance level

=====

Additional Message Information

Message ID : AMQ7472 Severity : 10
Message type : Information

supports media recovery and the damage is detected once the queue manager
has started, it may be recovered from a media image using the rcrmqobj
command or it may be deleted.
Technical Description : None.

```

Figure 22. MQSeries for AS/400 Damaged Object

This command is intended to enable journal receivers, detached prior to the current date, to be disconnected. On successful completion of this command those journals are no longer required to be present for a Re-create MQM Object (RCRMQMOBJ) command on this set of MQM objects to succeed.

It does the following:

- It advances the next AMQ7462 restart receiver.
- It forces an MQSeries journal checkpoint.

## AS/400

- It resets the media recovery point.

### 3.4.2 Size of Journal Receivers

The smaller the threshold value is for journal receivers, the more often journal receiver switching takes place. This has a direct impact on performance. You may find that STRMQM will take a longer time to complete. Increasing the threshold size from a default of 16 KB to a much higher number will help to reduce the number of journal receivers in library QMQMDATA.

### 3.4.3 Sign Off and Sign On before STRMQM

In the recovery procedures for MQSeries you will find us recommending signing off and signing back on the AS/400 before restarting the queue manager. This has the effect of making sure pointers to these QMQMDATA objects are purged before restarting the queue manager. In fact, our own testing indicated that doing this is essential to enable us to recover back to a known point.

### 3.4.4 Quiescing the AS/400

Quiesce is the orderly shutdown of MQSeries. There is a set of CL commands that would result in an orderly shutdown of MQSeries. Unpredictable results can occur if these shutdown procedures are not employed. Below is the recommended shutdown procedure for MQSeries for AS/400:

1. ENDMQMCSVR
2. ENDMQM \*CNTRLD
3. DLY
4. CALL QMQM/AMQIQEJ4
5. CALL QMQM/AMQIQES4
6. CALL QMQM/AMQIQEM4  
(Does not quiesce, only cleans up shared storage.)

**Note:** You can also code the above procedure into a CL program.

### 3.4.5 Shared Memory

When a queue manager is active, some shared memory or storage is allocated for queue manager status and some is allocated for each MQ job. The CL program AMQIQEM4 has the effect of tidying or cleaning up the shared storage spaces left behind. This is done before restarting the queue manager.



### 3.4.6 Journal Receiver Chain

All local AMQA\* journal receivers associated with the AMQAJRN journals are linked to one or more chains. Each journal receiver, except the first one, has a previous counterpart that was detached when the current receiver was attached. Figure 23 illustrates the concept.

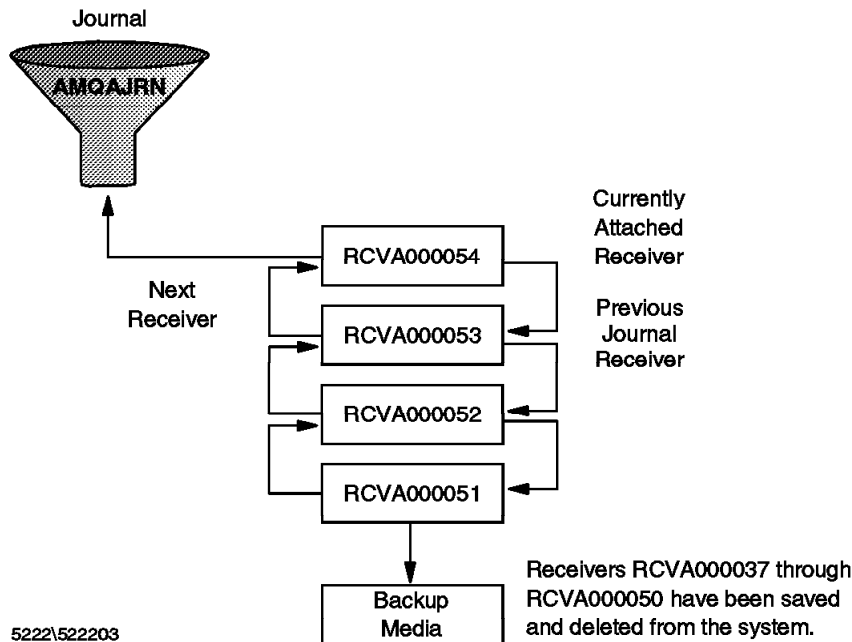


Figure 23. MQSeries for AS/400 Display Journal Receiver Chain

Below is an example of how you can work through the receiver chains.

1. Issue `WRKJRNA JRN(QUSRSYS/AMQAJRN)`.

The panel shown in Figure 20 on page 44 illustrates the point.

```

 Work with Journal Attributes

Journal : AMQAJRN Library : QUSRSYS

Auxiliary storage
 pool : 1 Journal type : *LOCAL
Message queue . . . : AMQAJRNMSG Journal state : *ACTIVE
 Library : QMQMDATA Receiver size options: *NONE
Manage receivers . . : *SYSTEM
Delete receivers . . : *NO
Text : *BLANK

Type options, press Enter.
 8=Display attributes

 Attached
Option Receiver Library
 AMQA000054 QMQMDATA

Bottom

F3=Exit F5=Refresh F12=Cancel F13=Display journaled files
F14=Display journaled access paths F24=More keys

```

Figure 24. Work with Journal Attributes Menu

```

 Work with Receiver Directory

Journal : AMQAJRN Library : QUSRSYS

Total size of receivers : 255664128

Type options, press Enter.
 4=Delete 8=Display attributes

Opt Receiver Library Number Attach Status Save
 8 AMQA000042 QMQMDATA 00001 06/24/98 ONLINE 00/00/00
 AMQA000043 QMQMDATA 00002 06/24/98 ONLINE 00/00/00
 AMQA000044 QMQMDATA 00003 06/24/98 ONLINE 00/00/00
 AMQA000045 QMQMDATA 00004 06/24/98 ONLINE 00/00/00
 AMQA000046 QMQMDATA 00005 06/24/98 ONLINE 00/00/00
 AMQA000047 QMQMDATA 00006 06/24/98 ONLINE 00/00/00
 More...

Parameters or command
====>
F3=Exit F4=Prompt F5=Refresh F11=Display size F12=Cancel

```

Figure 25. Display the Journal Receiver Attributes

2. Next, enter option 15, Work with Receiver Directory. You will then see a screen similar to the one shown in Figure 25.
3. Entering 8 display the attributes on receiver AMQA000042 and takes us to Figure 26.

```

 Display Associated Receivers
Receiver : AMQA000042 Library : QMQMDATA
Type options, press Enter.
 8=Display attributes

Option Receiver Library Association
 8 AMQA100041 QMQMDATA PREVIOUS
 AMQA000043 QMQMDATA NEXT

 Bottom
F3=Exit F12=Cancel

```

Figure 26. Display the Journal Receiver Attributes for AMQA000042

4. Work down the chain by typing 8 on the next AMQA000043 and you will get to the next pointer to AMQA000044 and so on. This enables you to work your way up the chain.

**AS/400**

---

## Chapter 4. MQSeries for UNIX Systems

This chapter covers the following topics:

- 4.1, Parameters for Creating the Queue Manager
- 4.2, MQSeries Logging and Logs
- 4.3, What to Backup
- 4.4, How to Re-synchronize the Channel
- 4.5, How to Re-synchronize the Channel with Media Command
- 4.6, How to Stop the Queue Manager
- 4.7, MQSeries for UNIX Systems Backup Procedures
  - 4.7.1, Two Backup Options
  - 4.7.2, How to Back Up All Queue Managers (System Backup)
  - 4.7.3, How to Back Up One Queue Manager (System Backup)
  - 4.7.4, How to Back Up One Queue Manager (Media Backup)
  - 4.7.5, How to Back Up a Synchronization File (Media Backup)
  - 4.7.6, How to Back Up a Queue (Media Backup)
- 4.8, MQSeries for UNIX Systems Recovery Scenarios
  - 4.8.2, Two Recovery Options
  - 4.8.3, How to Restore a Queue Manager (System Recovery)
  - 4.8.4, How to Recover a Queue Manager (Media Recovery)
  - 4.8.5, How to Recover the Synchronization File (Media Recovery)
  - 4.8.6, How to Recover a Queue (Media Recovery)

One of the main functions of MQSeries is to ensure that messages entered into the system are delivered to their destination.

MQSeries must also be capable of recovering messages if the system fails for any reason. To accomplish this, MQSeries maintains records (log files) of the queue manager's activities. Each queue manager handles the receipt, transmission and delivery of messages.

**Logs and Recovery**

MQSeries can use the logs for three ways of recovery:

1. RESTART RECOVERY, when you stop MQSeries in a planned way.
2. CRASH RECOVERY, when MQSeries is stopped due to a failure.
3. MEDIA RECOVERY, to restore damaged objects.

During recovery all persistent messages are restored; nonpersistent messages are lost.

---

#### 4.1 Parameters for Creating the Queue Manager

When you create a queue manager specify the type of logging that you will use, circular or linear logging. The type of logging cannot be changed, because it is specified only when the queue manager is created.

The default parameter for logging is *circular*.

If you need to change from circular logging to linear logging or change the size of the log, you have to re-create the queue manager. Use the following command to create a queue manager with the linear logging option:

```
crtmqm -ll -q QM_NAME
```

where:

|         |                                                                                              |
|---------|----------------------------------------------------------------------------------------------|
| -ll     | indicates that the queue manager will use <i>linear logging</i> with the default size of 1MB |
| -q      | indicates that this queue manager will be the default queue manager                          |
| QM_NAME | name of queue manager                                                                        |

**Recommendation**

1. In a production environment, you have to use linear logging. You must create the queue manager with the following parameters in order to perform media recovery:

```
crtmqm -ll -q QM_NAME
```

2. After creating and starting the queue manager QM\_NAME, you have to define, configure and create all customer application objects, such as channels and application queues with the *runmqsc* command using a predefined file, for example, CONFIG\_FILE.IN:

```
strmqm
runmqsc < CONFIG_FILE.IN > CONFIG_FILE.OUT
```

3. After all default system objects and customer application objects are created, it is recommended to perform a media backup for all objects to be used for the next media recovery:

```
rcdmqimg -m QM_NAME -t all *
```

**Or**

```
queue manager: rcdmqimg -m QM_NAME -t qmgr
```

```
and all queues: rcdmqimg -m QM_NAME -t q *
```

```
and all processes: rcdmqimg -m QM_NAME -t prcs *
```

```
and sync file: rcdmqimg -m QM_NAME -t syncfile amqrsyna.dat
```

MQSeries for UNIX has two configuration files:

**MQS.INI** This file contains the default parameters for the queue managers. It is located in `/var/mqm> mqs.ini`.

**MQ.INI** This contains the configuration for a specific queue manager. It is located in `/var/mqm/qmgrs/QM_NAME> qm.ini`.

For more information about configuration files refer to Chapter 7 in *MQSeries System Administration*, SC33-1873.

The MQSeries log files are located in the following path:

```
/var/mqm/log/QM_NAME/active>
```

## AIX

Each time an MQSeries object changes, information is written to a log before the changes become visible.

---

### 4.2 MQSeries Logging and Logs

MQSeries logs contain information you need to recover all updates to message queues by:

- Keeping records of queue manager changes
- Keeping records of queue updates for use by the restart process
- Enabling you to restore data after a hardware or software failure

MQSeries for UNIX has two types of logging:

|                 |                                                                                                                        |
|-----------------|------------------------------------------------------------------------------------------------------------------------|
| <b>Circular</b> | Circular logging keeps data in a ring of log files. When the last log is full it overwrites the first log.             |
| <b>Linear</b>   | Linear logging keeps the log data in a continuous sequence of files. This is necessary for restart and media recovery. |

#### 4.2.1 What Are the Necessary Logs for Recovery?

MQSeries issues two messages which tell about the necessary logs:

|                |                                                                            |
|----------------|----------------------------------------------------------------------------|
| <b>AMQ7467</b> | This message tells which is the oldest log necessary for MQSeries startup. |
| <b>AMQ7468</b> | This message tells which is the oldest journal for a media recovery.       |

---

### 4.3 What to Backup

We recommend that you make regular backups of the following:

- Main directory tree for all MQSeries environments
- Directory and files for one queue manager
- Queue manager
- System checkpoint file
- Objects (processes, queues, etc.)

#### 4.3.1 Directories and Files to Back Up

The main directory trees of MQSeries for UNIX systems, for example, MQSeries for AIX, are in:



```
/var/mqm/
/usr/lpp/mqm/
```

All objects for the queue managers in the UNIX system environment are located in the following directories:

```
/var/mqm/qmgrs/QM_NAME>amqalchk.fil
/var/mqm/qmgrs/QM_NAME/auth/
/var/mqm/qmgrs/QM_NAME/dce/
/var/mqm/qmgrs/QM_NAME/errors/
/var/mqm/qmgrs/QM_NAME/plugincomp/
/var/mqm/qmgrs/QM_NAME/prodef/
/var/mqm/qmgrs/QM_NAME/qmanager/
/var/mqm/qmgrs/QM_NAME>qm.ini
/var/mqm/qmgrs/QM_NAME/queues/
/var/mqm/qmgrs/QM_NAME/startprm/
/var/mqm/qmgrs/QM_NAME/esem/
/var/mqm/qmgrs/QM_NAME/isem/
/var/mqm/qmgrs/QM_NAME/msem/
/var/mqm/qmgrs/QM_NAME/shsem/PerQUEUE/
/var/mqm/qmgrs/QM_NAME/ssem/
/var/mqm/qmgrs/QM_NAME/@ipcc/
```

The logs and synchronization system files are located in the following path:

```
/var/mqm/qmgrs/QM_NAME/@ipcc> AMQCLCHL.TAB
/var/mqm/qmgrs/QM_NAME/@ipcc> AMQRFCD.ADAT
/var/mqm/qmgrs/QM_NAME/@ipcc> AMQRSYNA.DAT
/var/mqm/qmgrs/QM_NAME/@ipcc/esem/
/var/mqm/qmgrs/QM_NAME/@ipcc/isem/
/var/mqm/qmgrs/QM_NAME/@ipcc/msem/
/var/mqm/qmgrs/QM_NAME/@ipcc/shsem/PerQUEUE/
/var/mqm/qmgrs/QM_NAME/@ipcc/ssem/

/var/mqm/log/QM_NAME> amqhlctl.lfc
/var/mqm/log/QM_NAME/active> S0000000.LOG
/var/mqm/log/QM_NAME/active> S0000001.LOG
/var/mqm/log/QM_NAME/active> S0000002.LOG
```

### 4.3.2 System Checkpoint Files to Back Up

MQSeries in the UNIX environment has two important files:

*amqhlctl.lfh* contains checkpoint updates.  
*amqrsyna.dat* is the channel synchronization file.

## AIX

The files are located in the following directories:

```
/var/mqm/qmgrs/QM_NAME>amqalchk.fil
/var/mqm/qmgrs/QM_NAME/@ipcc> AMQRSYNA.DAT
```

The main MQSeries checkpoint files for UNIX are located in the following directories:

```
/var/mqm/qmgrs/QM_NAME>amqalchk.fil
/var/mqm/qmgrs/QM_NAME/@ipcc> AMQCLCHL.TAB
/var/mqm/qmgrs/QM_NAME/@ipcc> AMQRFCD.ADAT
/var/mqm/qmgrs/QM_NAME/@ipcc> AMQRSYNA.DAT
```

### 4.3.3 Queue Manager and Queues to Back Up

Take your regular backups for MQSeries objects, such as processes and queues, and the queue managers themselves. This backup should be used for the next restore to avoid a critical situation if there is any problem with a damaged queue or queue manager.

---

## 4.4 How to Re-synchronize the Channel

When the queue manager or one of its objects is damaged and you start the queue manager and channels, you will find some error codes in the MQSeries logs. Table 1 shows some of them. AMQ9526 is the most famous one.

| <i>Table 1 (Page 1 of 2). MQSeries Error Codes</i> |                                                   |
|----------------------------------------------------|---------------------------------------------------|
| <b>Error</b>                                       | <b>Description</b>                                |
| AMQ7088                                            | OBJECT ...TYPE...DAMAGED                          |
| AMQ7472                                            | OBJECT...TYPE...DAMAGED                           |
| AMQ8149                                            | MQSERIES OBJECT DAMAGED                           |
| AMQ9505                                            | CHANNEL SEQUENCE NUMBER WRAP VALUES ARE DIFFERENT |
| AMQ9506                                            | MESSAGE RECEIPT CONFIRMATION FAILED               |
| AMQ9507                                            | CHANNEL...IS CURRENTLY IN-DOUBT                   |
| AMQ9526                                            | MESSAGE SEQUENCE NUMBER ERROR FOR CHANNEL         |
| AMQ9543                                            | STATUS TABLE DAMAGED                              |
| AMQ9544                                            | MESSAGE WRITTEN TO THE DEAD LETTER QUEUE          |
| AMQ9550                                            | CHANNEL PGM...CANNOT BE STOPPED AT THIS TIME      |
| AMQ9556                                            | CHANNEL SYNCHRONIZATION FILE MISSING OR DAMAGED   |
| AMQ9563                                            | SYNCHRONIZATION FILE CANNOT BE DELETED            |

| <i>Table 1 (Page 2 of 2). MQSeries Error Codes</i> |                              |
|----------------------------------------------------|------------------------------|
| <b>Error</b>                                       | <b>Description</b>           |
| AMQ9999                                            | CHANNEL PGM ENDED ABNORMALLY |

This means that for each case of failure we have to determine the cause of inconsistency.

#### **Recommendation**

- Check if there is a hardware network connection (for example, no hardware problem with the token-ring card) or a local software network connection.
  - PING IP\_Code\_Local\_Queue\_Manager
  - PING IP\_Code\_Partners\_Queue\_Manager
- Check if the queue manager and the channels are ready to work:
  - runmqsc
  - dis qmgr all
  - ping qmgr
  - ping chl(CHL\_NAME)

If you need to re-synchronize the channels, the following tasks should be performed on the local and partner queue managers:

1. Verify the channel and transmission queue
2. Backout/commit and resolve channel
3. Reset the channel sequence number
4. Restart the channel

The next sections explain the above tasks.

You may also perform re-synchronization with the media recovery command. Refer to 4.5, "How to Re-synchronize the Channel with Media Command" on page 80. The command is:

```
rccmqobj -m QM_NAME -t syncfile amqrsyna.dat
```

#### **4.4.1 Verify Channel and Transmission Queue**

Verify the status of all channels and the number of the message held in the transmission queues.

- Step 1. Obtain the names of all sender channels and write them down:
- ```
DIS CHL(*) CHLTYPE(SDR)
```

AIX

- Step 2. Check each sender channel to find out the connection names and the transmission types (write them down):

```
DIS CHL(CHL_NAME) CONNAME TRPTYPE
```

- Step 3. With the following command, check if the channels are stopped:

```
DIS CHSTATUS(CHL_NAME) all
```

- If MCASTAT shows "not running" then the channel is inactive; otherwise stop the channel.
- If the channel is active, look for a JOBNAME. Write the job name down, just in case one of the users will call to find out why he cannot work any longer.
- If CURLUWID and LSTLUWID are zero than the channel is inactive. Otherwise, the channel will be in-doubt when it is started again. You have to resolve the channel. This is discussed later.

- Step 4. Find out the names of all transmission queues and write them down:

```
DIS CHL(*) XMITQ
```

- Step 5. For all active channels, check if there are messages in the transmission queue:

```
DIS Q(XMITQ_NAME) CURDEPTH
```

If there are messages in the queue, make a note of it.

- Step 6. Ping the channel to check if there is a network connection problem or a remote configuration definition problem.

```
PING CHL(CHL_NAME)
```

- Step 7. Start the channel:

```
STA CHL(CHL_NAME)
```

- Step 8. When starting the channel you may get error codes. Check the queue manager log:

```
/var/mqm/qmgrs/QM_NAME/errors> AMQERR01.LOG
```

- Step 9. If you get one of the following error codes you need to re-synchronize with the remote channel:

```
AMQ9556 CHANNEL SYNCHRONIZATION FILE MISSING OR DAMAGED.  
AMQ9563 SYNCHRONIZATION FILE CANNOT BE DELETED.
```

Use the media recovery command:

```
rcrmqobj -m QM_NAME -t syncfile amqrsyna.dat
```

This rebuilds the MQSeries channel synchronization file `amqrsyna.dat`, because the channel synchronization file is missing or does not correspond to the stored channel information for the queue manager.

4.4.2 Backout/Commit and Resolve Channel

If you get the error code "AMQ9507 CHANNEL IS CURRENTLY IN-DOUBT", you will have to examine the status of the channel and restart it to resolve the in-doubt state.

If after restarting the channel it is not possible to re-establish the channel connection then the SENDING END remains in doubt, as to whether or not the messages were received. Any outstanding Logical Unit of Work (LUW) needs to be resolved by being backed out or committed.

Recommendation

```
RESOLVE CHL(CHL_NAME) ACTION ( backout or commit )
```

Care must be exercised in the use of this command. If the specified resolution of the SENDING END is *not* the same as the resolution at the RECEIVING END, messages can be *lost* or *duplicated*.

1. If *backout* is used, the messages are backed out; that is, they are RESTORED TO the transmission queue.
2. If *commit* is used, the messages are committed; that is, they are DELETED FROM the transmission queue.

4.4.3 Reset the Sequence Number in the Channel

To synchronize the message sequence number with the partner queue manager in a distributed environment use RESET CHANNEL to reset the message sequence number for a message channel with, optionally, a specified sequence number to be used the next time the channel is started:

```
RESET CHL(CHL_NAME) SEQNUM(n)
```

- The new message sequence number must be greater than or equal to 1 and less than or equal to 999 999 999.
- This sequence number is used the next time the channel is started.
- This command can be issued for a channel of any type, except SVRCONN or CLNTCONN channels.

AIX

- If the command is issued for a SENDER or SERVER channel, then, in addition to resetting the value at the local end at which the command is issued, the value at the partner end (receiver or requester) will also be reset to the same value the next time this channel is initiated and re-synchronized, if necessary.

4.4.4 Restart the Channel

1. Start the channel with the following command:

```
STA CHL(CHL_NAME)
```

2. Display the status of the channel:

```
DIS CHS(CHL_NAME) ALL
```

3. Display the number of messages in the transmission queue:

```
DIS Q(XMITQ_NAME) CURDEPTH
```

4. If there are no messages in the transmission queue then they have been sent to the partner queue manager.

4.5 How to Re-synchronize the Channel with Media Command

In the case of a damaged channel synchronization file you will need to restore this file and re-synchronize the channel. To restore and re-synchronize the channel by media recovery use the following command:

```
rcrmqobj -m QM_NAME -t syncfile amqrsyna.dat
```

This command can be used after the image of the channel synchronization file has been recorded with the media backup command `rcdmqimg`. This command must be used when a queue manager is active.

4.6 How to Stop the Queue Manager

MQSeries provides three ways to end a queue manager:

4.6.1 Normal Shutdown

The `endmqm` command is used to end (stop) a specified local queue manager, for example:

```
endmqm QM_NAME
```

The queue manager ends only after all applications have been disconnected from the queue manager. Any MQI calls currently being processed will have been completed. This is the default.

4.6.2 Immediate Shutdown

The parameter "i" in the endmqm command will stop the queue manager immediately:

```
endmqm -i QM_NAME
```

The queue manager stops after it has completed all MQI calls currently being processed. Any MQI requests issued after the command has been issued will fail. Any incomplete logical units of works (LUW) are rolled back when the queue manager is started again.

4.6.3 Preemptive Shutdown

This option is caused by the parameter "p" in the endmqm command":

```
endmqm -p QM_NAME
```

You should use this type of shutdown *only in exceptional circumstances*, for example, when a queue manager does not stop as a result of a normal endmqm command. The queue manager will end without waiting for applications to disconnect or for MQI calls to complete. This can cause unpredictable results in the MQ applications. All processes of the queue manager that fail to stop are terminated 30 seconds after the command is issued.

4.7 MQSeries for UNIX Systems Backup Procedures

This section describes the two ways of backing up MQSeries objects and files through system and media backup. The following scenarios are discussed:

- System backup of all queue managers
- System backup of one queue manager
- Media backup of one queue manager
- Media backup of synchronization files
- Media backup of a queue

4.7.1 Two Backup Options

There are two kinds of MQSeries backup:

- System backup
- Media backup

4.7.1.1 System Backup

This operation makes full copies all MQSeries system and application files in order to restore all installed MQSeries directories and files in /var/mqm/ and /usr/lpp/mqm/ in case of a disaster.

You can back up all or a specific queue manager. The system backup will be performed when all queue managers are stopped.

4.7.1.2 Media Backup

You have to perform the media backup under the control of an active queue manager. This is done with the media command `rcdmqimg`.

The `rcdmqimg` command is used to write an image of an MQSeries object or a group of objects to the log for use in media recovery.

Examples

Back up all objects of queue manager MQ61:

```
rcdmqimg -m MQ61 -t all *
```

Back up the queue manager MQ61:

```
rcdmqimg -m MQ61 -t qmgr
```

Back up all queues that begin with SYSTEM* and MQ61*:

```
rcdmqimg -m MQ61 -t q SYSTEM*
```

```
rcdmqimg -m MQ61 -t q MQ61*
```

Back up all processes:

```
rcdmqimg -m MQ61 -t prcs *
```

To use the media command, the queue manager must have been created with the linear logging option.

4.7.1.3 About Error Code AMQ7044

When you try to use the media command you may get the following error:

```
AMQ7044 MEDIA RECOVERY NOT ALLOWED
```

The reason is that the queue manager uses circular logging instead of linear logging.

To check what logging option you use look in this file:

```
/var/mqm/qmgrs/QM_NAME> qm.ini
```


If you see the parameter `LogType=CIRCULAR` then you *cannot use* the media command for the queue manager. If you want to use the media command, you have to:

1. Delete the current queue manager.
2. Re-create the new queue manager with *linear logging* using the following command:

```
crtmqm -ll -q QM_NAME
```

4.7.2 How to Back Up All Queue Managers (System Backup)

The system recovery needs to restore all MQSeries system files, logs and objects, such as product system files, archive logs, active logs, configuration files, source and load of MQSeries system programs. To create a complete MQSeries backup you have to back up the MQSeries directory structure.

You can perform a system backup of all queue managers when all queue managers are NOT active (normally stopped). The following tasks are necessary:

1. Stop all channels.
2. Perform all media backups.
3. Shut down all queue managers.
4. Back up the system (`/var/mqm/` and `/usr/lpp/mqm/`).
5. Restart the queue managers.
6. Start the channels.

The above functions are described, in detail, in the following checklist.

4.7.2.1 Scenario

Step 1. Stop all channels

For each queue manager in the machine, you have to do the following:

- a. Stop each channel with the following command:

```
STOP CHL(CHL_NAME)
```

- b. Check for errors in the queue manager's log:

```
/var/mqm/qmgrs/QM_NAME/errors> AMQERR01.LOG
```

Step 2. Perform the media backup.

Perform a backup of each queue manager for next media recovery:

```
RCDMQIMG -m QM_NAME -t all *
```

AIX

Example for queue manager MQ61:

```
RCDMQIMG -m MQ61 -t all *
```

Or

Back up the queue manager object:

```
RCDMQIMG -m MQ61 -t q qmgr
```

and all queues starting with SYSTEM* and MQ61*:

```
RCDMQIMG -m MQ61 -t q SYSTEM*  
RCDMQIMG -m MQ61 -t q MQ61*
```

and all processes:

```
RCDMQIMG -m MQ61 -t prcs *
```

Step 3. Shut down all queue managers:

```
ENDMQM QM_NAME
```

Step 4. Back up all queue managers.

To perform a complete MQSeries backup, you have to back up the MQSeries product directory structure. This directory is the same for all queue managers in the same machine. Use the copy command to back up the following directories:

```
/var/mqm/  
/usr/lpp/mqm/
```

Write down the date and time in your Data Processing Operation Control Book.

Step 5. Restart the queue managers:

```
STRMQM QM_NAME
```

Step 6. Start the channels:

a. Start each channel with the command:

```
STA CHL(CHL_NAME)
```

b. Check the logs for each queue manager for errors. You find the logs in the directory:

```
/var/mqm/qmgrs/QM_NAME/errors> AMQERR01.LOG
```

c. If you got some MQSeries error codes during channel start you have to re-synchronize the local channel with the remote channel.

4.7.3 How to Back Up One Queue Manager (System Backup)

A system backup of a single queue manager is performed when the queue manager is not active (normally stopped). The following steps are necessary:

1. Stop all channels.
2. Perform the media backup of the queue manager.
3. Shut down the queue manager.
4. Back up `/var/mqm/log/QM_NAME/` and `/var/mqm/qmgrs/QM_NAME/`.
5. Restart the queue manager.
6. Start the channels.

4.7.3.1 Scenario

Step 1. Stop all channels.

For each channels, use the following command:

```
STOP CHL(CHL_NAME)
```

Check for any errors in the log:

```
/var/mqm/qmgrs/QM_NAME/errors> AMQERR01.LOG
```

Step 2. Back up the queue manager.

For next Media Recovery, use the following command:

```
RCDMQIMG -m QM_NAME -t all *
```

Example

Back up queue manager MQ61 and all of its objects:

```
RCDMQIMG -m QM_NAME -t all *
```

Back up the queue manager MQ61:

```
RCDMQIMG -m MQ61 -t qmgr
```

Back up all of its queues starting with SYSTEM* and MQ61*:

```
RCDMQIMG -m MQ61 -t q SYSTEM*
```

```
RCDMQIMG -m MQ61 -t q MQ61*
```

Step 3. Shut down the queue manager with the command:

```
ENDMQM QM_NAME
```

Step 4. Back up the queue manager.

AIX

To perform a single queue manager backup, you have to back up the queue manager's directory structure using the copy command.

The directories are:

```
/var/mqm/qmgrs/QM_NAME/  
/var/mqm/log/QM_NAME/
```

Write down the date and time of the system backup in your Data Processing Operation Control Book.

Step 5. Restart the queue manager.

If this backup operation is successful restart the queue manager:

```
STRMQM QM_NAME
```

Check for errors in the log:

```
/var/mqm/qmgrs/QM_NAME/errors> AMQERR01.LOG
```

Step 6. Start the channels.

Start each channel with the following command:

```
STA CHL(CHL_NAME)
```

If the log shows some errors that occurred during channel startup you have to re-synchronize the local channel with the remote channel.

4.7.4 How to Back Up One Queue Manager (Media Backup)

To back up a queue manager with the media backup command you have to do the following tasks:

1. Stop all channels.
2. Perform the media backup with: `rcdmqimg -m QM_NAME -t all *`
3. Start the channels.

When you perform a media backup of a queue manager it must be active.

4.7.4.1 Scenario

Step 1. Stop all channels.

Check if all channels are stopped with the command:

```
DIS CHSTATUS (CHL_NAME)
```

Stop a channel with this command:

```
STOP CHL(CHL_NAME)
```

Check if there are some messages in the transmission queue with the display command:

```
DIS Q(XMIT_NQME) CURDEPTH
```

Step 2. Back up the queue manager:

```
rcdmqimg -m QM_NAME -t all *
```

Example

Back up queue manager MQ61 and all of its objects:

```
RCDMQIMG -m QM_NAME -t all *
```

Back up the queue manager MQ61:

```
RCDMQIMG -m MQ61 -t qmgr
```

Back up all of its queues starting with SYSTEM* and MQ61*:

```
RCDMQIMG -m MQ61 -t q SYSTEM*
RCDMQIMG -m MQ61 -t q MQ61*
```

Check for errors in the log:

```
/var/mqm/qmgrs/QM_NAME/errors> AMQERR01.LOG
```

Step 3. Start the channels.

If the queue manager is OK, start the channels:

```
STA CHL(CHL_NAME)
```

4.7.5 How to Back Up a Synchronization File (Media Backup)

Perform the media backup of the channel synchronization file when the queue manager is running. Execute the following steps:

Step 1. Stop each channel with the command:

```
STOP CHL(CHL_NAME)
```

Step 2. Perform the media backup:

```
rcdmqimg -m QM_NAME -t syncfile amqrsyna.dat
```

Example

Backing up the sync file for queue manager MQ61:

```
rcdmqimg -m MQ61 -t syncfile amqrsyna.dat
```

Step 3. Start each channel with the command:

```
STA CHL(CHL_NAME)
```

AIX

4.7.6 How to Back Up a Queue (Media Backup)

The queue manager must be running when you back up a queue using media backup.

Example

Backing up the application queue MQ61.QL11:

```
rccmqimg -m MQ61 -t q MQ61.QL11
```

- To back up all queues of queue manager MQ61 starting with SYST* and MQ61* use the following commands:

```
rccmqimg -m MQ61 -t q SYST*
```

```
rccmqimg -m MQ61 -t q MQ61*
```

- To back up all local queues of queue manager MQ61 execute:

```
rccmqimg -m MQ61 -t ql MQ61*
```

- To back up all remote queues of queue manager MQ61 use the command:

```
rccmqimg -m MQ61 -t qr MQ61*
```

- Back up all alias queues in queue manager MQ61 with:

```
rccmqimg -m MQ61 -t qa MQ61*
```

- Back up all model queues of the queue manager MQ61 with:

```
rccmqimg -m MQ61 -t qm MQ61*
```

4.8 MQSeries for UNIX Systems Recovery Scenarios

This section covers the following topics:

- Reasons for a recovery
- System and media recovery
- System recovery of a queue manager
- Media recovery of a queue manager
- Media recovery of the channel synchronization files
- Media recovery of a queue

4.8.1 Reasons for a Recovery

The reasons for the recovery of MQSeries objects in a production environment can be:

- A disaster occurred or the queue manager is damaged.

- There is a problem with the system logs, for example, error AMQ7467 or AMQ7468 occurred.
- The channel synchronization file is damaged.
- Objects are damaged.

Recommendation

- All MQSeries objects must be backed up at a scheduled time in accordance with other remote queue managers. This will avoid channel synchronization and application integration problems.
- Before you perform the MQSeries recovery check if you have:
 1. All authorization access to MQSeries (group mqm/user mqm)
 2. A large disk space in the machine

4.8.2 Two Recovery Options

There are two kinds of MQSeries recovery:

- System recovery
- Media recovery

4.8.2.1 System Recovery

This operation uses the full copies all MQSeries system and application files in order to restore all installed MQSeries directories and files in /var/mqm/ and /usr/lpp/mqm/.

The system backup must have been performed when all queue managers were stopped. After restoring the system files and active logs you may get some error codes when starting the channels. In such a case you need to re-synchronize the local channel with the remote channel.

4.8.2.2 Media Recovery

You have to perform the media recovery under control of an active queue manager. Use the media command rcrmobj. The objects will be recovered automatically at their last normal commit state.

Examples

Recover all objects of queue manager MQ61:

```
rccrmqimg -m MQ61 -t all *
```

Recover all queues of queue manager MQ61:

```
rccrmqobj -m MQ61 -t q *
```

Recover all queues that begin with SYSTEM* and MQ61*:

```
rccrmqimg -m MQ61 -t q SYSTEM*
```

```
rccrmqimg -m MQ61 -t q MQ61*
```

Recover all processes:

```
rccrmqimg -m MQ61 -t prcs *
```

Note: To use the media command the queue manager must have been created with the *linear* logging option.

4.8.2.3 About Error Code AMQ7044

When you try to use the media recovery command you may get the following error:

```
AMQ7044 MEDIA RECOVERY NOT ALLOWED
```

The reason is that your queue manager uses circular logging instead of linear logging. To check which logging option you use look in this file:

```
/var/mqm/qmgrs/QM_NAME> qm.ini
```

If you see the parameter LogType=CIRCULAR the you *cannot use* the media command. In order to use it you have to:

1. Delete the current queue manager.
2. Re-create a new queue manager with linear logging using the following command:

```
crtmqm -ll -q QM_NAME
```

4.8.3 How to Restore a Queue Manager (System Recovery)

To restore a queue manager, you have to do the following:

1. Stop all channels.
2. Shut down the queue manger.
3. Archive the current logs.
4. Restore the queue manager files in /var/mqm/log/QM_NAME/ and /var/mqm/qmgrs/QM_NAME/.

5. Restart the queue manager.
6. Start the channels.
7. If necessary, re-synchronize the channels.

The above functions are described, in detail, in the following scenario.

4.8.3.1 Scenario

Step 1. Stop all channels.

Check if the sender channels are stopped:

```
DIS CHL(*) CHLTYPE(SDR)
DIS CHL(CHL_NAME) all
DIS CHSTATUS(CHL_NAME) all
```

Check if there are any messages in the transmission queue:

```
DIS CHL(*) XMITQ
DIS Q(XMITQ_NAME) CURDEPTH
```

If you want to restore all objects of a queue manager which has been backed up several days ago, and you need to get the data of all integrated applications from before and after the time of restore, you have to check the MQSeries remote logs for the number of messages in the transmission queues.

Stop each sender channel with the command:

```
STOP CHL(CHL_NAME)
```

Step 2. Shut down the queue manager.

To stop the queue manager and to end all MQSeries application programs, use the following command:

```
ENDMQM QM_NAME
```

If the queue manager is not successfully stopped, you can use one of the following:

```
ENDMQM -I QM_NAME
```

```
ENDMQM -P QM_NAME
```

Step 3. Archive your current system logs:

- a. Archive the `/var/mqm/log/QM_NAME/` directory.
- b. Archive the `amqrsyna.dat` file in the directory. It is in `/var/mqm/qmgrs/QM_NAME/@ipcc`.

Step 4. Restore the queue manager files.

AIX

For system recovery of the queue manager, the following directories have to be restored:

```
/var/mqm/qmgrs/QM_NAME/  
/var/mqm/log/QM_NAME/
```

Check for any errors in the log:

```
/var/mqm/qmgrs/QM_NAME/errors> AMQERR01.LOG
```

Recommendation

In a production environment, each cause of failure is different.

If there is a problem with the disk drive containing either the queue manager data, the log, or both the result may include data loss or data corruption.

You must first analyze the situation and check the directory structures `/var/mqm/` and `/usr/lpp/mqm/` for any damage, and repair such damage.

For example, to do a restore for the queue manager `QM_NAME` from the system backup files:

- a. Find the following current directories for the queue manager:

```
/var/mqm/qmgrs/QM_NAME>  
/var/MQM/log/QM_NAME>
```

- b. Clear these directories by deleting them and then re-creating them again.

- c. Restore the files from the last system backup of the queue manager using a copy command in the directories:

```
/var/mqm/qmgrs/QM_NAME>  
/var/mqm/log/QM_NAME>
```

- d. Restore the newest logs you archived into their original places.

- e. Restore `amqrsyna.dat` to its original place.

- f. Check if there are any errors in the log:

```
/var/mqm/qmgrs/QM_NAME/errors> AMQERR01.LOG
```

- Step 5. Restart the queue manager.

If all the above steps have completed successfully, you can start the queue manager:

```
STRMQM QM_NAME
```

All the objects will be recovered to their last operational state automatically.

Step 6. Start the channels.

If the Queue Manager is OK, start each channels with the command:

```
STA CHL(CHL_NAME)
```

Step 7. If necessary, re-synchronize the channel.

If starting the channel causes an error you have to re-synchronize the local channel with the remote channel.

4.8.4 How to Recover a Queue Manager (Media Recovery)

To recover a queue manager with the media recovery command you have to do the following:

1. Stop all channels.
2. End the queue manager.
3. Restart the queue manager.
4. Perform the media recovery with the command:


```
rcrmqobj -m QM_NAME -t all *
```
5. Start the channels.
6. If necessary, re-synchronize the channels.

4.8.4.1 Scenario

Step 1. Stop the channels.

Obtain the names of the sender channels with the command:

```
DIS CHL(*) TYPE(SDR) ALL
```

Check if all channels are stopped with the command:

```
DIS CHSTATUS (CHL_NAME) ALL
```

Stop each channel with the command:

```
STOP CHL(CHL_NAME)
```

Check if there are any messages in the transmission queue:

```
DIS Q(XMIT_NAME) CURDEPTH
```

Step 2. End the queue manager.

To stop the queue manager and to end all MQSeries application programs, use the following command:

```
ENDMQM QM_NAME
```

If the queue manager is not successfully stopped, you can use one of the following commands:

AIX

```
ENDMQM -I QM_NAME
```

```
ENDMQM -P QM_NAME
```

- Step 3. Restart the queue manager.

If all the above steps have completed successfully, you can start the queue manager:

```
STRMQM QM_NAME
```

All objects will be recovered to their last operational state automatically.

- Step 4. Perform the media recovery.

Use the following command when the queue manager is running:

```
rcrmqobj -m QM_NAME -t all *
```

Examples

Recover the queue manager MQ61 and all of its objects:

```
rcrmqobj -m MQ61 -t all *
```

Recover all queues belonging the MQ61:

```
rcrmqobj -m MQ61 -t q *
```

Recover all queues beginning with SYSTEM* and MQ61*:

```
rcrmqobj -m MQ61 -t q SYSTEM*
```

```
rcrmqobj -m MQ61 -t q MQ61*
```

Recover all processes:

```
rcrmqobj -m MQ61 -t prcs *
```

Make sure that there are no errors in the log:

```
/var/mqm/qmgrs/QM_NAME/errors> AMQERR01.LOG
```

Read the recommendations on page 95.

- Step 5. Start the channels.

If the queue manager is okay start each channel with the command:

```
STA CHL(CHL_NAME)
```

- Step 6. If there are problems, re-synchronize the channels.

If an error occurs while starting a channel you have to re-synchronize the local channel with the remote channel.

Recommendations for System / Media Recovery

1. *Media recovery* with the command `rcrmqobj -t all *` will recover all objects of any type of the queue manager EXCEPT channels.
2. In a *production environment* you must first analyze the situation because each case is different.
3. If there are problems with a disk drive, data loss or data corruption may have occurred.
4. You must check the directory structures `/var/mqm/` and `/usr/lpp/mqm/` for any damage and, if necessary, repair such damage.
5. If you lose queue manager data, there is the danger that the directory structure of related queue managers will be damaged.
6. This means that you will first need to perform a *system recovery* and then a *media recovery*:
 - a. If so, you must to *stop the queue manager*.
 - b. After that, you have to manually *re-create the directory tree* before you can try to restart the queue manager.
 - c. *Remove all log files* back to the queue manager name level `/var/mqm/qmgrs/QM_NAME/` including the configuration files, logs and queue manager directory.
 - d. *Restore the last system backup* in order to repair the damaged queue manager.
 - e. The checkpoint file `/var/mqm/qmgrs/QM_NAME/amqalchk.fil` must be restored as part of the queue manager data.
 - f. The log file directory must contain the *oldest log file* that was required to start the queue manager at the time of the backup and all subsequent log files.
 - g. Now you can *restart* the queue manager.

4.8.5 How to Recover the Synchronization File (Media Recovery)

The command `rcrmqobj` is used to re-create the channel synchronization file. This command must be used when the queue manager is running and after an image was recorded in the log.

Example

```
Recover the channel synchronization file:
rcrmqobj -m QM_NAME -t syncfile amqrsyna.dat
```

4.8.5.1 About Error Code AMQ9556

If you get the message "AMQ9556 Channel synchronization file missing or damaged", you have to perform a media recovery for this system file consisting of the following tasks:

1. Stop all channels.
2. End the queue manager.
3. Restart the queue manager.
4. Perform media recovery with the command:

```
rcrmqobj -m QM_NAME -t syncfile amqrsyna.dat
```
5. Start the channels.

4.8.6 How to Recover a Queue (Media Recovery)

Note: The queue manager must be running.

Example

Recover the queue MQ61.QL12 of queue manager MQ61 using media recovery:

```
rcrmqobj -m MQ61 -t q MQ61.QL12
```

4.8.6.1 About Error Code AMQ8149

If you get the message "AMQ8149 MQSeries object damaged" you have to perform a media recovery for this object.

The command `rcrmqobj` is used to re-create application or system queues. This command must be used when the queue manager is running and an image has been recorded in the log. Here are some examples:

- Media recovery of all queues of the queue manager MQ61:

```
rcrmqobj -m MQ61 -t q SYST*
```

```
rcrmqobj -m MQ61 -t q MQ61*
```
- Media recovery all local queues of queue manager MQ61:

```
rcrmqobj -m MQ61 -t ql MQ61*
```
- Media recovery of all remote queues of queue manager MQ61:

```
rcrmqobj -m MQ61 -t qr MQ61*
```
- Media recovery of all alias queues of queue manager MQ61:

```
rcrmqobj -m MQ61 -t qa MQ61*
```
- Media recovery of all model queues of the queue manager MQ61:

```
rcrmqobj -m MQ61 -t qm MQ61*
```

AIX

Chapter 5. MQSeries for Windows NT and MQSeries for OS/2

This chapter covers the following topics:

- 5.1, Parameters for Creating the Queue Manager
- 5.2, MQSeries Logging and Logs
- 5.3, What to Back Up
- 5.4, How to Re-synchronize the Channel
- 5.5, How to Re-synchronize the Channel with Media Command
- 5.6, How to Stop the Queue Manager
- 5.7, MQSeries for Windows NT and OS/2 Backup Procedures
 - 5.7.1, Two Backup Options
 - 5.7.2, How to Back Up All Queue Managers (System Backup)
 - 5.7.3, How to Back Up One Queue Manager (System Backup)
 - 5.7.4, How to Back Up One Queue Manager (Media Backup)
 - 5.7.5, How to Back Up the Synchronization File (Media Backup)
 - 5.7.6, How to Back Up a Queue (Media Backup)
- 5.8, MQSeries for Windows NT and OS/2 Recovery Scenarios
 - 5.8.1, Two Recovery Options
 - 5.8.2, How to Restore a Queue Manager (System Recovery)
 - 5.8.3, How to Recover a Queue Manager (Media Recovery)
 - 5.8.4, How to Recover the Synchronization File (Media Recovery)
 - 5.8.5, How to Recover a Queue (Media Recovery)

One of the principal functions of MQSeries is to ensure that messages entered into the system are delivered to their destination.

MQSeries must also be capable of recovering messages if the system fails for any reason. To accomplish this, MQSeries maintains records (log files) of the activities of the queue managers. Each queue manager handles the receipt, transmission and delivery of messages.

OS/2 and Windows NT

Logs and Recovery

MQSeries can use the logs for three ways of recovery:

1. RESTART RECOVERY, when you stop MQSeries in a planned way.
2. CRASH RECOVERY, when MQSeries is stopped due to a failure.
3. MEDIA RECOVERY, to restore damaged objects.

During recovery all persistent messages are restored; nonpersistent messages are lost.

5.1 Parameters for Creating the Queue Manager

When you create a queue manager you specify the type of logging that you will use, circular or linear logging. The type of logging cannot be changed because it is specified only when the queue manager is created.

The default parameter for logging option is *circular*.

If you need to change from circular logging to linear logging or the size of the logs you must delete and re-create the queue manager.

Use the following command to create a queue manager with linear logging:

```
crtmqm -ll -q QM_NAME
```

-ll	indicates that the queue manager will use LINEAR LOGGING with the default size (1MB).
-q	indicates that this queue manager will be the default queue manager.
QM_NAME	name of the queue manager

Recommendation

1. In a production environment, you have to use the linear logging. You must to create the queue manager with the following parameters in order to perform media recovery:

```
crtmqm -ll -q QM_NAME
```

2. After creating and starting the queue manager QM_NAME, you have to define, configure and create all customer application objects, such as channels and application queues with the *runmqsc* command using a predefined file, for example, CONFIG_FILE.IN:

```
strmqm  
runmqsc < CONFIG_FILE.IN > CONFIG_FILE.OUT
```

3. After all default system objects and customer application objects are created, it is recommended to perform a media backup of all objects to be used for the next media recovery:

```
rcdmqimg -m QM_NAME -t all *
```

You may also record the object types separately:

```
queue manager: rcdmqimg -m QM_NAME -t qmgr
```

```
and all queues: rcdmqimg -m QM_NAME -t q *
```

```
and all processes: rcdmqimg -m QM_NAME -t prcs *
```

```
and sync file: rcdmqimg -m QM_NAME -t syncfile amqrsyna.dat
```

If you want to change the log size to 16 MB for queue manager MQT1, for example, type the command:

```
CRTMQM -ll -lf4095 -q MQT1
```

The size of the logs are in units of 4KB. In MQSeries for Windows NT, the minimum value is 32 and the maximum is 4095. The default value is 256, giving a default log size of 1 MB.

MQSeries for Windows NT and MQSeries for OS/2 have two configuration files:

OS/2 and Windows NT

MQS.INI This file contains the default parameters for all queue managers. It is located in \MQM\MQS.INI.

MQ.INI This file contains the configuration for a specific queue manager. It is located in \MQM\qmgrs\QM_NAME\QM.INI.

For more information about configuration files refer to Chapter 7 in *MQSeries System Administration*, SC33-1873.

The MQSeries log files are located in the following path:

```
C:\MQM\log\QM_NAME\active>
```

The directory structure contains three files, for example:

S00000005.log	1,025 KB
S00000006.log	1,025 KB
S00000007.log	1,025 KB

Each time an MQSeries object changes, the information is written to a log before the changes become visible.

The default size for the logs in MQSeries is 1MB. The logs contain information about local objects and the status of all queues and their messages. The maintenance of the logs is the administrator's responsibility. The logs are created automatically. When one log fills up another log is created with the next sequence number. For example, when the current log, S0000005, becomes full the queue manager creates a new log with the name S0000006.

5.2 MQSeries Logging and Logs

MQSeries records all significant changes to the data controlled by the queue manager in a log. This includes events such as:

- The creation of objects (except channels)
- The deletion of objects
- All persistent message updates
- Several transaction states
- All changes of object attributes and channel activities (but not including channel definitions)

MQSeries logs contain the information you need to recover all updates to message queues by:

- Keeping records of queue manager changes

OS/2 and Windows NT

- Keeping records of queue updates for use by the restart process
- Enabling you to restore data after a hardware or software failure

MQSeries for Windows NT and MQSeries for OS/2 have two types of logging:

Circular	Circular logging keeps data in a ring of log files. When the third log is full, it overwrites with the first one.
Linear	Linear logging keeps the log data in a continuous sequence of files. This is necessary for restart and media recovery.

The size and number of logs depends on the size and number of messages.

5.2.1 Which Are the Necessary Logs for Recovery?

MQSeries issues two messages which tell about the necessary logs:

AMQ7467	This message tells which is the oldest log necessary for MQSeries startup.
AMQ7468	This message tells which is the oldest log for a media recovery.

Examples of the two messages are shown in Figure 27 on page 104 and Figure 28 on page 105.

OS/2 and Windows NT

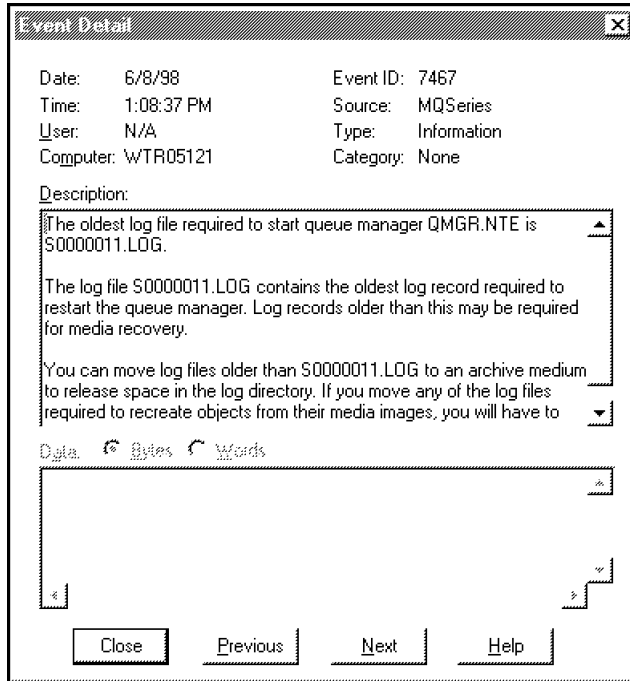


Figure 27. Message AMQ7467

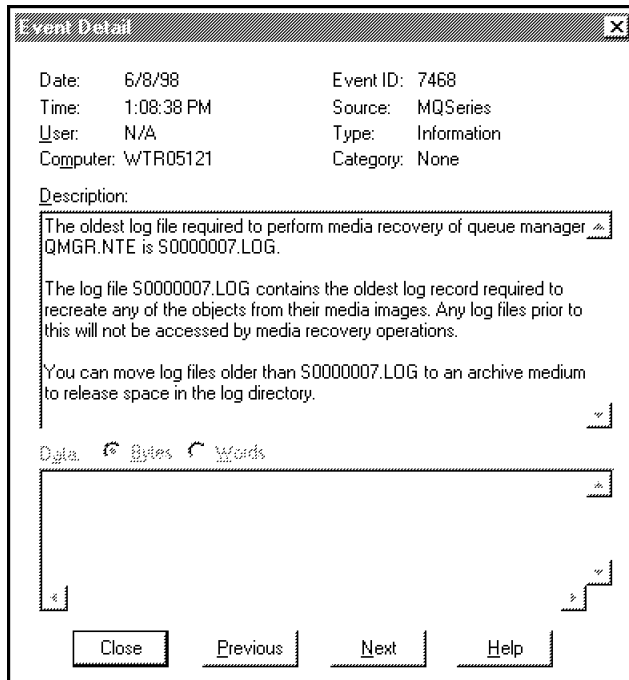


Figure 28. Message AMQ7468

In this example, the logs older than S0000006 can be archived or deleted. We recommend to back up the logs between S0000007 and the current log, in this case S0000010 because these are necessary for media recovery.

In this example, the current log is S0000010. How do we know that? You need to use the Windows NT Event Viewer.

5.2.2 Log Problems and Queues Containing Very Old Messages

If the queue contains very old messages the log entries necessary for the MQSeries startup will be very large. You should remove these old messages from the queues.

In a production environment, message delivery can take hours and the messages can remain in the queues for a long time. A record image is the best solution for this situation, because all changes within MQSeries and the data in the queues are recorded in a new log. The older logs will not be necessary and the MQSeries administrator can delete them.

Example:

OS/2 and Windows NT

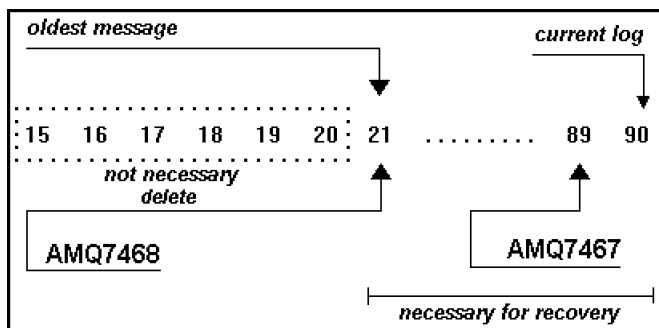


Figure 29. Dealing with Old Messages

Let us assume a queue receives a message (M1) from a remote system and the active log is S0000021. Therefore, the PUT of the message is registered in that log. Message M1 stays in the queue for a very long time while other messages arrive for the same queue manager. It doesn't matter if the messages are for the same or other queues. All messages except M1 are "gotten", that is, delivered to their target application. The logs fill up and the system creates new ones. So, currently we work with the log S0000090. For a media recovery we would need the logs S0000021 through S0000090, because the message M1 is still in the queue and the image is recorded in S0000021.

Even though MQSeries automatically records an image of an object, we strongly recommend to use the RCDMQIMG command. This will reduce the number of logs required for media recovery.

In MQSeries for Windows NT and MQSeries for OS/2, you record a record image with the following command:

```
rcdmqimg -m QM_NAME -t all *
```

With this command, all objects are recorded in a new log, and this log will be the first one required for a media recovery. After the media backup the queue manager processes more messages and the active log is now S0000095. This situation is illustrated in Figure 30 on page 107.

For a media recovery we need the logs S0000091 through S0000095, since the image of message M1 is recorded in log S0000091. We can archive or delete logs from S0000021 through S0000090; they are not necessary for either system startup or media recovery.

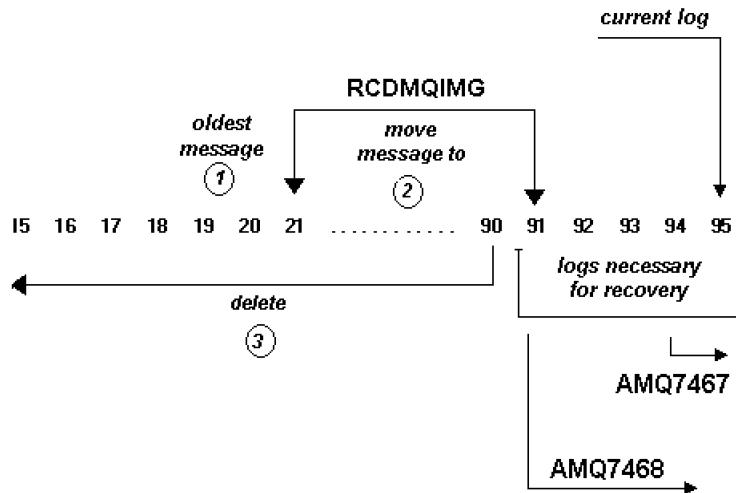


Figure 30. Logs Required for Startup and Media Recovery

5.3 What to Back Up

We recommend that you make regular backups of the following:

- Main directory tree for all MQSeries environments
- Directories and files for one queue manager
- Queue manager
- System checkpoint files (channel re-synchronization file)
- Objects (processes, queues, etc.)

5.3.1 Directories and Files to Back Up

The following directories will be used for a *system recovery* of the queue manager. The main directory tree of MQSeries for Windows NT and MQSeries for OS/2 is \MQM\.

All objects of the queue managers of the MQSeries for Windows NT and MQSeries for OS/2 are located in the following directories:

```

\MQM\qmgrs\QM_NAME\@ipcc
\MQM\qmgrs\QM_NAME\auth
\MQM\qmgrs\QM_NAME\dce
\MQM\qmgrs\QM_NAME\errors
\MQM\qmgrs\QM_NAME\plugcomp
\MQM\qmgrs\QM_NAME\procdef
\MQM\qmgrs\QM_NAME\Qmanager

```

OS/2 and Windows NT

```
\MQM\qmgrs\QM_NAME\Queues
\MQM\qmgrs\QM_NAME\startprm
```

The logs and synchronization system files are located in the following path:

```
\MQM\log\QM_NAME\active>S0000000.LOG
\MQM\log\QM_NAME\active>S0000001.LOG
\MQM\log\QM_NAME\active>S0000002.LOG
\MQM\log\QM_NAME>amqh1ct1.lfh
\MQM\qmgrs\QM_NAME\@ipcc>AMQCLCHL.TAB
\MQM\qmgrs\QM_NAME\@ipcc>AMQRFCD.A.DAT
\MQM\qmgrs\QM_NAME\@ipcc>AMQRSYNA.DAT
```

5.3.2 System Checkpoint Files to Back Up

These files will be used to recover the channel re-synchronization files in a system recovery using the xcopy or copy commands.

MQSeries has two important files:

amqh1ct1.lfh contains checkpoint updates.
amqrsyna.dat is the channel synchronization file.

The MQSeries checkpoint files are located in:

```
\MQM\log\QM_NAME>amqh1ct1.lfh
\MQM\qmgrs\QM_NAME\@ipcc>AMQCLCHL.TAB
\MQM\qmgrs\QM_NAME\@ipcc>AMQRFCD.A.DAT
\MQM\qmgrs\QM_NAME\@ipcc>AMQRSYNA.DAT
```

5.3.3 Queue Manager and Queues to Back Up

Take backups of MQSeries objects, such as processes, queues and the queue manager itself regularly. They will be used for the next restore and will help you avoid a critical situation if there are problems with damaged objects or the queue manager.

5.4 How to Re-synchronize the Channel

When the queue manager or one of its objects is damaged and you start the queue manager and channels, you will find some error codes in the MQSeries logs. Table 2 shows some of them. AMQ9526 is the famous one.

<i>Table 2 (Page 1 of 2). MQSeries Error Codes</i>	
Error	Description
AMQ7084	OBJECT...(1) TYPE ...DAMAGED
AMQ7472	OBJECT...(2) TYPE ...DAMAGED

OS/2 and Windows NT

Error	Description
AMQ8149	MQSERIES OBJECT DAMAGED
AMQ9505	CHANNEL SEQUENCE NUMBER WRAP VALUES ARE DIFFERENT
AMQ9506	MESSAGE RECEIPT CONFIRMATION FAILED
AMQ9507	CHANNEL...IS CURRENTLY IN-DOUBT
AMQ9526	MESSAGE SEQUENCE NUMBER ERROR FOR CHANNEL
AMQ9543	STATUS TABLE DAMAGED
AMQ9544	MESSAGE WRITTEN TO THE DEAD LETTER QUEUE
AMQ9550	CHANNEL PGM...CANNOT BE STOPPED AT THIS TIME
AMQ9556	CHANNEL SYNCHRONIZATION FILE MISSING OR DAMAGED
AMQ9563	SYNCHRONIZATION FILE CANNOT BE DELETED
AMQ9999	CHANNEL PGM ENDED ABNORMALLY

This means that for each failure we have to determine the cause of inconsistency.

If you need to re-synchronize the channels the following tasks should be performed on local and partner queue managers:

1. Verify the channel and transmission queue.
2. Back out/commit and resolve channel.
3. Reset the channel sequence number.
4. Restart the channel.

The next sections explain the above tasks.

You may also perform re-synchronization with the media recovery command, for example:

```
rcrmqobj -m QM_NAME -t syncfile amqrsyna.dat
```

Refer to 5.4, "How to Re-synchronize the Channel" on page 108 for more information.

OS/2 and Windows NT

Recommendation

- Check if there is a hardware network connection (for example, no hardware problem with the token-ring card) or a local software network connection:

```
PING IP_Code_Local_Queue_Manager  
PING IP_Code_Partner_Queue_Manager
```
- Do not forget to *restart the listener* before checking anything else, such as channels or network connection. The command to start the listener is:
 - For TCP/IP using the default port 1414:

```
runmq1sr -m QM_NAME -t tcp
```
 - For TCP/IP using a different port:

```
runmq1sr -m QM_NAME -t tcp -p PORT_NUMBER
```
 - For SNA LU_6.2

```
runmq1sr -m QM_NAME -t lu62 -n TP_NAME
```
- Check if the queue manager and the channels are ready to work:

```
runmqsc  
dis qmgr all  
ping qmgr  
ping chl(CHL_NAME)
```

5.4.1 Verify Channel and Transmission Queue

Verify the status of all channels and the number of messages held in the transmission queues.

Step 1. Obtain the names of all sender channels and write them down:

```
DIS CHL(*) CHLTYPE(SDR)
```

Step 2. Check each sender channel to find out the connection names and the transmission types (write them down):

```
DIS CHL(CHL_NAME) CONNAME TRPTYPE
```

Step 3. With the following command, check if the channels are stopped:

```
DIS CHSTATUS(CHL_NAME) all
```

- If MCASTAT shows "not running" then the channel is inactive; otherwise stop the channel.
- If the channel is active, look for a JOBNAME. Write the job name down, just in case one of the users calls to find out why he cannot work any longer.

OS/2 and Windows NT

- If CURLUWID and LSTLUWID are zero than the channel is inactive. Otherwise, the channel will be in-doubt when it is started again. You have to resolve the channel. This is discussed later.

Step 4. Find out the names of all transmission queues and write them down:

```
DIS CHL(*) XMITQ
```

Step 5. For all active channels, check if there are messages in the transmission queue:

```
DIS Q(XMITQ_NAME) CURDEPTH
```

If there are messages in the queue, write them down.

Step 6. Ping the channel to check if there is a network connection problem or a remote configuration definition problem.

```
PING CHL(CHL_NAME)
```

Step 7. Start the channel:

```
STA CHL(CHL_NAME)
```

Step 8. When starting the channel you may get error codes. Check them in the queue manager log:

```
C:\MQM\qmgrs\QM_NAME\errors> AMQERR01.LOG
```

Step 9. If you get one of the following error codes you need to resynchronize the channel:

```
AMQ9556 CHANNEL SYNCHRONIZATION FILE MISSING OR DAMAGED.  
AMQ9563 SYNCHRONIZATION FILE CANNOT BE DELETED.
```

Use the media recovery command:

```
rcrmqobj -m QM_NAME -t syncfile amqrsyna.dat
```

This rebuilds the channel synchronization file amqrsyna.dat, because the channel synchronization file is missing or does not correspond to the stored channel information for the queue manager.

5.4.2 Backout/Commit and Resolve Channel

If you get the error code "AMQ9507 CHANNEL IS CURRENTLY IN-DOUBT", you will have to examine the status of the channel and restart it to resolve the in-doubt state.

If after restarting the channel, it is not possible to re-establish the channel connection, then the SENDING END remains in-doubt, as to whether or not

OS/2 and Windows NT

the messages were received. Any outstanding Logical Unit of Work (LUW) needs to be resolved by being backed out or committed.

Recommendation

```
RESOLVE CHL(CHL_NAME) ACTION ( backout or commit )
```

Care must be exercised in the use of this command. If the specified resolution of the SENDING END is not the same as the resolution at the RECEIVING END, messages can be lost or duplicated.

1. If *backout* is used, the messages are backed out; that is, they are RESTORED TO the transmission queue.
2. If *commit* is used, the messages are committed; that is, they are DELETED FROM the transmission queue.

5.4.3 Reset the Sequence Number in the Channel

To synchronize the message sequence number with the partner queue manager in a distributed environment use RESET CHANNEL to reset the message sequence number for a message channel with, optionally, a specified sequence number to be used the next time the channel is started. The command is issued for a SENDER or a SERVER CHANNEL, then in addition to resetting the value at the end at which the command is issued, the value at the other (receiver or requester) end will also be reset to the same value the next time this channel is initiated and re-synchronized.

Use the following to reset a channel:

```
RESET CHL(CHL_NAME) SEQNUM(n)
```

- The new message sequence number must be greater than or equal to 1 and less than or equal to 999 999 999.
- This sequence number is used the next time the channel is started.
- This command can be issued for a channel of any type, except SVRCONN and CLNTCONN channels.
- If the command is used for a SENDER or a SERVER channel, then, in addition to resetting the value at the local end at which the command is issued, the value at the partner end (receiver or requester) will also be reset to the same value the next time this channel is initiated and re-synchronized, if necessary.

5.4.4 Restart the Channel

Step 1. Start the channel with the following command:

```
STA CHL(CHL_NAME)
```

Step 2. Display the status of the channel:

```
DIS CHS(CHL_NAME) ALL
```

Step 3. Display the number of messages in the transmission queue:

```
DIS Q(XMITQ_NAME) CURDEPTH
```

Step 4. If there are no messages in the transmission queue then they have been sent to the partner queue manager.

5.5 How to Re-synchronize the Channel with Media Command

If you have a damaged re-synchronization file use the media recovery command to restore the file:

```
rcrmqobj -m QM_NAME -t syncfile amqrsyna.dat
```

This command can be used after the image of the channel synchronization file has been recorded with the media backup command `rcdmqimg`. This command must be used when the queue manager is running.

5.6 How to Stop the Queue Manager

MQSeries provides four ways to end a queue manager:

5.6.1 Normal Shutdown

The `endmqm` command is used to end (stop) a specified local queue manager, for example:

```
endmqm QM_NAME
```

The queue manager ends after all applications have been disconnected. Any MQI calls currently being processed will be completed. This is the default.

5.6.2 Immediate Shutdown

The parameter `"i"` in the command will end the queue manager immediately:

OS/2 and Windows NT

```
endmqm -i QM_NAME
```

The queue manager stops after it has completed all MQI calls currently being processed. Any MQI requests issued after the command has been issued will fail. Any incomplete logical units of works (LUW) are rolled back when the queue manager is started again.

5.6.3 Preemptive Shutdown

This option is caused by the parameter "p" in the endmqm command:

```
endmqm -p QM_NAME
```

You should use this type of shutdown *only in exceptional circumstances*, for example, when a queue manager does not stop as a result of a normal endmqm command. The queue manager will end without waiting for applications to disconnect or for MQI calls to complete. This can cause unpredictable results in the MQ applications. All processes of the queue manager that fail to stop are terminated 30 seconds after the command is issued.

5.6.4 Using the Windows NT Task Manager

If the queue manager does not successfully end you can stop it manually using the Windows NT Task Manager. Click on the task bar and open the Task Manager window. Figure 31 on page 115 shows this window. End the following MQSeries active tasks:

<i>AMQHASMN.EXE</i>	The logger
<i>AMQHARMN.EXE</i>	Log formatter
<i>AMQZLLPO.EXE</i>	Check point process
<i>AMQZLAAO.EXE</i>	LQM agents
<i>AMQZTRCN.EXE</i>	Trace
<i>AMQZXMA0.EXE</i>	Execution controller
<i>AMQXSSVN.EXE</i>	Shared memory server
<i>RUNMQCHL.EXE</i>	Channel processes
<i>RUNMQLSR.EXE</i>	Listener

OS/2 and Windows NT

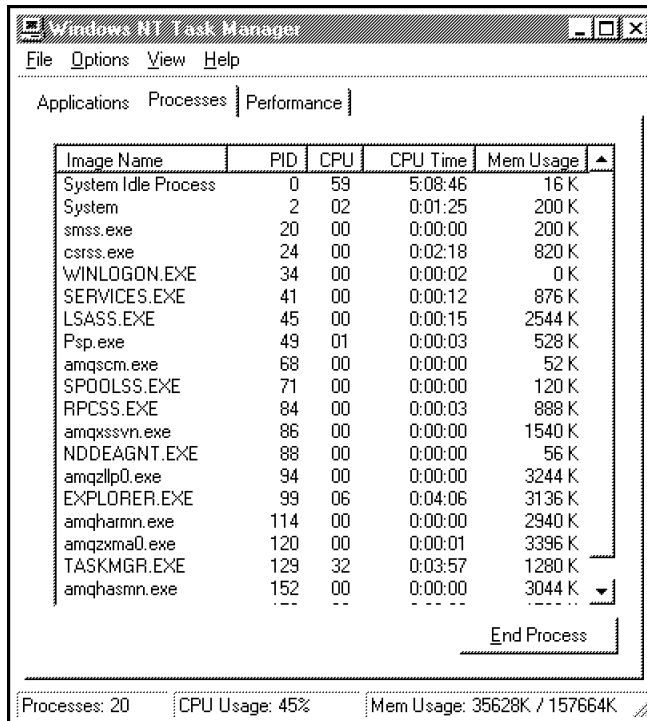


Figure 31. Windows NT Task Manager

The AMQSCM.EXE process cannot be ended from here because this is the MQSeries service called IBMMQSeries and can only be ended from the Services window in the Control Panel; see Figure 32.

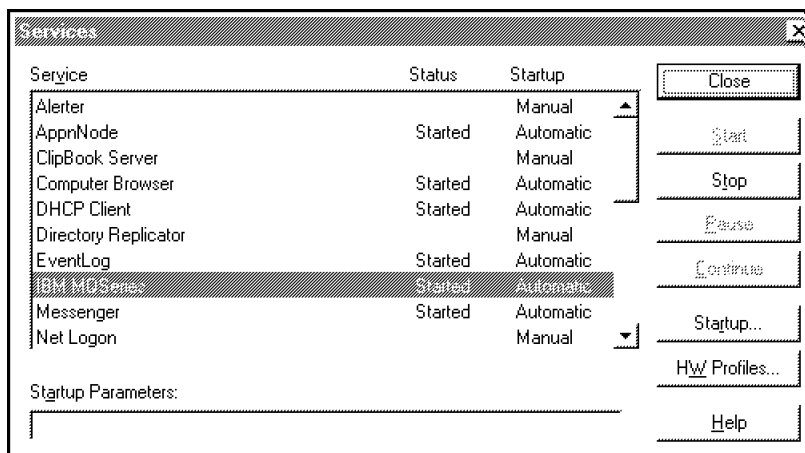


Figure 32. Windows NT Services

OS/2 and Windows NT

A backup can be done with all MQSeries processes in active status but the channels must commit or rollback the changes. The state of the channels after the backup could be abnormal termination.

5.7 MQSeries for Windows NT and OS/2 Backup Procedures

This section describes the two ways of backing up MQSeries objects and files through system and media backup. The following scenarios are discussed:

- System backup of all queue managers
- System backup of one queue manager
- Media backup of one queue manager
- Media backup of synchronization files
- Media backup of a queue

5.7.1 Two Backup Options

There are two kinds of MQSeries backup:

- System backup
- Media backup

5.7.1.1 System Backup

This operation uses the copy or xcopy command and runs under control of the operating system. The queue manager must be stopped.

5.7.1.2 Media Backup

This operation uses the media command `rcdmqimg` under control of a running MQSeries queue manager.

Recommendation

All MQSeries objects must be backed up at a scheduled time in accordance with the other remote queue managers' schedules. This will avoid channel synchronization problems or problems with application data.

5.7.1.3 About Error Code AMQ7044

When you try to use the media command you may get the following error:

AMQ7044 MEDIA RECOVERY NOT ALLOWED

OS/2 and Windows NT

The reason is that the queue manager is using circular logging instead of linear logging.

To check what logging option you are using look in this file:

```
/mqm/qmgrs/QM_NAME/qm.ini
```

If you see the parameter `LogType=CIRCULAR` then you *cannot use* the media command for the queue manager. If you want to use the media command, you have to:

1. Delete the current queue manager.
2. Re-create the new queue manager with linear logging using the following command:

```
crtmqm -ll -q QM_NAME
```

5.7.2 How to Back Up All Queue Managers (System Backup)

This backup is used for MQSeries disaster recovery of one or all queue managers in the same machine. The system recovery needs to restore all MQSeries system files, logs and objects, such as MQSeries product system files, archive logs, active logs, configuration files, source and load modules. To create a complete MQSeries backup you have to back up the `\MQM\` directory structure.

All queue managers must have been ended normally. The following tasks are necessary:

1. Stop all channels.
2. Perform all media backups.
3. Shut down all queue managers.
4. End all listener processes.
5. Back up MQSeries (`\MQM`).
6. Start the queue managers.
7. Start the listeners.
8. Start the channels.
9. If problems, re-synchronize the channels.

The above functions are described, in detail, in the following checklist.

5.7.2.1 Scenario

Step 1. Stop all channels.

For each queue manager in the same machine, you have to do the following:

OS/2 and Windows NT

- a. Stop each channel with the command:
STOP CHL(CHL_NAME)
- b. Check for errors in the queue manager's log:
C:\MQM\qmgrs\QM_NAME\errors> AMQERR01.LOG

Step 2. Perform the media backup for each queue manager:

```
RCDMQIMG -m QM_NAME -t ALL *
```

Or

Back up the queue manager object:

```
RCDMQIMG -m QM_NAME -t q qmgr
```

and all queues:

```
RCDMQIMG -m QM_NAME -t q *
```

and all processes:

```
RCDMQIMG -m QM_NAME -t prcs *
```

and the synchronization file:

```
RCDMQIMG -m QM_NAME -t syncfile amqrsyna.dat
```

Step 3. End the queue managers.

For each MQSeries environment, end the queue manager:

```
ENDMQM QM_NAME
```

Step 4. End the listener processes.

End the listener process for each queue manager with the following command:

```
ENDMQLSR QM_NAME
```

Note: The queue manager must be stopped before the command `endmq|sr` can be issued.

Step 5. Back up all queue managers.

To perform a complete MQSeries backup, you have to back up the MQSeries directory. The directory `\MQM\` contains all queue managers in the same machine.

For example, to do a full system backup to disk E:

- a. Check if there is enough space on the E-drive.
- b. Create a new directory `E:\BACKUP\MQM\` on disk E:
E:\> mkdir BACKUP\MQM\
 - c. Switch to the current MQSeries directory, here on the C-drive:

OS/2 and Windows NT

```
cd C:\MQM
```

d. Now back up MQSeries:

```
C:\MQM> xcopy *.* E:\BACKUP\MQM\ /S
```

The backup can be put in a different disk partition, on tape, on cartridge or any media you choose.

e. Write down the date and time of the backup operation in your Data Processing Operation Control Book.

Step 6. Restart each of the queue managers with the command:

```
STRMQM QM_NAME
```

Step 7. Start the listeners.

Start the listener for each MQSeries environment:

- For TCP/IP that uses the default port 1414:

```
runmq1sr -m QM_NAME -t tcp
```

- For TCP/IP that uses the port P_CODE:

```
runmq1sr -m QM_NAME -t tcp -p P_CODE
```

- For SNA LU_6.2:

```
runmq1sr -m QM_NAME -t lu62 -n TP_NAME
```

Step 8. Start the channels.

Start each channel for each MQSeries environment:

```
STA CHL(CHL_NAME)
```

Check for error messages in the queue manager's log:

```
C:\MQM\qmgrs\QM_NAME\errors> AMQERR01.LOG
```

Step 9. If there are problems, re-synchronize the channels.

If you get errors while starting a channel you will have to re-synchronize the local channel with the remote channel.

Usually, after starting the channel, there are no messages in the transmission queue because these messages are sent to the remote queue manager. Therefore, the queue manager is ready to work.

5.7.3 How to Back Up One Queue Manager (System Backup)

To back up a single queue manager you only need to back up the following two directories:

```
\MQM\qmgrs\QM_NAME\  
\MQM\log\QM_NAME\  

```

OS/2 and Windows NT

This operation can only be performed when the queue manager is stopped normally. The following steps are necessary:

1. Stop all channels.
2. Perform the media backup of the queue manager.
3. End the queue manager.
4. End the listener process.
5. Back up the directories `\MQM\log\QM_NAME\` and `\MQM\qmgrs\QM_NAME\`.
6. Restart the queue manager.
7. Start the listener.
8. Start the channels.
9. If problems, re-synchronize the channel.

5.7.3.1 Scenario

Step 1. Stop all channels.

Stop each channel with the following command:

```
stop chl(CHL_NAME)
```

Check for any errors in the log:

```
C:\MQM\qmgrs\QM_NAME\errors> AMQERR01.LOG
```

Step 2. Back up the queue manager.

Use the following command to back up the queue manager and all of its objects:

```
rcdmqimg -m QM_NAME -t all *
```

Or

Back up the queue manager object:

```
RCDMQIMG -m QM_NAME -t q qmgr
```

and all queues:

```
RCDMQIMG -m QM_NAME -t q *
```

and all processes:

```
RCDMQIMG -m QM_NAME -t prcs *
```

and the synchronization file:

```
RCDMQIMG -m QM_NAME -t syncfile amqrsyna.dat
```

Step 3. End the queue manager:

```
endmqm QM_NAME
```


OS/2 and Windows NT

```
runmq1sr -m QM_NAME -t 1u62 -n TP_NAME
```

Step 8. Start the channels.

Start each channel with the following command:

```
STA CHL(CHL_NAME)
```

Step 9. If problems, re-synchronize the channel. If you get errors while starting a channel you will have to re-synchronize the local channel with the remote channel.

Usually, after starting the channel, there are no messages in the transmission queue because these messages are sent to the remote queue manager. Therefore, the queue manager is ready to work.

5.7.4 How to Back Up One Queue Manager (Media Backup)

To back up a queue manager with the media command, you have to do the following tasks:

1. Stop all channels.
2. Perform the media backup with one of the following commands:

```
rcdmqimg -m QM_NAME -t all *  
rcdmqimg -m QM_NAME -t qmgr
```

3. Start the channels.

Note: The queue manager must be running.

5.7.4.1 Scenario

Step 1. Stop the channels.

Check the channels with the command:

```
DIS CHL(*) ALL
```

Check if each channel is stopped:

```
DIS CHSTATUS (CHL_NAME) ALL
```

Stop each channel with the command:

```
STOP CHL(CHL_NAME)
```

Check if there are some messages in the transmission queue:

```
DIS Q(XMITQ_NAME) CURDEPTH
```

Check if there are errors in the log:

```
C:\MQM\qmgrs\QM_NAME\errors> AMQERR01.LOG
```

Step 2. Back up the queue manager.

OS/2 and Windows NT

Note: The queue manager must be running.

Back up the queue manager and all of its objects with this command:

```
rcdmqimg -m QM_NAME -t all *
```

Alternatively, execute the following commands to separately back up the queue manager and all queues, processes, and the synchronization file:

```
rcdmqimg -m QM_NAME -t qmgr
rcdmqimg -m QM_NAME -t q *
rcdmqimg -m QM_NAME -t prcs *
rcdmqimg -m QM_NAME -t syncfile amqrsyna.dat
```

Step 3. Start the channels.

If the media backup operation was successful, start each of the channels with the command:

```
STA CHL(CHL_NAME)
```

5.7.5 How to Back Up the Synchronization File (Media Backup)

Note: The queue manager must be running.

Execute the following steps:

Step 1. Stop all channels.

Step 2. Back up the synchronization file with the command:

```
rcdmqimg -m QM_NAME -t syncfile amqrsyna.dat
```

Step 3. Start the channels.

5.7.6 How to Back Up a Queue (Media Backup)

Note: The queue manager must be running.

The command `rcdmqimg` is used to write an image of an application queue or system queue, to the log for use in a media recovery.

Example

Back up the application queue MQT1.QL11:

```
rcdmqimg -m MQT1 -t q MQT1.QL11
```

- To back up all queues of queue manager MQT1 execute:

OS/2 and Windows NT

```
rctmqimg -m MQT1 -t q *
```

- To back up all local queues of queue manager MQT1 use the command:

```
rctmqimg -m MQT1 -t ql *
```

- Back up all remote queues of queue manager MQT1 with:

```
rctmqimg -m MQT1 -t qr *
```

- Back up all alias queues of queue manager MQT1 with:

```
rctmqimg -m MQT1 -t qa *
```

- Back up all model queues of queue manager MQT1 with:

```
rctmqimg -m MQT1 -t qm *
```

5.8 MQSeries for Windows NT and OS/2 Recovery Scenarios

This section covers the following topics:

- System and media recovery
- System recovery of a queue manager
- Media recovery of a queue manager
- Media recovery of the channel synchronization file
- Media recovery of a queue

5.8.1 Two Recovery Options

There are two kinds of MQSeries recovery:

- System recovery
- Media recovery

5.8.1.1 System Recovery

This is done with the copy or xcopy command under control of the operating system. The queue manager must be stopped.

5.8.1.2 Media Recovery

This operation uses the media command rctmqobj and runs under control of a running MQSeries queue manager.

5.8.1.3 Reasons for a Recovery

The reasons for an MQSeries recovery in a production environment could be:

- A disaster occurred or the queue manager is damaged.

OS/2 and Windows NT

- There is a problem with the logs, for example, error AMQ7467 or AMQ7468 occurred.
- The channel synchronization file is damaged.
- Objects are damaged (all types of queues).

Recommendation

Before performing the MQSeries recovery operations, check if you:

1. Are authorized to access the MQSeries system
2. Have a large disk space in the machine

5.8.1.4 About Error Code AMQ7044

When you try to use the media recovery command you may get the following error:

```
AMQ7044 MEDIA RECOVERY NOT ALLOWED
```

The reason is that your queue manager uses circular logging instead of linear logging. To check which logging option you use look in this file:

```
\mqm\qmgrs\QM_NAME\qm.ini
```

If you see the parameter `LogType=CIRCULAR` then you *cannot use* the media command. In order to use it you have to:

1. Delete the current queue manager.
2. Re-create a new queue manager with linear logging using the following command:

```
crtmqm -11 -q QM_NAME
```

5.8.2 How to Restore a Queue Manager (System Recovery)

To restore a queue manager, you have to do the following:

1. Stop all channels.
2. Stop the queue manager.
3. End the listener process.
4. Archive the current logs.
5. Restore the queue manager files in `\mqm\log\QM_NAME\` and `\mqm\qmgrs\QM_NAME\`.
6. Restart the queue manager.
7. Start the listeners.

OS/2 and Windows NT

8. Start the channels.
9. If necessary, re-synchronize the channels.

The above functions are described, in detail, in the following scenario.

5.8.2.1 Files Needed for Recovery

You need the following files:

- The most recent queue manager backup and its logs
- The current logs generated by MQSeries after the backup
- The current amqhlctl.lfh file which contains the checkpoint updates of the logging system
- The file amqrsyna.dat which contains channel synchronization information

5.8.2.2 Scenario

1. Stop the channels.

Check if the sender channels are stopped and if any messages are in the transmission queue:

```
DIS CHSTATUS(CHL_NAME) a11
DIS CHL(*) XMITQ
DIS Q(XMITQ_NAME) CURDEPTH
```

Stop each of the channels:

```
STOP CHL(CHL_NAME)
```

If you want to restore all objects of a queue manager which has been backed up several days ago, and you need to get the data of all integrated applications from before and after the time of restore, you have to check the MQSeries remote logs for the number of messages in the transmission queues. This means:

- The MQSeries administrator has to check the MQSeries logs in the local and remote queue managers before the recovery operation.
- The application programs have to check the status of the messages in local or remote queues. This is an application design responsibility.
- The current messages in the transmission queue could be lost if someone resets the channel sequence number after recovery of all old logs and old messages in the transmission queue.

2. End the queue manager:

```
ENDMQM QM_NAME
```

OS/2 and Windows NT

Only if the queue manager is not successfully stopped, can you use one of the following commands:

```
ENDMQM -I QM_NAME  
ENDMQM -P QM_NAME
```

or end the MQSeries tasks via the Task Manager.

3. End the listener.

The queue manager must be stopped before the command `endmq|sr` can be issued. End the listener process with this command:

```
endmq|sr QM_NAME
```

4. Archive the current logs:

- a. Archive your current `\MQM\log\QM_NAME\` directory.
- b. Archive the `amqrsyna.dat` file in the `\MQM\qmgrs\QM_NAME\@ipcc` directory.

5. Restore the queue manager files.

The following directories have to be restored:

```
C:\MQM\qmgrs\QM_NAME\  
C:\MQM\log\QM_NAME\
```

Recommendation

In a production environment, each cause of failure is different.

If there is a problem with the machine or the hard drive that contains the queue manager data, the logs, or both the result may be data loss or data corruption.

You must first analyze the situation and check the directory `\mqm\` for any damage and repair such damage.

- a. Restore all objects in the directory `\MQM\qmgrs\QM_NAME\` from the most recent backup.
- b. Restore the `\MQM\log\QM_NAME\` directory.
- c. Restore the newest archive logs.
- d. Restore the `amqrsyna.dat` file.

Check for errors in the log:

```
C:\MQM\qmgrs\QM_NAME\errors> AMQERR01.LOG
```

Example

OS/2 and Windows NT

Let us restore the queue manager QM_NAME from a system backup file that resides on the F-drive. The queue manager resides on the C-drive.

- 1) Locate the current directories for queue manager QM_NAME on the C-drive. They are:

```
C:\MQM\qmgrs\QM_NAME\  
C:\MQM\log\QM_NAME\  

```

- 2) Clear the current directories by deleting them. You may drag the directory icons into the recycle bin.

- 3) Re-create the directories:

```
C:\> mkdir \MQM\qmgrs\QM_NAME\  
C:\> mkdir \MQM\logs\QM_NAME\  

```

- 4) Copy the backed up files from the backup directory \b\ on the F-drive into the new current directories on the C-drive:

```
Xcopy F:\b\MQM\qmgrs\QM_NAME>*. * C:\MQM\qmgrs\QM_NAME /S  
Xcopy F:\b\MQM\log\QM_NAME>*. * C:\MQM\log\QM_NAME /S  

```

6. Restart the queue manager.

After all the above steps have been completed successfully you now can start the queue manager:

```
strmqm QM_NAME
```

All the objects will be recovered to their last operational state.

7. Start the listener:

- For TCP/IP using the default port 1414:

```
runmq1sr -m QM_NAME -t tcp
```

- For SNA LU_6.2:

```
runmq1sr -m QM_NAME -t lu62 -n TP_NAME
```

8. Start the channels.

Start each channel with the following command:

```
STA CHL(CHL_NAME)
```

9. If necessary, re-synchronize the channels.

If an error occurs when starting a channel you have to re-synchronize the local channel with the remote channel.

5.8.3 How to Recover a Queue Manager (Media Recovery)

To recover a queue manager with the media recovery command you have to do the following:

1. Stop all channels.
2. End the queue manager.
3. End the listener.
4. Restart the queue manager.
5. Perform the media recovery with the command:

```
rcrmqobj -m QM_NAME -t all *
```
6. Start the listener.
7. Start the channels.
8. If necessary, re-synchronize the channels.

5.8.3.1 Scenario

1. Stop the channels.

Obtain the names of the sender channels:

```
DIS CHL(*) TYPE(SDR) ALL
```

Check if each channel is stopped:

```
DIS CHSTATUS (CHL_NAME) ALL
```

If not, stop it:

```
STOP CHL(CHL_NAME)
```

Check if there are any messages in the transmission queue:

```
DIS Q(XMIT_NAME) CURDEPTH
```

If you want to restore all objects of a queue manager which has been backed up several days ago, you have to check if there are any errors in the logs. Also, the application programs have to check the status of the messages in the local and remote queues. This is an application design responsibility.

2. End the queue manager:

```
ENDMQM QM_NAME
```

Only if the queue manager is not successfully stopped, can you use one of the following commands:

```
ENDMQM -I QM_NAME
```

```
ENDMQM -P QM_NAME
```

Alternatively, you can end the queue manager via the Task Manager.

OS/2 and Windows NT

3. End the listener.

The queue manager must be stopped before the command `endmq1sr` can be issued.

```
endmq1sr QM_NAME
```

4. Restart the queue manager:

```
STRMQM QM_NAME
```

5. Perform the media recovery.

Note: The queue manager must be running before issuing the following:

```
rccrmqobj -m QM_NAME -t all *
```

Alternatively, you can recover the queue manager, its objects and the synchronization file separately:

```
rccrmqobj -m QM_NAME -t q *  
rccrmqobj -m QM_NAME -t prcs *  
rccrmqobj -m QM_NAME -t syncfile amqrsyna.dat
```

Check if there are errors recorded in the log:

```
C:\MQM\qmgrs\QM_NAME\errors> AMQERR01.LOG
```

Note: Read the recommendations for `sysem/media` recovery on page 133.

6. Start the listener:

- For TCP/IP using the default port 1414:

```
runmq1sr -m QM_NAME -t tcp
```

- For SNA LU_6.2:

```
runmq1sr -m QM_NAME -t lu62 -n TP_NAME
```

7. Start the channels.

Check if there are no errors in the log: :

```
C:\MQM\qmgrs\QM_NAME\errors> AMQERR01.LOG
```

Start each channel with the following command:

```
STA CHL(CHL_NAME)
```

8. If necessary, resynchronize the channels.

If an error occurs when starting a channel you have to re-synchronize the local channel with the remote channel.

5.8.4 How to Recover the Synchronization File (Media Recovery)

The command `rcrmqobj` is used to re-create the channel synchronization file. This command must be used when the queue manager is running and after an image was recorded in the log.

Example

```
Recover the channel synchronization file:
rcrmqobj -m QM_NAME -t syncfile amqrsyna.dat
```

5.8.4.1 About Error Code AMQ9556

If you get the message "AMQ9556 Channel synchronization file missing or damaged", you will have to perform a media recovery for this system file consisting of the following tasks:

1. Stop all channels.
2. End the queue manager.
3. End the listener.
4. Restart the queue manager.
5. Start the listener again.
6. Perform media recovery with the command:


```
rcrmqobj -m QM_NAME -t syncfile amqrsyna.dat
```
7. Start the channels.

5.8.5 How to Recover a Queue (Media Recovery)

Note: The queue manager must be running.

Example

```
Recover queue MQT1.QL11 of queue manager MQT1 with this command:
rcrmqobj -m MQT1 -t q MQT1.QL11
```

5.8.5.1 About Error Code AMQ8149

If you get the message "AMQ8149 MQSeries object damaged" you will have to perform a media recovery for this object. Here are some examples:

- Recover all queues of queue manager MQT1:


```
rcrmqobj -m MQT1 -t q *
```
- Recover all local queues of queue manager MQT1:


```
rcrmqobj -m MQT1 -t ql *
```

OS/2 and Windows NT

- Recover all remote queues of queue manager MQT1:
rcrmqobj -m MQT1 -t qr *
- Recover all alias queues of queue manager MQT1:
rcrmqobj -m MQT1 -t qa *
- Recover all model queues of queue manager MQT1:
rcrmqobj -m MQT1 -t qm *

Recommendations for System/Media Recovery

1. *Media recovery* with the command `rcrmqobj -t all *` will recover all objects of any type of the queue manager *except* channels.
2. In a *production environment* each failure is different. You must analyze the situation first.
3. If there is a problem with the hard drive on which MQSeries is installed, data may be lost or corrupted.
4. Problems with a disk drive containing either the queue manager data, the log or both can cause data loss or corruption.
5. You must first check the directory structure for any damage and, if necessary, repair such damage. If you lost queue manager data, there is a danger that the queue manager directory structure has also been damaged.

This means that you will need to perform first a *system recovery* and then a *media recovery*:

- a. If so, you have to *stop the queue manager*.
- b. Then you must manually *re-create the directory tree* before you can try to restart the queue manager.
- c. If there is major damage to the directory structure or any damage to the log, *remove all log files* back to the queue manager name level `C:\MQM\qmgrs\QM_NAME\`, including the configuration files, the log, and the queue manager directory.
- d. *Restore the last system backup* to repair the damaged queue manager.
- e. The MQSeries checkpoint file `C:\MQM\qmgrs\QM_NAME\amqh1ct1.1fh` must be restored as part of the queue manager data. This file contains information to determine how much of the data in the log must be applied.
- f. You must have the oldest log file that was required to start the queue manager at the time of the backup and all subsequent log files, available in the log file directory.
- g. Now you can *restart* the queue manager.

OS/2 and Windows NT

Appendix A. Guide to Unplanned Outages on the AS/400 (DASD Loss)

In order for a computer to achieve a rating of at least "Highly Available", outages caused from DASD (disk units) failures must be eliminated. On the AS/400, there are two options:

1. OS/400 DASD mirroring protection
2. AS/400 RAID-5 DASD protection

OS/400 DASD mirroring protection has the distinction of being able to provide the highest level of disk protection available. OS/400 DASD Mirroring can provide redundancy for I/O buses, DASD I/O processors (IOPs), I/O cables, disk unit controller, disk unit power supply and the disk unit itself. OS/400 DASD mirroring allows the system to continue operations in the event one of these protected components fails and allows deferred maintenance and hot-swapping (9337 and AS/400 internal disks only). Although DASD mirroring typically costs more than RAID-5 DASD protection, the level of redundancy selected (for example, eliminating bus and adapter redundancy) can reduce costs. In addition, older AS/400s can implement OS/400 DASD mirroring without replacing disk units or I/O adapters (so long as there is enough storage capacity available to turn DASD mirroring on). Unlike other platforms, the AS/400 allows DASD protection methods to be mixed (for example, DASD mirroring and RAID-5 in the same ASP (Auxiliary Storage Pool)), protecting your DASD investment. OS/400 DASD mirroring is included "no-charge" with all versions of OS/400.

Although RAID-5 DASD tends to be both the industry and AS/400's strategic disk protection method, many very large AS/400 environments with continuous availability requirements select OS/400 DASD mirroring to ensure maximum data path redundancy by reducing dependencies on clustered or dual-system solutions for data protection.

AS/400 RAID-5 DASD protection is the most cost efficient method of protecting AS/400 disk units. Providing disk unit controller, power supply and disk unit protection, RAID-5 allows the system to continue operations in the event one of these components fails and allows deferred maintenance and hot-swapping (9337 and AS/400 internal disks only). AS/400 RAID-5 is the choice for Highly Available and High Availability requirements.

In addition to protecting DASD with OS/400 DASD mirroring or AS/400 RAID-5, additional protection is available via OS/400 Auxiliary Storage Pools (ASPs). These are included in all versions of OS/400 (no charge), ASPs allows disk units be segmented into DASD pools at the user's discretion. ASPs may be used with or without DASD mirroring or RAID-5. For AS/400s

with large DASD capacities, user ASPs provide additional protection by eliminating the need to recover the entire system in the event of a failure in one ASP. When used without some form of DASD protection, recovery time is reduced since only the disk units assigned to the failed ASP must be reloaded. When used with OS/400 DASD mirroring or AS/400 RAID-5, user ASPs provide an additional level of insurance should these protection methods be unable to recover from a disk loss (highly unlikely). For systems with highly available, high availability and continuous availability requirements, all DASD in ASPs should be protected with some form of DASD protection.

ASPs will be a strategic function of AS/400 in the near future for functions such as clustering and Hierarchical Storage Management and will play a big role on the AS/400 in moving past the current 1 terabyte DASD maximum capacity.

Beginning with V3R7, the AS/400 provides the ability to remotely mirror the load source unit to another location on the campus (up to 500 or 2000 meters depending on AS/400 model) for additional protection.

Appendix B. AS/400 DASD Storage Management: ASPs (Auxillary Storage Pools)

An auxiliary storage pool (ASP) is a software definition of a group of disk units on your system. Conceptually, each ASP on your system is a separate pool of disk units for single-level storage. The system spreads data across the disk units within an ASP. If a disk failure occurs, you need to recover only the data in the ASP that contained the failed unit.

All systems have a system ASP. It contains all of the configured disk units that are not assigned to a user ASP. The system ASP must be large enough to hold the operating system and other IBM-supplied objects. One disk unit in the system ASP contains the information that the system needs to start up (IPL). This disk unit is called the load source unit (LSU).

User ASPs are optional. You can assign some disk units to a user ASP and place objects, such as libraries, folders, or journal receivers, in that ASP. Up to 15 user APSs may be created by either adding new disks, using existing disks (moved from another ASP) or both. The maximum number of user ASPs is 16 (system ASP and user ASP 2-16). When you assign the disk units on your system to more than one ASP, each ASP can have different strategies for availability, backup and recovery, and performance.

Auxiliary storage pools are primarily a recovery function. ASPs can reduce the amount of data that you need to recover if a disk failure occurs. Auxiliary storage pools also play a part in availability planning because you specify mirrored protection separately for each ASP on your system. You can combine mirrored protection and device parity protection for each ASP to ensure that all of the disk units on your system are protected.

B.1 Additional Disk Protection for Large AS/400s

Since the AS/400 now offers disk capacities of over a terabyte (TB), user ASPs may be considered as an additional insurance step for data protection. Although extremely rare, should OS/400 DASD mirroring or RAID-5 DASD protection not be able to recover a failed disk unit, the time to restore all data in the single system ASP (ASP-1) will be long. Breaking large DASD configurations into multiple, protected ASPs isolates the failure to that ASP only. Recovery time is reduced since data in other ASPs is still intact.

B.2 Partial DASD Protection and User ASPs

Since DASD protection may be selected by ASP (that is, system ASP has DASD mirroring and user ASP-2 has RAID-5 protection), an ASP could also be configured to have no protection at all (system ASP has DASD mirroring and user ASP-2 has no protection). In most business environments, it is highly recommended that all ASPs have some form of DASD protection implemented.

Note: When a drive fails in a user ASP, and that ASP does not have either OS/400 DASD mirroring or RAID-5 DASD protection implemented, the system will halt and stop processing for all jobs/users on the system, regardless of the ASP they may be accessing. On a busy system, this will usually happen immediately. So while some money may be able to be saved by not implementing DASD protection on a user ASP, availability of other ASPs, and the system, can be impacted should a drive fail in an unprotected ASP.

Appendix C. RAID

RAID (Redundant Array of Independent Disks) applies to the AS/400 only.

Logs should be on RAID (0/1, 1 or 5) DASD.

0/1 Most protection/performance/expense

1 Excellent redundant protection

5 Very good parity protection/cost effective

RAID 0 is striping and increases performance. It does not add redundancy.

RAID 1 is 100% redundant drives.

RAID 5 uses a form of parity. It does not provide redundancy. A common RAID 5 configuration is to have four data drives and one parity drive. For every byte on the four data drives there is a parity byte on the parity drive.

Using an overly simplified example, assume that byte zero of the four data drives contains the following four bytes, 1 2 4 6, and that the parity byte is the sum of the four data bytes. The parity byte would be 13. If you were to lose one (and only one) of the data bytes (drives) it would be a simple matter to determine what that byte was by subtracting the sum of the remaining bytes from the parity byte. This is how RAID 5 works.

Byte#	Drive A	Drive B	Drive C	Drive D	Parity Drive
0000	1	2	3	4	10
0001	2	3	4	1	10
0002	12	1	32	4	49
0003	1	2	4	8	15

Appendix D. Special Notices

This publication is intended to help system administrators and operators to design and administer backup and recovery procedures of MQSeries objects. The information in this publication is not intended as the specification of any programming interfaces that are provided by MQSeries or the operating systems of the platforms discussed. See the PUBLICATIONS section of the IBM Programming Announcement for MQSeries for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

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 Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and

integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX	AIX/6000
Application System/400	AS/400
Enterprise System/3090	Enterprise Systems Architecture/370
Enterprise Systems Architecture/390	ESA/370
ESA/390	IBM
MQSeries	MVS
MVS/ESA	Operating System/2
Operating System/400	OS/390
OS/400	RISC System/6000
RS/6000	S/370
S/390	

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix E. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

E.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How to Get ITSO Redbooks" on page 147.

- *Using MQSeries on the AS/400*, SG24-5236

E.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

E.3 Other Publications

These publications are also relevant as further information sources:

- *MQSeries for AS/400 Administration Guide*, GC33-1956
- *MQSeries System Administration*, SC33-1873

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, Redpieces, and CD-ROMs. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information for customers may be found at <http://www.redbooks.ibm.com/>. and for IBM employees at <http://w3.itso.ibm.com/>.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, Redpieces, materials, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Redbooks Web Site on the IBM Intranet**

Search for, view or download redbooks, Redpieces, or materials or download a printable copy of a redbook from the intranet Redbooks Web Site at:

<http://w3.itso.ibm.com/>

Register for information on workshops, residencies, and redbooks, by selecting the "ITSO Mailing List" button.

- **Hardcopy** —

Order hardcopies in the United States with PUBORDER.

IBMers outside the United States should use the normal hardcopy ordering process for their country.

- **REDBOOKS Category on INEWS**

View redbook, residency and workshop announcements at:

<http://inews.ibm.com/>

Redpieces

Redpieces are redbooks in progress; not all redbooks become Redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows. Look at all Redpieces on the intranet Redbooks Web Site by selecting the "Redpieces" button at: (<http://w3.itso.ibm.com/>).

Materials

Materials are a repository of workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals. Look at all Materials on the intranet Redbooks Web Site by selecting the "Additional Materials" button at: (<http://w3.itso.ibm.com/>).

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, Redpieces, and CD-ROMs) in the following ways:

- **Redbooks Web Site on the World Wide Web**

Search for, view or download redbooks and Redpieces at:

<http://www.redbooks.ibm.com/>

- **Online Orders** — send orders to:

In United States:
In Canada:
Outside North America:

IBMMAIL
usib6fpl at ibmmail
caibmbkz at ibmmail
dkibmbsh at ibmmail

Internet
usib6fpl@ibmmail.com
lmannix@vnet.ibm.com
bookshop@dk.ibm.com

- **Telephone Orders**

United States (toll free)
Canada (toll free)

1-800-879-2755
1-800-IBM-4YOU

Outside North America
(+45) 4810-1320 - Danish
(+45) 4810-1420 - Dutch
(+45) 4810-1540 - English
(+45) 4810-1670 - Finnish
(+45) 4810-1220 - French

(long distance charges apply)
(+45) 4810-1020 - German
(+45) 4810-1620 - Italian
(+45) 4810-1270 - Norwegian
(+45) 4810-1120 - Spanish
(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications
Publications Customer Support
P.O. Box 29570
Raleigh, NC 27626-0570
USA

IBM Publications
144-4th Avenue, S.W.
Calgary, Alberta T2P 3N5
Canada

IBM Direct Services
Sortemosevej 21
DK-3450 Allerød
Denmark

- **Fax** — send orders to:

United States (toll free)
Canada
Outside North America

1-800-445-9269
1-403-267-4455
(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States) or USA International Access Code-408-256-5422 (Outside USA)** — ask for:

Index # 4421 Abstracts of new redbooks
Index # 4422 IBM redbooks
Index # 4420 Redbooks for last six months

- **On the World Wide Web**

IBM Direct Publications Catalog

<http://www.elink.ibm.com/pbl/pbl>

Redpieces

For information so current it is still in the process of being written, look at all Redpieces on the Redbooks Web Site by selecting the "Redpieces" button at: (<http://www.redbooks.ibm.com/>). Redpieces are redbooks in progress; not all redbooks become Redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

First name _____ Last name _____

Company _____

Address _____

City _____ Postal code _____ Country _____

Telephone number _____ Telefax number _____ VAT number _____

• Invoice to customer number _____

• Credit card number _____

Credit card expiration date _____ Card issued to _____ Signature _____

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

List of Abbreviations

AIX	Advanced Interactive Executive (IBM's flavor of UNIX)	LUWID	logical unit of work identifier
AS/400	Application System/400 (IBM)	MVS	multiple virtual storage (IBM System 370 & 390)
APPC	advanced program-to-program communication	MQ	message queuing
ASP	auxiliary storage pool	MQM	MQ queue manager
BSDS	boot strap data set	MSN	message sequence number
DASD	direct access storage device	OS/2	Operating System/2
IBM	International Business Machines Corporation	OS/400	operating system for AS/400
IFS	AS/400 internal file system	PC	personal computer
IP	internet protocol	PSID	page set identifier
ITSO	International Technical Support Organization	RAID	redundant array of independent disks
KSDS	keyed-sequence data set	RBA	relative byte address
LDS	linear dataset	TCP	transmission control protocol
		UR	unit of recovery
		URL	Uniform Resource Locator
		VSAM	virtual storage access method (IBM)

Index

A

- abbreviations 151
- acronyms 151
- active log 10
- AIX 71
- AMQ7044 82, 90, 116, 125
- AMQ7460 5, 41, 45
- AMQ7462 5, 41, 45
- AMQ7467 5, 74, 103, 104
- AMQ7468 5, 74, 103, 105
- AMQ8149 96, 131
- AMQ9507 79
- AMQ9556 78, 111, 131
- AMQ9563 78, 111
- AMQ99556 96
- AMQAJRN 39
- AMQALMP4 41
- AMQERR01.LOG 5
- amqhctl.lfh 75, 108
- AMQIQES4 44
- AMQRJRN 39
- amqrsyna.dat 75, 108
- archive 1
- archive log 11, 15, 17
- archive log (AIX) 91
- archive log (MVS) 10
- archiving 1, 10
 - AS/400 39
 - MVS 3
- AS/400 37
 - backup procedures 47
 - journal receivers 39
 - journaling 37
 - logging 3
 - record/recover media image 7
 - recovery scenarios 49
 - resources to back up 7
 - save object definitions 4

B

- back up logs 105
- background information (MVS) 9
- backing up 1
- backout 32, 79, 112
- backup 1
 - all queue managers 83, 117
 - AS/400 libraries 47
 - data libraries (AS/400) 48
 - directories 74, 107
 - files 74, 107
 - MQ OS/400 objects 47
 - MVS 12
 - object definitions 21
 - one queue manager 85, 86, 119, 122
 - OS/2 116
 - page sets 14, 17
 - procedures (AS/400) 47
 - queue 88, 123
 - queue manager objects 15
 - queues 22
 - scenarios 16
 - synchronization file 87, 123
 - two options 81, 116
 - UNIX 81
 - what? 74, 107
 - Windows NT 116
- backup and recovery
 - procedures (MVS) 16
 - tips for MVS 14
- backup plan 12
- bibliography 145
- bootstrap data set 9, 11
- BSDS 3, 9, 11, 13
 - recover 26

C

- change journal 46
- change receivers 46
- channel 31
 - display status 32

- channel (*continued*)
 - messages 33
 - reset 34, 79, 112
 - resolve 32, 79, 111
 - restart 34, 80, 113
 - sequence number error 34
 - start 33
 - stopped 32
- channel initiator restart 32
- channel re-synchronization 31
 - manually 32
 - OS/2 108
 - UNIX 76, 80, 113
 - Windows NT 108
 - with media command 80, 113
- checkpoint 11, 13
 - force (AS/400) 45
- checkpoint files 75, 108
- checkpoint record 11
- CHGJRN 46
- circular 74, 103
- circular logging 2, 74, 103
- clear library 51
- commit 32, 79, 112
- commit point 13
- configuration files 73, 101
- corrupted logs 23
- create objects 4
- create queue manager 58, 72, 100
- CSQI024I 5, 7, 16, 17
- CSQI025I 5, 7, 16, 17
- CURLUWID 78, 111
- current journal attributes 43
- current receiver 43

D

- data libraries
 - backup 48
- data sets (MVS) 9
- dd name 16
- dead-letter queue 34
- default queue manager 72, 100
- define
 - local queue 9
 - storage class 9

- delete journal 56
- delete receiver 56
- delete receivers 44, 46, 49
- delete user space 45
- directories
 - OS/2 107
 - UNIX 75
 - Windows NT 107
- disaster recovery 28
- disk crash (AS/400) 50
- display
 - channel status 32
 - system operator msg 45
- DLTRCV 42
- dual BSDS 11
- dual logging 10, 15

E

- earliest RBA 15, 16
- end listener 118
- end queue manager 80, 113
- error codes 76, 82, 108, 116
- event viewer 105

F

- files
 - checkpoint 75, 108
 - UNIX 75
 - Windows NT 108
- forcing MQSeries checkpoint 45
- fuzzy backup 17

H

- hints (AS/400) 64

I

- I/O error (MVS) 26
- image
 - record/recover manually 6
 - when written 2
- immediate shutdown 81, 113
- in-doubt 78, 79, 111

incomplete state 13
indoubt UR 15

J

JCL
AMS VERIFY function 25
back up page sets 19
backup queue 22
define new page sets 18
MAKEDEF command of CSQUTIL 21
print log map utility 16
recover page set 20
RESETPAGE function 25
restore active log 24
restore from backup page set 26
restore queue 22
journal
 change 46
journal attributes 43
 current 43
journal management 39
journal receiver 3
 local and remote 39
journal receiver chain 67
journaling 37
 concept 38

K

KSDS 11

L

LDS 9
libraries (AS/400) 47
 restore 52
linear 74, 103
linear logging 2, 72, 74, 90, 100, 103
listener 110
ll parameter 3, 72, 100
lock 60
log 1
 active 15
 archive 15
 back up 105
 directories 73

log (*continued*)
 recover 23
 required for recovery 74, 103
 size 72, 100
log data sets 9, 10
log files 73, 102
log messages 5
log records 11
log size 101
logging 74, 102
 AS/400 3
 circular 2
 dual 10, 15
 linear 2, 72, 100
 MVS 3
 UNIX, OS/2, NT 3
lost logs 23
LSTLUWID 78, 111
LUWID 31

M

MAKEDEF 21
manage receivers 42
MAXARCH 15
MCASTAT 78, 110
media backup 82, 116
 one queue manager 86, 122
 queue 88, 123
 synchronization file 87, 123
media image
 record/recover manually 6
media recovery 89, 124
message
 persistent 13
 recovery 16
 sequence number 79, 112
 sequence number error 34
messages
 regarding logs 5
 where to find them 5
mirrored log 3
MQ.INI 73, 102
MQS.INI 73, 102
MQSeries
 and OS/400 journaling 39
 create objects 4

MQSeries (*continued*)
object definitions 4
 backup 21
re-create objects 4
re-install 50
recover objects 16
resources to backup 7
restart (MVS) 25
save objects. 4
startup 105

MVS 9
archiving 3
background information 9
backup 12
backup and recovery
 procedures 16
channel re-synchronization 31
data sets 9
disaster recovery 28
logging 3
record/recover media image 7
recovery scenarios 22
resources to back up 7
restart and recovery 13
save object definitions 4
tips 14

N

normal shutdown 80, 113

O

object damage 64
object definitions 4
 recommendations
 AS/400 4
 MVS 4
 UNIX, OS/2, NT 4
objects (AS/400) 47
 backup 47
old messages 105
orderly shutdown 66
OS/2 99, 108
 backup procedures 116
 logging 3
 record/recover media image 6

OS/2 (*continued*)
recovery scenarios 124
resources to back up 7
 save object definitions 4
OS/400 journaling 3, 37
out dd 19
overview 1

P

page set
 JCL to recover 20
 recover
 scenarios 19
page set association 16
page set control record 11
page set zero 3, 4, 10
page sets 9, 12
 backup 17
 backup scenarios 16
 define new 17
 JCL to back up 19
 JCL to define 18
 recover 26
 take backup copies 14
ping 77, 110
ping channel 78, 111
point of recovery 16
preemptive shutdown 81, 114

Q

QMQRDATA 39
queue 1, 12
 backup (AIX) 88
 backup (NT) 123
 backup and restore 22
 damaged 64
 logs required 6
 media backup (AIX) 88
 media backup (NT) 123
 recover 96, 131
queue manager
 back up objects 15
 backup (AIX) 83, 85, 86
 backup (NT) 117, 119, 122
 cannot start 59

- queue manager (*continued*)
 - cannot stop 59
 - create 58, 72, 73, 100, 101
 - deleted (AS/400) 56
 - end 45, 80, 113
 - media backup (AIX) 86
 - media backup (NT) 122
 - messages when stopped 33
 - quiesce 17
 - recover (AIX) 93
 - recover (NT) 129
 - resources to protect 7
 - restart 19, 45
 - restore 90, 91, 125
 - restore (AS/400) 54
 - system backup (AIX) 83, 85
 - system backup (NT) 117, 119
- quiesce 17
 - description 66

R

- RAID 139
- RBA 11
- RBA (earliest) 15, 16
- rcdmqimg 6
- RCDMQMIMG 7, 41, 45
- RCRMQMIMG
 - description 64
- RCRMQMOBJ
 - edia image contained in the log. 7
 - rcrmqobj 6, 80, 113
- re-create MQ objects 4
- re-create object 6
- re-synchronize channels manually 32
- receiver 3, 44
 - attributes 43
 - chain 67
 - change 46
 - current 43
 - delete 42, 46, 49
 - delete redundant 44
 - reduce 42
 - required 40, 42
 - size 66
- recommendation
 - restore queue manager 92

- recommendations
 - backup (MVS) 21
 - CONNNAME (MVS) 31
 - create queue manager 73, 101
 - hardware failure 77, 110
 - media backup 116
 - minimize data loss (MVS) 10
 - re-sync channels (MVS) 34
 - receivers (AS/400) 38
 - recovery (AIX) 89
 - resolve channel 79, 112
- record image 105
- record media image 6, 42, 64
- recover
 - corrupted logs 23
 - damaged BSDS 26
 - damaged object (AS/400) 64
 - damaged page set 26
 - from disk crash (AS/400) 50
 - from system failure 60
 - lost BSDS 26
 - lost logs 23
 - messages 16
 - objects 16
 - page set 13, 19
 - page set (JCL) 20
 - queue 96, 131
 - queue manager 93, 129
 - synchronization file 95, 131
- recovery 1
 - reasons 88
 - required logs 74, 103
 - two options 89, 124
 - UNIX 72
- recovery log 13
- recovery scenarios
 - AS/400 49
 - MVS 22
 - OS/2 124
 - UNIX 88
 - Windows NT 124
- REDO pass 13
- replay transactions 12
- reset channel 34, 79, 112
- RESETPAGE function 24

- resolve channel 32, 79, 111
- restart and recovery
 - MVS 13
- restart channel 80, 113
- restart listener 110
 - start listener 110
- restart queue manager 45
- restart TCP/IP 31
- restore 1
 - all objects (AS/400) 56
 - damaged AS/400 MQM 54
 - deleted AS/400 MQM 56
 - in other system 60
 - journals 54
 - library 52
 - objects 51, 62
 - queue manager 90, 91, 125
 - queues 22
 - specific objects (AS/400) 57
- restore and recovery 1
- restore order 51, 54
- resync channels 31
- RSTLIB 52
- RSTOBJ 51
- runmqlsr 110
- runmqsc 4, 32, 73, 101

S

- save MQ objects 60
- save object definitions 4
- sender channel 91
- sequence number 79, 112
- sequence number error 34
- shared memory (AS/400)
- sign off and on 66
- size of receivers 66
- SMS management 15
- start channel 33
- startup 105
- stop queue manager 17, 80, 113
- storage class 3, 9
- strmqm 45
- STRMQMMQSC 4
- synchronization file
 - backup (AIX) 87
 - backup (NT) 123

- synchronization file (*continued*)
 - media backup (AIX) 87
 - media backup (NT) 123
 - recover 95, 131
- synchronize channel 76, 80, 108, 113
- syncpoint 13
- system backup 82, 116
 - all queue managers 83, 117
 - one queue manager 85, 119
- system failure 59
- system managed receivers 46
- system recovery 89, 124
- SYSTEM.CHANNEL.SYNCQ 31

T

- task manager 114
- TCP/IP restart 31
- terminology 1
- tidying up shared storage 44
- tips for journaling 43
- tips for MVS 14
- transaction log 1, 12
- transmission queue 78

U

- uncommitted UR 15
- unit of work 79, 112
- unit-of-recovery log record 11
- units of recovery 13
- UNIX 71
 - backup procedures 81
 - logging 3
 - record/recover media image 6
 - recovery scenarios 88
 - resources to back up 7
 - save object definitions 4

V

- verify channel
- verify xmitq

W

- Windows NT 99
 - backup procedures 116
 - logging 3
 - record/recover media image 6
 - recovery scenarios 124
 - resources to back up 7
 - save object definitions 4
- work with ...
 - channels 48
 - job 46
 - journal attributes 43
 - objects 45, 49
- WRKJRNA 43, 46

X

- xcopy MQSeries 119

ITSO Redbook Evaluation

MQSeries Backup and Recovery
SG24-5222-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at <http://www.redbooks.ibm.com>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?

Customer **Business Partner** **Solution Developer** **IBM employee**
 None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

What other redbooks would you like to see published?

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

