



Nastel Workgroup Server

Administrator's Guide

Version 6

Document Number: NAVCLAS-ADM 667.005

PUBLISHED BY:

R&D DEPARTMENT

NASTEL TECHNOLOGIES, INC.

88 SUNNYSIDE BLVD, SUITE 101

PLAINVIEW, NY 11803

Copyright © 2001-2020 by: Nastel Technologies

ALL RIGHTS RESERVED. NO PART OF THE CONTENTS OF THIS DOCUMENT MAY BE PRODUCED OR TRANSMITTED IN ANY FORM, OR BY ANY MEANS WITHOUT THE WRITTEN PERMISSION OF NASTEL TECHNOLOGIES.

TITLE: **Nastel Workgroup Server Administrator's Guide**

DOCUMENT RELEASE DATE: **July 2020**

NASTEL DOCUMENT NUMBER: **NAVCLAS-ADM 667.005**

PRINTED IN THE UNITED STATES OF AMERICA

CONFIDENTIALITY STATEMENT: THE INFORMATION WITHIN THIS MEDIA IS PROPRIETARY IN NATURE AND IS THE SOLE PROPERTY OF NASTEL TECHNOLOGIES, INC. ALL PRODUCTS AND INFORMATION DEVELOPED BY NASTEL ARE INTENDED FOR LIMITED DISTRIBUTION TO AUTHORIZED NASTEL EMPLOYEES, LICENSED CLIENTS, AND AUTHORIZED USERS. THIS INFORMATION (INCLUDING SOFTWARE, ELECTRONIC AND PRINTED MEDIA) IS NOT TO BE COPIED OR DISTRIBUTED IN ANY FORM WITHOUT THE EXPRESSED WRITTEN PERMISSION FROM NASTEL TECHNOLOGIES, INC.

ACKNOWLEDGEMENTS:

THE FOLLOWING TERMS ARE TRADEMARKS OF NASTEL TECHNOLOGIES CORPORATION IN THE UNITED STATES OR OTHER COUNTRIES OR BOTH: AUTOPILOT/IT, AUTOPILOT-WMQ, AUTOPILOT /WEB, M6 WEB SERVER, M6 FOR WMQ, AUTOPILOT M6, MQCONTROL.

THE FOLLOWING TERMS ARE TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES OR OTHER COUNTRIES OR BOTH: IBM, MQ, WIN-OS/2, AS/400, DB2, AND AIX.

THE FOLLOWING TERMS ARE TRADEMARKS OF HEWLETT-PACKARD IN THE UNITED STATES OR OTHER COUNTRIES OR BOTH: OPENVIEW, HP-UX.

COMPAQ, THE COMPAQ LOGO, ALPHASERVER, COMPAQ INSIGHT MANAGER, CDA, DEC, DECNET, TRUCLUSTER, ULTRIX, AND VAX REGISTERED IN U.S. PATENT AND TRADEMARK OFFICE. ALPHA AND TRU64 ARE TRADEMARKS OF COMPAQ INFORMATION TECHNOLOGIES GROUP, L.P IN THE UNITED STATES AND OTHER COUNTRIES

SNMPC, SNMPC, WORKGROUP, AND SNMPC ENTERPRISE ARE TRADEMARKS OF CASTLE ROCK COMPUTING IN THE UNITED STATES OR OTHER COUNTRIES, OR BOTH.

JAVA AND THE JAVA LOGOS ARE TRADEMARKS OF ORACLE CORPORATION AND/OR ITS AFFILIATES. IN THE UNITED STATES OR OTHER COUNTRIES, OR BOTH.

INSTALLANYWHERE IS A REGISTERED TRADEMARK OF ZEROG SOFTWARE IN THE UNITED STATES OR OTHER COUNTRIES, OR BOTH.

THIS PRODUCT INCLUDES SOFTWARE DEVELOPED BY THE APACHE SOFTWARE FOUNDATION ([HTTP://WWW.APACHE.ORG/](http://www.apache.org/)), INCLUDING DERBY DATABASE SERVER. THE JAKARTA PROJECT" AND "TOMCAT" AND THE ASSOCIATED LOGOS ARE REGISTERED TRADEMARKS OF THE APACHE SOFTWARE FOUNDATION.

INTEL, PENTIUM AND INTEL486 ARE TRADEMARKS OR REGISTERED TRADEMARKS OF INTEL CORPORATION IN THE UNITED STATES, OR OTHER COUNTRIES, OR BOTH.

MICROSOFT, WINDOWS, WINDOWS XP, THE WINDOWS LOGOS, AND SQL SERVER ARE REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION.

UNIX IS A REGISTERED TRADEMARK IN THE UNITED STATES AND OTHER COUNTRIES LICENSED EXCLUSIVELY THROUGH X/OPEN COMPANY LIMITED.

"LINUX" AND THE LINUX LOGOS ARE REGISTERED TRADEMARKS OF LINUS TORVALDS, THE ORIGINAL AUTHOR OF THE LINUX KERNEL. ALL OTHER TITLES, APPLICATIONS, PRODUCTS, AND SO FORTH ARE COPYRIGHTED AND/OR TRADEMARKED BY THEIR RESPECTIVE AUTHORS.

ORACLE IS A REGISTERED TRADEMARK OF ORACLE CORPORATION AND/OR ITS AFFILIATES.

SYBASE IS A TRADEMARK OF SYBASE, INC. ® INDICATES REGISTRATION IN THE UNITED STATES OF AMERICA.

MYSQL IS A REGISTERED TRADEMARK OF MYSQL AB IN THE UNITED STATES, THE EUROPEAN UNION AND OTHER COUNTRIES.

SCO CUSA, SCO DOCTOR, SCO DOCTOR FOR NETWORKS, SCO DOCTOR LITE, SCO GLOBAL ACCESS, SCO MPX, SCO MULTIVIEW, SCO NIHONGO OPENSERVR, SCO OK, THE SCO OK LOGO, SCO OPENSERVR, SCO OPEN SERVER, SCO PORTFOLIO, SCO POS SYSTEM, SCO TOOLWARE, AND THE WORLD NEVER STOPS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF CALDERA INTERNATIONAL, INC. IN THE U.S.A. AND OTHER COUNTRIES. ALL RIGHTS RESERVED.

OTHER COMPANY, PRODUCT, AND SERVICE NAMES MAY BE TRADEMARKS OR SERVICE MARKS OF OTHERS.

Table of Contents

Chapter 1: Introduction	13
1.1 <i>How this Guide is Organized</i>	13
1.2 <i>History of this Document</i>	14
1.2.1 User Feedback	15
1.3 <i>Related Documents</i>	15
1.4 <i>Release Notes</i>	15
1.4.1 What is New in M6 for WebSphere MQ?	15
1.5 <i>Use of this Document</i>	16
1.5.1 Intended Audience	16
1.6 <i>System Requirements</i>	16
1.7 <i>Terms and Abbreviations</i>	17
1.8 <i>Technical Support</i>	18
1.9 <i>Conventions</i>	18
1.10 <i>AutoPilot M6 for WMQ Installation Support (32-bit/64-bit)</i>	18
Chapter 2: About M6-WMQ	20
2.1 <i>M6-WMQ Functionality</i>	20
2.1.1 Management Needs of WebSphere MQ	20
2.1.2 Management Tasks Performed by M6-WMQ	20
2.2 <i>M6-WMQ's System Components</i>	21
2.2.1 Workgroup Server	22
2.2.2 M6Managed Node(CEP Server)	23
2.2.3 M6-WMQ Experts	23
2.2.4 Automation Policies	23
2.2.5 Business Views	24
2.2.6 Agents	24
2.2.7 Front-end Management Applications	24
2.2.8 Plug-ins	25
2.2.9 Components by Platform	25
Chapter 3: Management Planning	26
3.1 <i>Essential Management Issues</i>	26
3.2 <i>Management Groups</i>	26
3.3 <i>Basic Management Structure</i>	27
3.3.1 Problems with Basic Management Structure	27
3.4 <i>Fault-tolerant Management Structure</i>	28
3.4.1 Rules for Overlapping Groups	28
3.5 <i>Planning M6-WMQ Configuration</i>	30
3.5.1 Basic Management Configuration	30
Chapter 4: Configuring M6-WMQ	31
4.1 <i>Basic Configuration Tasks</i>	31
4.2 <i>Defining an M6-WMQ Workgroup Server</i>	31
4.2.1 Defining a Workgroup Using the nsqsqlmk Utility (SQL Database)	31

4.3	<i>Configuring a Workgroup Server</i>	34
4.3.1	Obtaining M6-WMQ License	34
4.3.2	Installing M6-WMQ License	35
4.3.3	Checking Workgroup Server Limit Parameters	36
4.3.4	Modifying Limit Parameters using M6-WMQ Explorer	37
4.3.5	Configuring Workgroup Server Discovery Policy	37
4.3.6	Configuring Workgroup Server Event Policy	38
4.3.7	Configuring Workgroup Server Accounting and Statistics Log Sizes	38
4.4	<i>M6-WMQ Configuration Files</i>	39
4.4.1	Location of M6-WMQ Configuration Files	39
4.4.2	Workgroup Configuration File (mqgroup.ini)	39
4.4.3	Verify Syntax of Configuration Files	40
4.5	<i>Configuring WebSphere MQ Nodes</i>	41
4.5.1	Setting M6-WMQ Environment Variable	41
4.5.2	Including a WMQ Agent in a Workgroup	41
4.5.3	Registering WMQ Agents Under Workgroup Servers	41
4.5.4	Setting up WMQ Agents Configuration Parameters	42
4.5.5	Configuring Remote Queue Manager Connections Using the Connection Manager	44
Chapter 5: Deleted		50
Chapter 6: Managing the WMQ Network		52
6.1	<i>Configuration Management</i>	52
6.1.1	Defining, Modifying or Deleting Objects	52
6.1.2	Methods for Defining Objects	52
6.1.3	Defining Objects with MQSC	52
6.1.4	Using Interactive MQSC	53
6.1.5	Using MQSC Scripts for Queue Manager (Fault) Recovery	53
6.1.6	Using MQSC Configuration Snapshots for (Fault) Recovery	54
6.1.7	Creating and Removing Queue Managers	54
6.1.8	Executing User Action as a Detached Process	55
Chapter 7: Maintaining M6-WMQ		56
7.1	<i>Enabling/Disabling Trace for Workgroup and WMQ Agents</i>	56
7.2	<i>Optimizing Network Speed</i>	57
Chapter 8: M6-WMQ Security		58
8.1	<i>AutoPilot M6-based User Authentication</i>	58
8.2	<i>OS-based User Authentication</i>	58
8.3	<i>User Authorization (“+u” Parameter)</i>	58
8.3.1	How it Works	59
8.4	<i>Editing Security Statements in permits.ini Files</i>	61
8.4.1	Object Specification	62
8.4.2	Other Field Specifications	63
8.4.3	Effective User Name Statements	64
8.4.4	User Authority Statements	65
8.4.5	TRUST Statements	70
8.4.6	Audit Statements	70
8.5	<i>Examples of permits.ini Statements</i>	71
8.6	<i>Use Case Examples</i>	76
8.6.1	Use Case Example 1	76

8.6.2	Use Case Example 2	79
8.7	<i>FirewallAdministration</i>	82
8.8	<i>WebSphere MQ Access Requirements</i>	83
8.9	<i>Local Context Access to WebSphere MQ Objects</i>	83
Chapter 9: Using M6-WMQ API		90
9.1	<i>M6-WMQ API Services</i>	90
9.1.1	Coordinating Data Formats	91
9.2	<i>Developing Management Applications</i>	93
9.2.1	Building Client/Server Applications	93
9.2.1	WebSphere MQ-based Applications	94
Chapter 10: Integrating the M6-WMQ Ping Utility		96
10.1	<i>AutoPilot M6 Configuration</i>	97
10.1.1	Deploying Process Wrapper	97
10.1.2	Configuring Process Wrapper	98
10.2	<i>Measuring Message Flow Performance</i>	100
10.3	<i>Monitor Message Metrics</i>	103
Chapter 11: Performance Tuning		104
Chapter 12: Z/OS Accounting and Statistics (ZAS) Collector		106
12.1	<i>Configuring NSQZAS</i>	108
12.2	<i>Starting NSQZAS</i>	110
12.3	<i>Stopping/Restarting NSQZAS</i>	110
12.4	<i>Mapping Between Accounting and Statistics PCF Messages on Z/OS and Distributed Platforms</i>	111
12.5	<i>Best Practices for ZAS Collection</i>	128
12.6	<i>Initializing SMF Data Sets</i>	130
12.7	<i>SMF Commands</i>	131
12.7.1	Running SMF Command from SDSF	131
12.7.2	Displaying SMF Status	131
12.7.3	Activating SMF for Recording Record Types 116	131
12.7.4	Starting SMF	132
12.7.5	Displaying Trace Status	132
12.7.6	Starting Trace	132
12.7.7	Stopping Trace	132
12.7.8	Switching the Active SMF Data Set	133
12.7.9	Running the SMF Exit Manually	133
12.8	<i>Useful WMQ Commands</i>	134
12.9	<i>Utilities</i>	135
12.9.1	How to Process an SMF Dump File	135
12.9.2	How to Use AMQSMON to Print Formatted Accounting and Statistics PCF Messages	136
12.9.3	How to Manually Dump an SMF Data Set	140
12.9.4	How to Manually Process an Existing SMF Dump File	141
12.9.5	How to Print Formatted SMF Records	143
12.10	<i>SMF Exit JCL Listing</i>	144

Chapter 12A: nsqprid Utility	146
12A.1 Description	146
12A.2 Command	146
12A.3 Sample Output	146
Chapter 12B: nsqmigr Utility	147
12B.1 Parameters	147
12B.2 Get / Put File	147
12B.3 Recreating a WGS Database	147
12B.3.1 Imported / Exported Fields	148
12B.3.2 Example Get/Put WGS nodes.info File	149
12B.4 Remote Queue Managers	149
12B.4.1 Bulk Importing Remote Queue Managers (-p parameter)	150
12B.4.2 Bulk Exporting Remote Queue Managers (-g parameter)	150
12B.4.3 Example Get/Put Remote Queue Managers Commands and Info File	150
12B.5 Remote EMS Servers	151
12B.5.1 Bulk Importing Remote EMS Servers (-p parameter)	152
12B.5.2 Bulk Exporting Remote EMS Servers (-g parameter)	152
12B.5.3 Example Get/Put Remote EMS Server Commands and Info File	152
Chapter 13: Troubleshooting	155
13.1 Identifying Common Problems	155
13.2 Troubleshooting a WMQ Agent	155
13.2.1 Checking WMQ Agent Parameters	158
13.3 Troubleshooting a Workgroup Server	158
13.3.1 Checking Workgroup Server Parameters	158
13.4 Getting a Spool File from the AS/400	159
Appendix A: References	161
A.1 Nastel Documentation	161
A.2 IBM Documentation	161
Appendix B: Conventions	163
B.1 Typographical Conventions	163
B.2 Naming Conventions	164
B.2.1 Component Naming Conventions	164
Appendix C: Executing User-Defined Scripts	165
C.1 Script Naming Conventions	165
C.2 Locations and Scripts	174
C.3 How Scripts are Invoked	175
C.4 Script Examples	176
Appendix D: Differences Between Platforms	181
D.1 Supported Platforms	181
D.1.2 Specific Features by Platform	181

Appendix E: M6-WMQ Component Summary	183
<i>E.1 Workgroup Server</i>	183
E.1.1 Platforms	183
E.1.2 Command Line Arguments	183
<i>E.2 WMQAgent</i>	185
E.2.1 Platforms	185
E.2.2 Command-Line Arguments	186
<i>E.3 Deleted</i>	187
<i>E.4 MMF Server</i>	187
E.4.1 Platforms	187
E.4.2 Command-Line Arguments	188
<i>E.5 MQ Event Publisher</i>	188
E.5.1 Platforms	188
E.5.2 Command-Line Arguments	188
<i>E.6 M6-WMQ z/OS PCF Command Server</i>	191
E.6.1 Platforms	191
E.6.2 Command-Line Arguments	191
<i>E.7 MQSC Utility</i>	192
E.7.1 Platforms	192
E.7.2 Command-Line Arguments	192
<i>E.8 nsqping (Ping) Utility</i>	193
E.8.1 Platforms	193
E.8.2 Command-Line Arguments	193
<i>E.9 nsqcode Utility</i>	194
E.9.1 Platforms	194
E.9.2 Additional Information: Using nsqcode	194
<i>E.10 Connection Manager</i>	195
E.10.1 Platforms	195
E.10.2 Command-Line Arguments	195
<i>E.11 Log Adapter</i>	197
E.11.1 Platforms	197
E.11.2 Command-Line Arguments	197
<i>E.12 Message Management Utility</i>	199
E.12.1 Platforms	199
E.12.2 Command-Line Arguments	199
E.12.2 Message Management Commands	200
<i>E.13 Queue Monitor Utility</i>	208
E.13.1 Platforms	208
E.13.2 Command-Line Arguments	208
<i>E.14 Message Get/Put Utility</i>	209
E.14.1 Platforms	209
E.14.2 Command-Line Arguments	209
E.14.3 Options to Get Messages from Queue to File	210
E.14.4 Options to Put Messages from File to Queue	211
<i>E.15 SQL Database Creation Utility</i>	216
E.15.1 Platforms	216
E.15.2 Command-Line Arguments	216
<i>E.16 Remote Shutdown Utility</i>	217

E.16.1	Platforms	217
E.16.2	Command-LineArguments	217
<i>E.17</i>	<i>Trigger Message Utility</i>	<i>218</i>
E.17.1	Platforms	218
E.17.2	Command-LineArguments	218
<i>E.18</i>	<i>WMQ Event Adapter</i>	<i>219</i>
E.18.1	Platforms	219
E.18.2	Command-LineArguments	219
<i>E.19</i>	<i>NSQZAS</i>	<i>220</i>
E.19.1	Platforms	220
E.19.2	Command-LineArguments	220
Appendix F: M6-WMQ Objects		221
Appendix G: Deleted		223
Appendix H: WMQ Agent Run-Time Messages		225
Appendix I: Workgroup Server Run-Time Messages		229
Appendix J: Deleted		235
Appendix K: WMQ Agent z/OS Error Codes and Messages		237
Appendix L: Exit Codes When M6-WMQ Runs as Service on Windows		251
Glossary		253

Figures

FIGURE 1-1. AUTOPILOT M6 FOR MIDDLEWARE (APMW) INSTALLATION SUPPORT (32-BIT/64-BIT)	19
FIGURE 2-1. M6-WMQ SYSTEM ARCHITECTURE	21
FIGURE 2-2. MQ SYSTEM METRICS	22
FIGURE 2-3. QUEUE MANAGER PROPERTIES	23
FIGURE 3-1. MANAGING WEBSPHERE MQ NODES BASIC MANAGEMENT STRUCTURE	27
FIGURE 3-2. FAULT-TOLERANT MANAGEMENT OF WEBSPHERE MQ	28
FIGURE 3-3. USING M6-WMQ FOR SNMP-BASED MANAGEMENT OF WEBSPHERE MQ	30
FIGURE 4-1. REMOTE QUEUE MANAGER CONNECTIONS DIALOG BOX	45
FIGURE 4-2. ADD QUEUE MANAGER CONNECTION DIALOG BOX – GENERAL	46
FIGURE 4-3. ADD QUEUE MANAGER CONNECTION DIALOG BOX – COMMUNICATION	46
FIGURE 4-4. ADD QUEUE MANAGER CONNECTION DIALOG BOX – SSL	47
FIGURE 6-1. M6-WMQ MQSC UTILITY START COMMANDS	53
FIGURE 8-1. TWO-LEVEL SECURITY CHECKING BY M6-WMQ	60
FIGURE 8-2. SERVER TO AGENT COMMUNICATION	82
FIGURE 9-1. RELATION BETWEEN M6-WMQ AND API SERVICES	90
FIGURE 9-2. ROLE OF THE M6-WMQ API	93
FIGURE 9-3. MANAGING WEBSPHERE MQ DIRECTLY USING APPLICATION DEVELOPED IN M6-WMQ API	94
FIGURE 10-1. QUEUE PING UTILITY INTEGRATION PATH (NSQPING)	96
FIGURE 10-2. DEPLOYING PROCESS WRAPPERS	97
FIGURE 10-3. PROCESS WRAPPER CONFIGURATION PROPERTIES	98
FIGURE 10-4. PROCESS WRAPPER GENERAL CONFIGURATION PROPERTIES	99
FIGURE 10-5. PROCESS WRAPPER UDP OPTION CONFIGURATION PROPERTIES	99
FIGURE 10-6. PUBLISHED MESSAGE PERFORMANCE METRICS	103
FIGURE 12-1. ZAS FLOW DIAGRAM	107
FIGURE 12-2. SMFPRMXX MEMBER	108
FIGURE 12B.1 – NSQMIGR UTILITY	147
FIGURE 12.B.3.2-A. EXAMPLE GET / PUT WGS FILE	149
FIGURE 12B.4.3-A. EXAMPLE REMOTE QUEUE MANAGERS INFO FILE	151
FIGURE 12B.5.3-A. EXAMPLE REMOTE EMS SERVERS INFO FILE	153
FIGURE E-1. NSQCODE DEFINITIONS	194
FIGURE E-2. MMF COMMAND LINE INTERFACE	201
FIGURE E-3. MQRC COMMAND	204
FIGURE E-4. SAMPLE DUMP OF EXPORTED MESSAGE FILE	215

Tables

TABLE 1-1. DOCUMENT HISTORY	14
TABLE 2-1. M6-WMQ COMPONENTS BY PLATFORM	25
TABLE 4-1. POSSIBLE CRITERIA FOR CREATING MANAGEMENT WORKGROUPS	31
TABLE 4-2. M6-WMQ CONFIGURATION FILES	39
TABLE 4-3. WMQ AGENT PARAMETERS	42
TABLE 7-1. M6-WMQ POTENTIAL PROBLEMS AND SOLUTIONS	56
TABLE 8-1. M6-WMQ USER AUTHORIZATION OBJECT TYPES.....	65
TABLE 8-2. WEBSPHERE MQ PCF CODES.....	84
TABLE 8-3. M6-WMQ EXTENDED PCF CODES.....	87
TABLE 9-1. M6-WMQ API, SERVICES AVAILABLE	90
TABLE 10-1. PROCESS WRAPPER EXPERT: GENERAL	98
TABLE 10-2. PROCESS WRAPPER EXPERT: UDP OPTIONS	99
TABLE 10-3. NSQPING COMMAND LINE OPTIONS AND USAGES.....	102
TABLE 11-1. PERFORMANCE TUNING SUGGESTION	104
TABLE 12-1. MQI ACCOUNTING MESSAGE PARAMETERS.....	111
TABLE 12-2. QUEUE ACCOUNTING MESSAGE PARAMETERS	114
TABLE 12-3. QUEUE STATISTICS MESSAGE.....	118
TABLE 12-4. MQI STATISTICS PARAMETERS.....	121
TABLE 12-5. QUEUE STATISTICS MESSAGE.....	123
TABLE 12-6. CHANNEL STATISTICS MESSAGE.....	126
TABLE 13-1. COMMON PROBLEMS ENCOUNTERED RUNNING M6-WMQ	155
TABLE A-1. NASTEL DOCUMENTATION	161
TABLE B-1. TYPOGRAPHICAL CONVENTIONS	163
TABLE B-2. AUTOPILOT M6 RELATED NAMING CONVENTIONS.....	164
TABLE B-3. M6 FOR WEBSPHERE MQ NAMING CONVENTIONS	164
TABLE C-1. PCF CATEGORIES INTEGER VALUES	165
TABLE C-2. INTEGER VALUES FOR PCF EVENTS/REASON CODES (FROM HEADER FILE CMQC.H).....	166
TABLE C-3. INTEGER VALUES FOR EXTENDED PCF EVENTS/REASON CODES.....	173
TABLE D-1. PLATFORMS SUPPORTED BY M6-WMQ	181
TABLE D-2. PLATFORM SPECIFIC M6-WMQ FEATURES	181
TABLE E-1. WORKGROUP SERVER COMMAND LINE ARGUMENTS.....	184
TABLE E-2. WMQ AGENT COMMAND-LINE ARGUMENTS	186
TABLE E-3. DELETED	187
TABLE E-4. START MMF SERVER COMMAND-LINE ARGUMENTS.....	188
TABLE E-6. STARTING MQ EVENT PUBLISHER COMMAND-LINE ARGUMENTS	189
TABLE E-7. STOPPING MQ EVENT PUBLISHER COMMAND-LINE ARGUMENTS.....	190
TABLE E-8. M6-WMQ z/OS PCF COMMAND SERVER COMMAND-LINE ARGUMENTS	191

TABLE E-9. MQSC UTILITY COMMAND-LINE ARGUMENTS	192
TABLE E-10. PING UTILITY COMMAND-LINE ARGUMENTS	193
TABLE E-11. CONNECTION MANAGER COMMAND-LINE ARGUMENTS	195
TABLE E-12. LOG ADAPTER COMMAND LINE ARGUMENTS	197
TABLE E-13. MESSAGE MANAGEMENT UTILITY COMMAND LINE ARGUMENTS	199
TABLE E-13A. MQPMO PUT MESSAGE OPTIONS	205
TABLE E-14. QUEUE MONITOR UTILITY COMMAND LINE ARGUMENTS	208
TABLE E-15. MESSAGE GET/PUT UTILITY COMMAND LINE ARGUMENTS	209
TABLE E-16. SQL DATABASE CREATION UTILITY COMMAND-LINE ARGUMENTS.....	216
TABLE E-17. REMOTE WORKGROUP SERVER SHUTDOWN COMMAND-LINE ARGUMENTS	217
TABLE E-18. REMOTE WMQ AGENT SHUTDOWN COMMAND-LINE ARGUMENTS	218
TABLE E-19. TRIGGER MESSAGE UTILITY COMMAND-LINE ARGUMENTS	218
TABLE E-20. WMQ EVENT ADAPTER COMMAND-LINE ARGUMENTS.....	219
TABLE E-21. ZAS COMMAND-LINE ARGUMENTS	220
TABLE F-1. M6-WMQ OBJECTS AND DESCRIPTIONS.....	221
TABLE F-2. WEBSPHERE MQ OBJECTS USED BY M6-WMQ COMPONENTS	222
TABLE H-1. RUN-TIME MESSAGES RETURNED BY WMQ AGENTS (SHEET 1 OF 3)	225
TABLE H-1. RUN-TIME MESSAGES RETURNED BY WMQ AGENTS (SHEET 2 OF 3)	226
TABLE H-1. RUN-TIME MESSAGES RETURNED BY WMQ AGENTS (SHEET 3 OF 3)	227
TABLE I-1. RUN-TIME MESSAGES RETURNED BY WORKGROUP SERVER	229
TABLE K-1. WMQ AGENT Z/OS ERROR CODES AND MESSAGES (SHEET 1 OF 14).....	237
TABLE K-1. WMQ AGENT Z/OS ERROR CODES AND MESSAGES (SHEET 2 OF 14).....	238
TABLE K-1. WMQ AGENT Z/OS ERROR CODES AND MESSAGES (SHEET 3 OF 14).....	239
TABLE K-1. WMQ AGENT Z/OS ERROR CODES AND MESSAGES (SHEET 4 OF 14).....	240
TABLE K-1. WMQ AGENT Z/OS ERROR CODES AND MESSAGES (SHEET 5 OF 14).....	241
TABLE K-1. WMQ AGENT Z/OS ERROR CODES AND MESSAGES (SHEET 6 OF 14).....	242
TABLE K-1. WMQ AGENT Z/OS ERROR CODES AND MESSAGES (SHEET 7 OF 14).....	243
TABLE K-1. WMQ AGENT Z/OS ERROR CODES AND MESSAGES (SHEET 8 OF 14).....	244
TABLE K-1. WMQ AGENT Z/OS ERROR CODES AND MESSAGES (SHEET 9 OF 14).....	245
TABLE K-1. WMQ AGENT Z/OS ERROR CODES AND MESSAGES (SHEET 10 OF 14).....	246
TABLE K-1. WMQ AGENT Z/OS ERROR CODES AND MESSAGES (SHEET 11 OF 14).....	247
TABLE K-1. WMQ AGENT Z/OS ERROR CODES AND MESSAGES (SHEET 12 OF 14).....	248
TABLE K-1. WMQ AGENT Z/OS ERROR CODES AND MESSAGES (SHEET 13 OF 14).....	249
TABLE K-1. WMQ AGENT Z/OS ERROR CODES AND MESSAGES (SHEET 14 OF 14).....	250
TABLE L-1. EXIT CODES WHEN M6-WMQ APPLICATIONS RUN AS SERVICES ON WINDOWS.....	251

This Page Intentionally Left Blank

Chapter 1: Introduction

Welcome to the *Nastel Workgroup Server Administrator's Guide*. This guide explains how to use the Nastel workgroup server, a complete management solution for IBM MQ.

1.1 How this Guide is Organized

- [Chapter 1:](#) Introductory information about this document and M6-WMQ.
- [Chapter 2:](#) Descriptions of how M6-WMQ and WebSphere MQ work and interact.
- [Chapter 3:](#) Information for planning the management of WebSphere MQ.
- [Chapter 4:](#) Instructions, direction and data for configuring M6-WMQ.
- [Chapter 5:](#) Deleted.
- [Chapter 6:](#) Outlines the methods of using the information gathered by M6-WMQ to manage the WebSphere MQ network.
- [Chapter 7:](#) Describes potential maintenance issues in M6-WMQ.
- [Chapter 8:](#) Outlines Security in M6-WMQ.
- [Chapter 9:](#) Outlines how you can extend and enhance M6-WMQ functionality using the M6-WMQ Application Programming Interface (API)
- [Chapter 10:](#) Provides details for integrating M6-WMQ Ping utility.
- [Chapter 11:](#) Provides performance tuning suggestions.
- [Chapter 12:](#) Describes the z/OS Accounting and Statistics Collector (ZAS).
- [Chapter 13:](#) Outlines troubleshooting techniques and advice for M6-WMQ.
- [Appendix A:](#) Detailed list of all reference information required for the operation of M6-WMQ.
- [Appendix B:](#) Conventions used in M6-WMQ and Nastel documentation.
- [Appendix C:](#) Outlines the use of user-defined scripts.
- [Appendix D:](#) Defines the feature differences between platforms.
- [Appendix E:](#) Summarizes the component differences between supported platforms.
- [Appendix F:](#) Lists and describes M6-WMQ and WebSphere MQ objects that are created by the M6-WMQ components.
- [Appendix G:](#) Deleted.
- [Appendix H:](#) Lists run-time messages returned by the WMQ Agents.
- [Appendix I:](#) Lists run-time messages returned by the workgroup servers.
- [Appendix J:](#) Deleted.
- [Appendix K:](#) Lists WMQ Agent z/OS error codes and messages.
- [Appendix L:](#) Lists exit codes when M6-WMQ applications run as services on Windows.
- [Glossary:](#) Listing of unique and common acronyms, words and definitions.

1.2 History of this Document

Release Date	Document Number	Versio	Summary
January 2008	M6WMQ 600.001	600.00	Initial release for WebSphere 6.0
August 2008	M6WMQ 600.002	600.00 2	Addition of Z/OS Accounting and Statistics Collector and general updates.
June 2009	M6WMQ 610.001	6.1	Update for WebSphere 6.1
March 2010	M6WMQ 610.002	6.1	Update for WebSphere 6.1.6
June 2010	M6WMQ 620.001	6.2	Support for MQ7
August 2010	M6WMQ 620.002	6.2.10	Update nsqsqlmk Utility (SQL Database) procedures
May 2011	M6WMQ 620.003	6.2.10	Errata
October 2011	M6WMQ 620.003	6.2.10	Errata
March 2012	M6WMQ 620.004	6.2.10	Errata
March 2013	M6WMQ 650.001	6.5	Mantis 2791, 6936, and 7918
May 2013	M6WMQ 650.002	6.5	Errata
June 2013	M6WMQ 650.003	6.5	Update to Kerberos-based user authentication
November 2013	M6WMQ 651.001	6.5.1	Mantis 8943
May 2014	M6WMQ 652.001	6.5.2	Mantis 8109, 9225, 9456
August 2014	M6WMQ 653.001	6.5.3	Document number change to coincide with software version.
September 2014	M6WMQ 653.002	6.5.3	Errata (Mantis 9910)
November 2015	M6WMQ 655.001	6.5.5	Errata (Mantis 10940, 11734, 11943) Remove proxy management (Mantis 11944)
March 2016	M6WMQ 656.001	6.5.6	Mantis 10659, 11526, 12866
May 2016	M6WMQ 656.002	6.5.6	Errata (Mantis 13131, 12986)
January 2017	M6WMQ 658.001	6.5.8	Ability to browse encrypted message headers (Mantis 14628), added +E WGS Command Line Argument (Mantis 11825), ability to collect MQ system metrics (Mantis14749).
July 2017	M6WMQ 658.002	6.5.8	Update Nastel's phone number and street address
August 2017	M6WMQ 660.001	6.6	Update for WebSphere 6.6
February 2019	M6WMQ 660.002	6.6	Errata
May 2019	M6WMQ 660.003	6.6	Update Figure 1-1
September 2019	M6WMQ 660.004	6.6	Update Figure 1-1
January 2020	M6WMQ 660.005	6.6	Add Chapter 12B, nsqmigr Utility. Add Model Queue content to section 4.5.5. Update document title and year.

February 2020	M6WMQ 660.006	6.6	Format update to Chapter 12B.
March 2020	NAVCLAS-ADM 667.001	6.7	Update version number and Chapter 1.
April 2020	NAVCLAS-ADM 667.002	6.7	Add +E command line argument to table E2.
April 2020	NAVCLAS-ADM 667.003	6.7	Remove +E command line argument from table E1.
June 2020	NAVCLAS-ADM 667.004	6.7	Fix image issue.
July 2020	NAVCLAS-ADM 667.005	6.7	Updates to Figure 1-1.

1.2.1 User Feedback

Nastel encourages all users of AutoPilot M6 to submit comments, suggestions, corrections and recommendations for improvement for all AutoPilot M6 documentation. Please email your comments support@nastel.com. You will receive a written response, along with status of any proposed change, update, or correction.

1.3 Related Documents

Complete listings of documents related to M6-WMQ can be found in [Appendix A](#).

1.4 Release Notes

See README.HTM file and README_INSTALL.txt on installation media.

1.4.1 What is New in M6 for WebSphere MQ?

Below are the new features in AutoPilot M6 for WMQ version 6.5.

- Support for WebSphere MQ 7.1/7.5
- A single install for APWMQ, APODWMQ, APOD Security Manager, and AP experts.
- New License – A new license is required for version 6.5. It is required for users of APODWMQ and users running agents on z/OS.
- Support for new WMQ Object attributes
- Support for new Telemetry Channel Connections

Below are the new features in M6-WMQ version 6.2.

- Administration of WMQ Topic and Subscription objects
- New attributes for existing WMQ objects

Below are the new features in AutoPilot M6 for WMQ version 6.1.

- WMQ Authority records are now saved in the M6-WMQ database for Windows, UNIX, and iSeries platforms.
- Set of WMQ object attributes displayed in object listing is now configurable.

Below is a list of new features in M6-WMQ version 6.0.

- Enhanced M6-WMQ Explorer
 - Integrated multi-web view support
 - Consolidate all web views into one tool
 - WMQ 6.0 configuration management support
 - Improved object refresh support

- Color coded objects that have not been refreshed
- See outdated objects quickly
- CMDB: WMQ Configuration Management Database
 - SQL: Oracle, DB2, Microsoft SQL Server, and MySQL
 - Maintain all WMQ objects and their definitions in SQL database
 - Store configuration changes and audit trail
 - Integrate with existing CMDB tools
 - Use existing reporting and data mining tools to generate reports
- Web-based Security Manager
 - Import existing permissions (from permits)
 - Manage all permissions from a web based GUI
 - Consolidate all access via one web based interface
- Web Message Manager
 - Manage WMQ messages from a web based GUI
 - Great for development and production support
 - No need to install consoles
- Enhanced support for WebSphere MQ 5.3 and 6.x
 - Support for Listeners and Services object types
 - Enhanced administration capabilities
 - Support for 6.0 extended object attributes and parameters
 - Support for new 6.0 event types
- Improved performance & scalability
 - Consolidate all Workgroup Servers into a single data repository (CMDB)
 - Improved WMQ agent performance (64 bit support)
 - Improved Workgroup Server performance
- Native 64-bit platform support
- Improved Licensing Model: CPU based

1.5 Use of this Document

This document provides clear, concise instructions for the configuration and deployment of M6-WMQ and its components. Use of this document is intended for Administrators and Developers of Nastel's M6-WMQ.

1.5.1 Intended Audience

The Administrator's Guide is intended for:

- System administrators who manage WebSphere MQ networks
- Developers who want to add their own functionality to M6-WMQ
- Readers who have a general understanding of middleware (especially WebSphere MQ), network protocols, and systems management. Users and administrators will find this guide useful for planning and customizing the WebSphere MQ management structure.

1.6 System Requirements

All software and hardware requirements are defined in *M6 for WebSphere MQ Installation Guide*. M6-WMQ's system requirements vary by component and platform. For information about your platform, see

the appropriate platform chapter in the *M6 for WebSphere MQ Installation Guide*, included on the M6 for WebSphere MQ installation media and downloadable version from our FTP site.

1.7 Terms and Abbreviations

A list of Terms and Abbreviation used in this document is located in the [Glossary](#).

1.8 Technical Support

If you need additional technical support, you can contact Nastel by telephone or e-mail. To contact Nastel technical support by telephone, call **800-963-9822 ext. 1**, if you are calling from outside the United States dial **1-516-801-2100**. To contact Nastel technical support by e-mail, send a message to support@nastel.com. You can also contact Nastel support via the support website. Contact your M6-WMQ Administrator for access information. To access the Nastel automated support system (user ID and Password are required) go to: <http://support.nastel.com/>. Contact your local M6-WMQ Administrator for further information.

1.9 Conventions

Refer to [Appendix B](#) for conventions used in this guide.

1.10 Nastel Navigator Installation Support (32-bit/64-bit)

See [Figure 1-1](#) for Nastel Navigator Installation Support. This platform is for the workgroup server and databases supported. Agents may be available for other versions of operating systems and middleware versions. You should contact their support representatives if your platform is not listed.

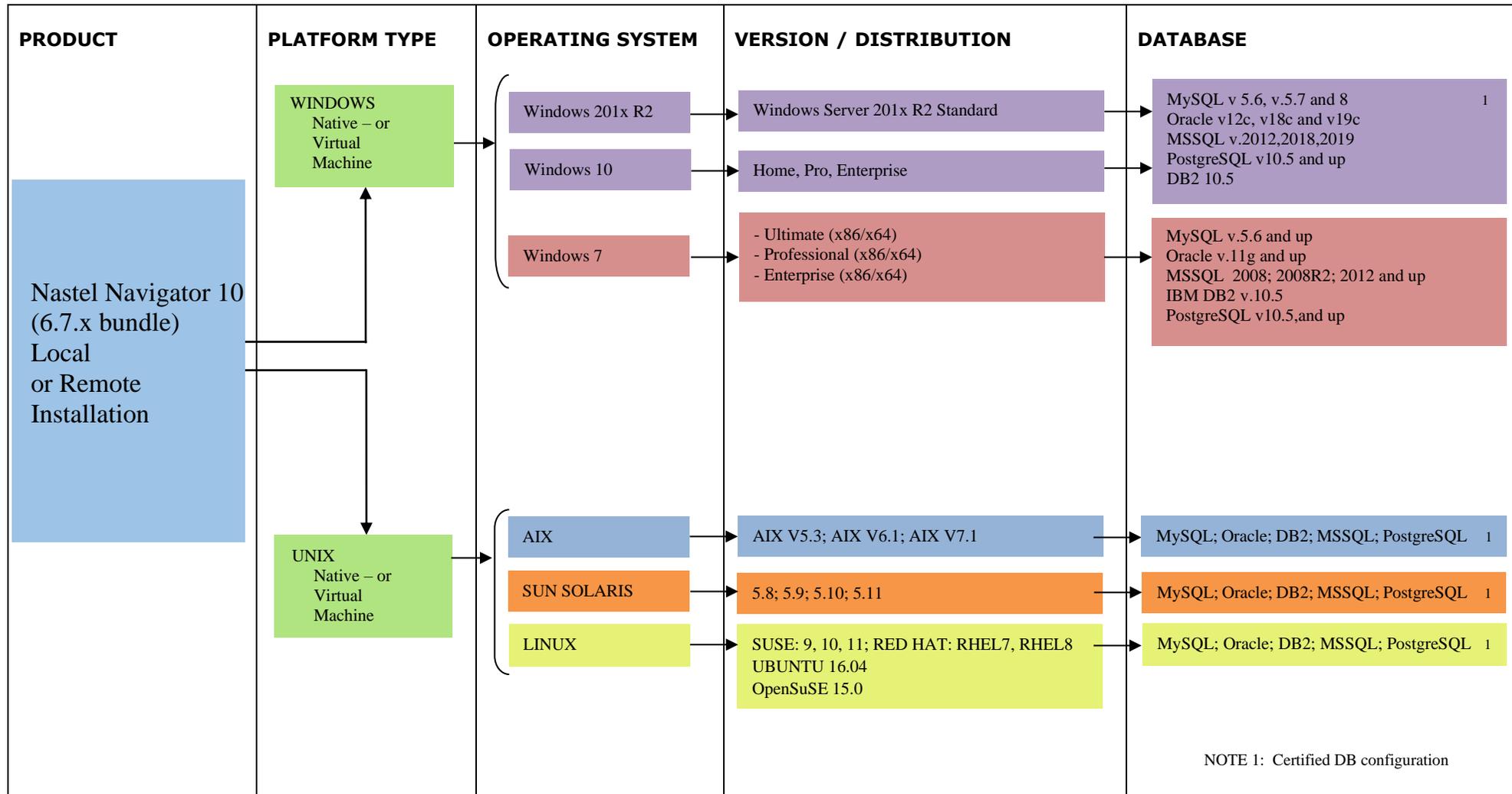


Figure 1-1. Nastel Navigator Installation Support (32-bit/64-bit)

Chapter 2: About M6-WMQ

This chapter provides an overview of M6-WMQ and its components.

2.1 M6-WMQ Functionality

Nastel's M6-WMQ enables users to identify and resolve the problems associated with managing IBM's WebSphere MQ. WebSphere MQ itself is an unsupervised communications backbone used for critical applications on various platforms.

2.1.1 Management Needs of WebSphere MQ

When administrators deploy a WebSphere MQ network, they need a management tool that:

- Ensures uninterrupted operation of each WebSphere MQ component
- Discovers and corrects WebSphere MQ system failures
- Prevents performance bottlenecks and conflicts that might affect business applications.

2.1.2 Management Tasks Performed by M6-WMQ

To meet the management needs of WebSphere MQ, M6-WMQ:

- Collects management data, statistics, and events (including performance and fault events) through intelligent agents.
- Forwards management information to a central management presentation platform with a Management Information Base (MIB) browser.
Example: HP OpenView, Tivoli, or NetView. For a full list of acceptable front-end platforms with MIB browsers, contact [Nastel Support](#). You can also use a front-end without a MIB browser, if you want.
- Takes corrective actions when WebSphere MQ problems are identified.

2.2 M6-WMQ's System Components

M6-WMQ has multi-tier client/server architecture, consisting of the following components:

- Workgroup Servers
- WMQ Agents, Message Servers and Event Publish-Subscribers
- Front-end administration applications (Web-based and console)

WMQ Experts monitors data and updates even with the command server down. The Universal Agent, on the other hand, uses PCF commands and will not update with the command server down. M6-WMQ system architecture is shown in Figure 2-1.

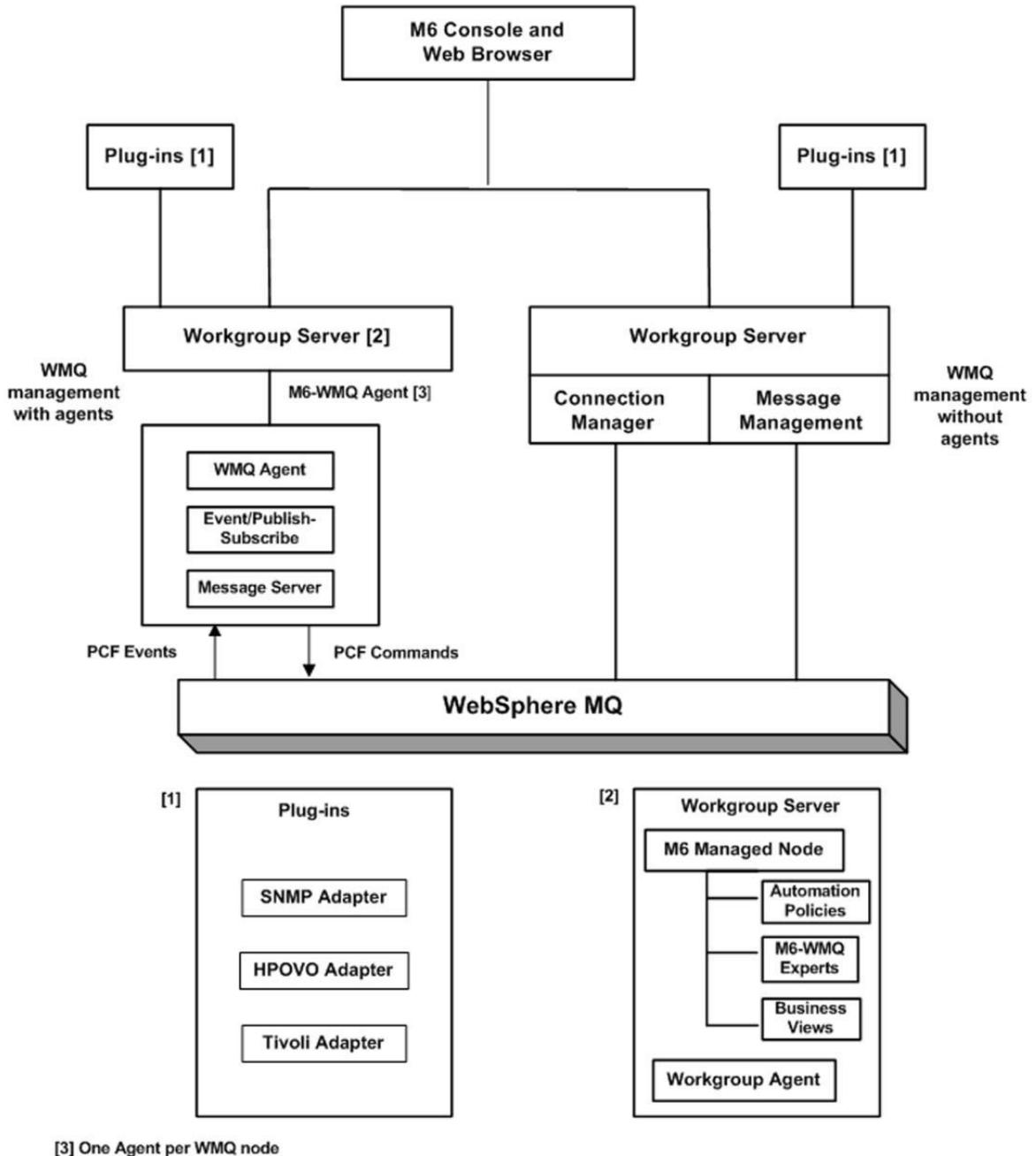


Figure 2-1. M6-WMQ System Architecture

2.2.1 Workgroup Server

M6-WMQ places WebSphere MQ nodes into groups. A workgroup server (WGS) oversees a group of WMQ nodes. A workgroup server consists of two agents, M6 managed node/CEP server with M6 SU6 or higher) and workgroup. The workgroup server collects information about a group of registered WMQ nodes and stores it in its database. This server makes it readily available for management applications. It can be used with or without local agents. The workgroup server is used to issue WebSphere MQ commands (PCF/MQSCs) remotely, without setting up WebSphere MQ channels. It is capable of carrying out instructions at any level of organization, including:

- Workgroup
- Node
- Queue manager
- Queue, channel, process, listener, namelist, authentication information, service, etc

	NOTE:	A single workgroup server can manage up to 2048 WebSphere MQ nodes, with a maximum of 100 queue managers on each node.
---	--------------	--

Recording MQ System Metrics and publishing them as Facts is supported when using WGS10 and an Agent running 6.5.8 or later against a MQ V9 or later queue manager. In the figure below you see **QM_VER9** and underneath it **METRICS** which is further divided into **CPU** and **DISK** where the system metrics MQ is reporting are published.

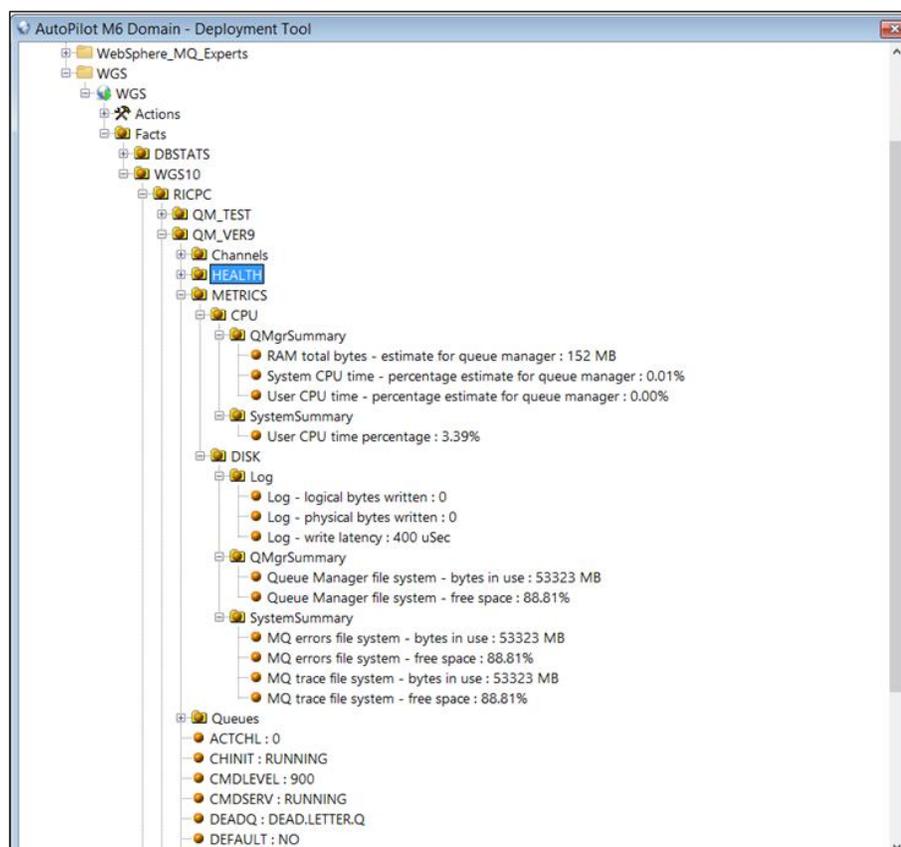


Figure 2-2. MQ System Metrics

This feature **Collect MQ SysMetric** is enabled in the GUI under the **General** tab of Queue Manager Properties in version 6.5.9 or later

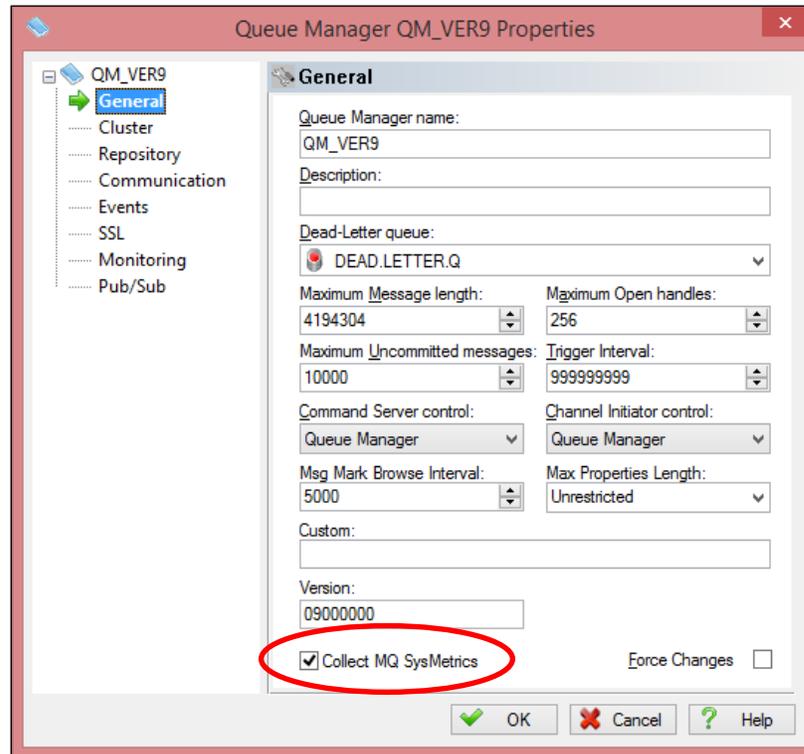


Figure 2-3. Queue Manager Properties

2.2.2 M6Managed Node(CEP Server)

The M6 Managed Node provides M6-WMQ monitoring experts, automation policies, and business views. It is installed for every workgroup server and monitors only one workgroup.

2.2.3 M6-WMQExperts

The official name for application specific module is Expert.

- Experts monitor and control specific applications
- Experts interface with managed applications using application specific interfaces. (Example: Experts for WebSphere application server, WebSphere MQ, Windows 2000, 2003, and XP.)
- Experts do not make judgments about the managed environment.
- Expert focus on data collection and execution of actions.

2.2.4 Automation Policies

Policies use proactive automation rules and procedures to perform actions on one or more management services:

- Perform actions on one or more management services
- Policies are capable of writing information to database, log files, and other data stores
- Policies subscribe to facts and act on fact changes, time events, or other conditions
- Usually specific to monitored applications
- Managers manage policies. (Example: Scheduling Manager, WebSphere MQ Manager, etc.). Any number of relevant policies can be assigned to a manager. (Example: Alert or execute an action based on a condition/event.)

2.2.5 Business Views

- Collection of rules (also known as sensors) that define a desired state of an eBusiness environment. The business views can be tailored to present the information in the form that best suites your needs.
- Proactive user-defined policies that:
 - Correlate facts/events
 - Automate and alert
 - Generate user-defined events
 - Collect historical data for future analysis
- Defined using domain console
- Deployable as policies within any defined network of managers (running in the background)
- Once deployed, available for remote viewing on M6 Web Console.

2.2.6 Agents

The *M6-WMQ Agent* is an intelligent agent that runs on a WebSphere MQ node. It executes management commands on one or more queue managers within a local node. Every managed WebSphere MQ node must have its own WMQ Agent unless agent-less monitoring is preferred. In the M6-WMQ hierarchy, events occurring at the WMQ Agent level are directed back up to the workgroup server, and then distributed among client applications.

The *M6-WMQ Publish Subscriber* server is an agent that distributes WebSphere MQ events among subscribers. It is installed once on WebSphere MQ node and once activated monitors events from all queue managers on this node. Also, M6-WMQ Publish Subscriber monitors dead letter queue and generates event if message arrived on the dead letter queue.

The *M6-WMQ Message Server* provides message management of WebSphere MQ queue managers. It is also installed once per MQ node, but could be activated per queue manager. It allows copy, move, delete, find, and edit of individual or group of messages.

The *M6-WMQ Connection Manager* allows you to connect to queue managers running on remote WMQ nodes without installing agents on the nodes. It uses Server-Conn channels to connect and issue administration commands and receive replies and events. While the Server-Conn channel is available, you will be able to perform most of the MQ administration functions except a few like starting/stopping queue managers, browsing logs etc.

2.2.7 Front-end Management Applications

M6-WMQ is manageable with any front-end management application that supports SNMP, including HP OpenView, Tivoli, and NetView. The M6-WMQ Explorer administers WMQ networks and performs various administrative functions of WMQ objects such as creating, deleting and modifying. The Message Explorer manages WMQ messages such as creating, deleting, modifying, copying, moving and finding. If the message queue is AMS protected, the message header can be browsed but the message payload will remain encrypted.

2.2.8 Plug-ins

Plug-in tools for the integration of Non-SNMP Enterprise.

2.2.8.1 ITO

Provides the adapter to integrate M6-WMQ with HPOV (ITO) Enterprise Management.

2.2.8.2 Tivoli Adapter

Provides the adapter to integrate AutoPilot M6 with the Tivoli Enterprise Management solution.

2.2.9 Components by Platform

The *Components by Platform* table lists all M6-WMQ components currently available and indicates the associated platforms.

	z/OS	OS/400	AIX	Linux	SunOS	HP-UX	Windows
CEP Server			*	*	*	*	*
Workgroup Server			*	*	*	*	*
WMQ Agent	*	*	*	*	*	*	*
Event Publish-Subscriber	*	*	*	*	*	*	*
Message Server	*	*	*	*	*	*	*
Connection Manager				*			*
Log Adapter (Tivoli, HP-OV)			*	*	*	*	*
M6-WMQ Explorer							*
Message Explorer							*
HP-OV Front End				*	*	*	

* Available

Windows 2000, 2003 and XP operating systems are supported.

HP-UX PA-RISC and Itanium are supported.

Chapter 3: Management Planning

Every organization has different needs and goals. To use M6-WMQ effectively a management plan is required. The management plan must fit local organizational goals and needs. This chapter provides detailed information for the development of Management Plans.

3.1 Essential Management Issues

M6-WMQ imposes a hierarchical management structure on all WebSphere MQ components. The building blocks of an effective management structure are:

- A view of all WebSphere MQ nodes
- Small, easily managed node groups
- Overlapping groups of nodes that share management responsibilities and provide higher fault tolerance
- A team of system administrators, each oversees a limited number of groups

3.2 Management Groups

In M6-WMQ, a management group is a set of nodes managed by a single workgroup server. Before creating and defining management groups, consider the following questions:

Question: How many WebSphere MQ nodes will there be in a single group?

Answer: A single workgroup server can manage up to 2048 WebSphere MQ nodes, with a maximum of 2048 queue managers on each node.

Question: What criteria should be used to create groups?

Answer: System managers should develop meaningful criteria for grouping nodes under specific workgroup servers. The numbers of objects per queue could limit the number of nodes assigned to a workgroup server.

Question: On what machines will the workgroup servers run?

Answer: Workgroup servers must run on Windows, Linux, or UNIX platforms.

3.3 Basic Management Structure

M6-WMQ's basic management structure, calls for a strict many-to-one relationship between nodes and groups. Theoretically, in effect:

- There should be multiple nodes assigned to a single workgroup server
- Each node should belong to one group and one group only

Figure 3-1 depicts a basic, many-to-one management structure. Note the two workgroup servers (Development and Accounting).

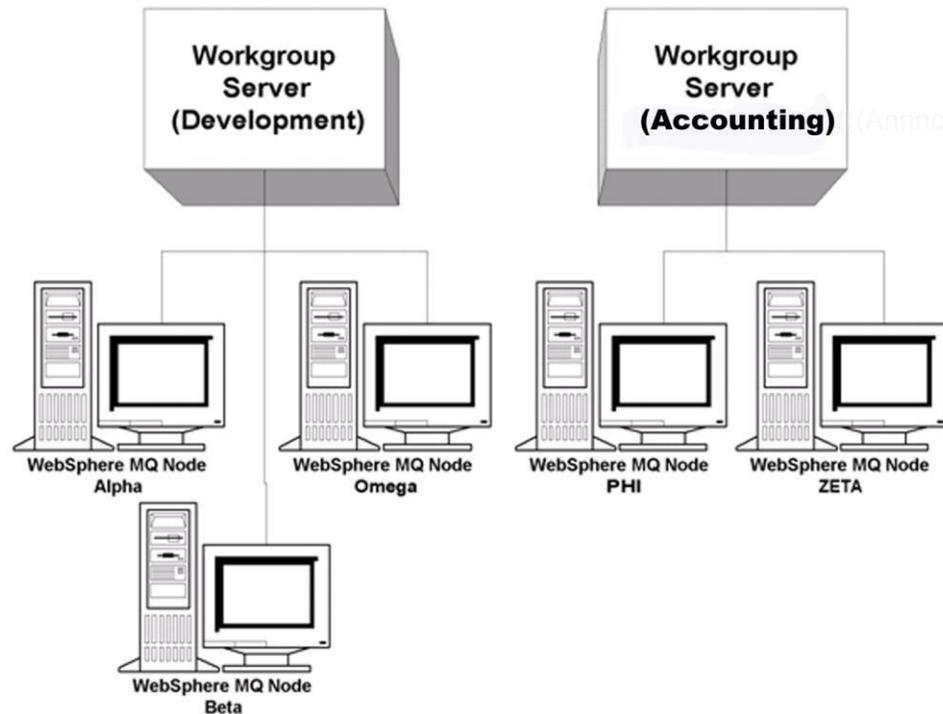


Figure 3-1. Managing WebSphere MQ Nodes Basic Management Structure

3.3.1 Problems with Basic Management Structure

A basic management structure is easy to configure and maintain, however, it has two critical limitations (illustrated in Figure 3-1):

- If the first workgroup server (Development) crashes due to system failure, the nodes Alpha, Omega and Beta are temporarily unmanaged. The condition of the WebSphere MQ objects on those nodes becomes "unknown," and WebSphere MQ failures may occur without the knowledge of system managers.
- Each WebSphere MQ node can only be viewed through a single workgroup server. If the number of management applications is large and network traffic is heavy, the Workgroup server can become overloaded, causing poor system performance.

3.4 Fault-tolerant Management Structure

Fault-tolerant management lets several workgroup servers monitor a single WebSphere MQ node, as shown in Figure 3-2. In Figure 3-2, the network human manager determined that two nodes are of critical importance. These two nodes should always be managed. The manager has created a backup group (top right) to manage these nodes if there is a problem with either of the other workgroup servers. Together, these groups are considered overlapping groups.

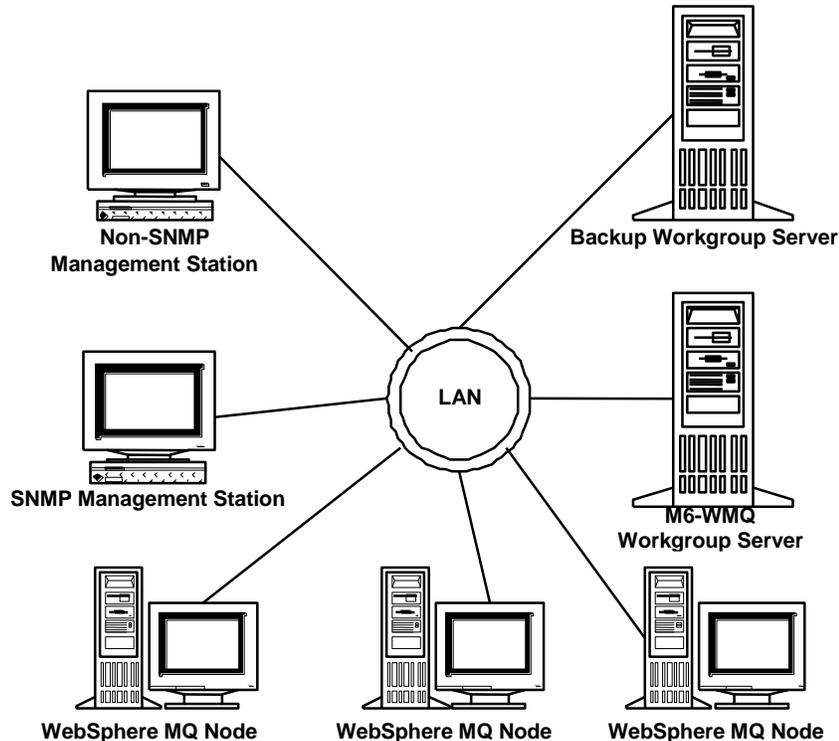


Figure 3-2. Fault-tolerant Management of WebSphere MQ

3.4.1 Rules for Overlapping Groups

Groups may overlap partially or completely, preferred. Create overlapping groups as follows:

- Select nodes that will participate under one or more workgroup servers
- Register these nodes under all the desired workgroup servers
- Define the nodes as either managed, or unmanaged by all selected workgroup servers, using the M6-WMQ Explorer

When a node is registered under multiple workgroup servers, one or more workgroup server can simultaneously manage it. If desired, a single workgroup server can manage the node. Define it, using the M6-WMQ Explorer, as managed under one workgroup server and as unmanaged under the other workgroup server(s).

Whenever a node is defined as unmanaged by a (back-up) workgroup server, the workgroup server will not actively manage that node. For example: When the primary Workgroup server fails, the back-up workgroup server should then be instructed by a human manager via M6-WMQ Explorer to actively manage the nodes of the failed workgroup server.

In order to achieve fault tolerance, define nodes as managed by at least two workgroup servers.

	NOTE:	If there is a node registered under two or more workgroup servers, Event Publisher must be running on the node.
---	--------------	---

3.5 Planning M6-WMQ Configuration

Generally, system managers have a great deal of flexibility when configuring and distributing M6-WMQ components. Nonetheless, M6-WMQ configurations must follow these basic guidelines:

- All installed components must run on WebSphere MQ nodes.
- There must be at least one workgroup server.
- There must be a WMQ Agent on each node.
- Each WMQ Agent must be overseen by at least one workgroup server.
- For each WebSphere MQ queue manager, there should be a running instance of the MQ Event Publisher.
- For mainframe installations, a PMS Command Server should run on each mainframe WebSphere MQ node.

3.5.1 Basic Management Configuration

A basic management configuration as seen in Figure 3-3 below, consists of WebSphere MQ nodes and one or more management stations. Nodes are divided into groups. Each node in a group must run WMQ Agent. In addition, one of the nodes must run an M6-WMQ workgroup server.

The configuration below features:

- Two management stations (one SNMP-based, the other non-SNMP-based)
- One group consisting of four nodes; three nodes are running WebSphere MQ WMQ Agents; the other is running a workgroup server.

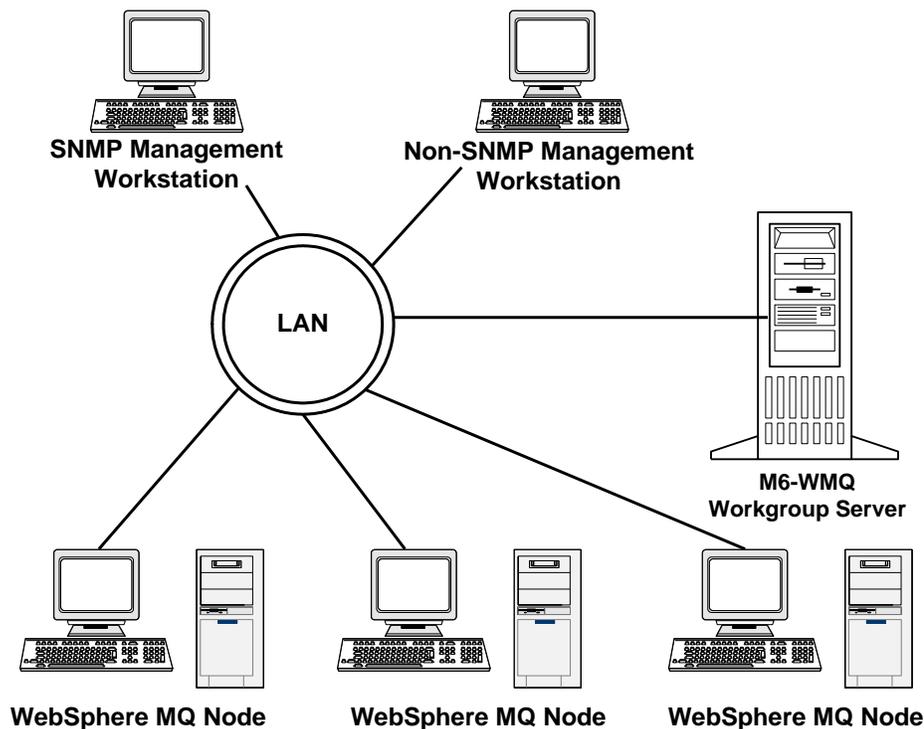


Figure 3-3. Using M6-WMQ for SNMP-based Management of WebSphere MQ

Chapter 4: Configuring M6-WMQ

This chapter provides instructions, direction and data for configuring M6-WMQ.

4.1 Basic Configuration Tasks

To configure M6-WMQ you need to:

- Define one or more M6-WMQ workgroups
- Configure each workgroup server

When the configuration is complete, each of your WebSphere MQ nodes must be registered with the appropriate workgroup server.

4.2 Defining an M6-WMQ Workgroup Server

Before configuring each workgroup server, define a management workgroup. A management workgroup includes a set of WebSphere MQ nodes that share common characteristics. The criteria for creating workgroups are defined by a human manager.

	NOTE:	A single workgroup server can manage up to 2048 WebSphere MQ nodes, with a maximum of 2048 queue managers on each node.
---	--------------	---

Table 4-1. Possible Criteria for Creating Management Workgroups

Criteria	Explanation
Application	Nodes are grouped on the basis of the WebSphere MQ-based application they are running.
Location	Nodes are grouped on the basis of their proximity on the network.
Platform	Nodes are grouped on the basis of their hardware and software platforms.

4.2.1 Defining a Workgroup Using the nsqsqlmk Utility (SQL Database)

The nsqsqlmk utility creates a new workgroup database and defines workgroup records used by the M6-WMQ workgroup server. This utility also allows additional workgroup records for use by additional M6-WMQ Workgroup Servers. You can use separate databases for each workgroup, or all workgroups can be maintained in the same database. Note that all the workgroup definition data is stored in the directory:

```
[APWMQ_HOME]\groups\[WorkgroupName]
```

1. Before running nsqsqlmk utility, your ODBC Data Source must be configured. Create two data sources: one for the Workgroup Server to use to connect to the database for the workgroup server information and the other for the permits information (the latter is optional and is only required maintaining the permits information in the database).
 - a. If you are using Windows, configure ODBC as follows:
 - 1) In database server, create a database for M6-WMQ to store workgroup server and Permits information, or create two databases, one for the workgroup server information and the other for the Permits information. If using separate databases, the user name for the workgroup server to connect with MUST be the same in each database.
 - 2) Open the ODBC Data Source Administrator window by selecting **Start > Settings > Control Panel > Administrative Tools > Data Sources (ODBC)**.

	NOTE:	On 64-bit Windows system, the 32-bit ODBC Data Source Administrator must be used. To start it, run the following command from a command prompt: <code>%SystemRoot%\SysWoW64\odbcad32.exe</code>
---	--------------	--

- 3) First configure the data source for the workgroup server information.

- 4) Select the appropriate data source type(s):
 - a) If running the Workgroup Server as a Windows Service, create System data sources (select *System DSN* tab).
 - b) If only running the Workgroup Server from a command prompt, create a User data source (select *User DSN* tab).
- 5) In *ODBC Data Source Administrator* window, click **Add**. The *Create New Data Source* wizard will be displayed.
- 6) In *Create New Data Source* wizard, select the appropriate driver and click **Finish**.
- 7) The configuration window for the selected driver will open. The contents of this window are driver-specific. In general, the defaults are fine. However, there are a few fields that must be configured properly:

Data Source Name – Choose any name desired for workgroup server information.

Enter **NASTEL_PERMITS** for Permits Data Source.

Server or Host – Specify appropriate database server host name or IP address

Port – Specify appropriate port number for database server

User – User name to connect to the database

Password – Password for database user

Database – Name of database

- 8) If there is a **Test** button for testing the connection, click it to verify the data source configuration.
 - 9) Click **OK** to save the data source configuration.
 - 10) Repeat steps 1.a.4 through 1.a.7 to create Permits Data Source. For this data source, the Data Source Name as specified in step 1.a.7 above **MUST** be **NASTEL_PERMITS**.
- b. If you are using UNIX, configure ODBC as follows:
- 1) In database server, create a database for M6-WMQ to store workgroup server and Permits information, or create two databases, one for the workgroup server information and the other for the Permits information.
 - 2) Open file `odbc.ini` located in directory `[APWMQ_HOME]/odbc/config`.
 - 3) Modify DRIVER PATH, SERVER, PORT, USER and Password for two data sources:
 - a) **NASTEL_APWMQ** – This data source is for the database containing the workgroup server information. You can change this name to anything you like.
 - b) **NASTEL_PERMITS** – This data source is for the database containing the Permits information. **THIS NAME CANNOT BE CHANGED!**
 - 3) Save file.
 - 4) Define the environment variable ODBCINI that gives UNIX SQL programs, such as workgroup server, the path to the data source names:


```
export ODBCINI=$APWMQ_HOME/odbc/config/odbc.ini
```

This should be defined in the login profile (or on the command line) for user ID mqm and any other user ID that will be used to run the workgroup server.

2. To configure an M6-WMQ database, enter the command:

```
nsqsqlmk -mWorkgroupName -sPort -DdataSourceName -UdbUserId
-PdbPasswd
```

The default port assignment is 4010. The dataSourceName, dbUserId, and dbPasswd are the same as those used in step 1.a.7. (See *SQL Database Creation Utility* in [Appendix E](#) for additional arguments.) nsqsqlmk will display the names of the SQL schema files that it will process, along with the default workgroup information. If dataSourceName, dbUserId, or dbPasswd are not specified, you will be prompted to enter them.

Select one of the following options as described below:

Option 1 – Select this option to create a new M6-WMQ database or to drop an existing M6-WMQ database, including all workgroup definitions in it, and to recreate it. This option will also add the workgroup record for the specified default workgroup.

Option 2 – Select this option to create the M6-WMQ Permits database.

Option 3 – Select this option to create additional workgroups. Enter the requested information (values in brackets are the defaults).

Option 4 – Select this option to save the database connection information (data source, userid, password) into the workgroup configuration file. (See [section 4.4.2](#), Workgroup Configuration File (mqgroup.ini).) This option will save the password in an encrypted form. Saving this information allows the Workgroup Server to load the information from here instead of having to include it on the command line.

- When all the workgroups are defined, run the workgroup server by entering the command for each workgroup:

```
nsqmgr [-console] -mWorkgroupName -u -a -DDataSourceName -UdbUserId
-PdbPassword
```

	NOTE:	You can only run one instance of the workgroup server per M6-WMQ workgroup. If you start a duplicate instance of the workgroup server, it will be terminated automatically.
---	--------------	---

Adding Additional Windows Service for a Workgroup Server

To add an additional windows service for a workgroup server, enter the command:

```
nsqmgr -install -mWorkgroupName
```

The workgroup-specific environment variable must be defined only if each workgroup server is using different settings (with the exception of `-m`, which is ignored when running as a service).

The workgroup server reads its startup parameters from the environment variable `APWMQ_GS_OPTIONS`. All instances of the workgroup server will process this variable. The settings defined here can be overridden by setting command line arguments.

This workgroup server-specific environment variable is only processed by the workgroup server when running as a Windows Service. It allows options to be set for a specific workgroup server instance (each instance is configured for a different workgroup). The format of the variable is "`APWMQ_GS_OPTIONS_<wkgrp>`", where "`<wkgrp>`" is the name of the workgroup.

As an example, suppose there are two workgroup server service instances installed, `PROD` and `DEV`. When each starts, each will first process the arguments defined by `APWMQ_GS_OPTIONS`. Then, each will look to see if there are options specific to it. Each service instance will then look for environment variables `APWMQ_GS_OPTIONS_PROD` and `APWMQ_GS_OPTIONS_DEV`, respectively. If these variables are defined, then the arguments defined in them will be processed, overriding any defined by `APWMQ_GS_OPTIONS`.

To continue with this example, let's assume that both instances are connecting to the same SQL database, but each is requiring a different type of user authentication and authorization. So, the environment variables would be defined as follows:

```
APWMQ_GS_OPTIONS=-Dnastel_apwmq -Udbuser -Pdbpwd
APWMQ_GS_OPTIONS_PROD=+a2 +u
APWMQ_GS_OPTIONS_DEV=-a -u
```

4.3 Configuring a Workgroup Server

The following is required to configure a workgroup server:

- Obtain and install a license file
- Check the limit parameters for each workgroup server
- Modify any limit parameters you want to change, using the M6-WMQ Explorer
- Configure the workgroup server's discovery policy
- Configure the workgroup server's event policy.
- Configure the workgroup server's accounting and statistics log sizes

4.3.1 Obtaining M6-WMQ License

	IMPORTANT!	A new license is required for version 6.5.
---	-------------------	--

Before you start M6-WMQ, you need to obtain a CPU-based license file and install it on each system on which a workgroup server will run.

A workgroup server license is a digitally-signed text file that contains identification and license entitlement information for your M6-WMQ installation. Each time the workgroup server is started, it checks the license file for your licensing and configuration information.

To obtain a license file, do the following:

	NOTE:	The product name is M6-WMQ.
--	--------------	-----------------------------

1. Navigate to **Start > Programs > AutoPilot M6 for WMQ > Request License**.
2. Follow the instructions on the web page. You will receive an M6-WMQ license file that must be installed. (See procedure below.)

4.3.2 Installing M6-WMQ License

	NOTE:	Before installing your license, stop the workgroup server that the license will be installed on.
---	--------------	--

When you receive your license file, back it up in a safe place in case you ever need to reinstall M6-WMQ. For the system that will run the workgroup server identified in the license file, copy the license file to the directory:

```
[AUTOPILOT\WMQ]\config\groups
```

where:

```
[AUTOPILOT\WMQ] is the directory path where M6-WMQ is installed.
```

	NOTE:	The name of the license file must be AutoPilotWMQ_xyz.lic, where xyz is the name of the workgroup.
---	--------------	--

Check the license file information by starting your workgroup server and running it from the command line:

Type the command:

```
nsqmgr -console (Windows only) -m<GroupName>
```

The workgroup server will return the licensing information, similar to the following example:

```
LICENSING INFORMATION:
SOURCE(C:\Program
Files\nastel\apwmq\config\groups\AutoPilotWMQ_MQM.lic) WORKGROUP(MQM)

FEATURES(SQLDB)
WORKGROUP_HOSTS(host)
CPU_COUNT(14)
USER_COUNT(10)

AGENTLESS_CONNECTION_COUNT(250)
EXPIRATION_DATE(Wed May 21 00.00.00 2008)
```

where:

FEATURES: List of licensable features that the workgroup server is authorized to use.

WORKGROUP_HOSTS: List of network hosts that the workgroup server is licensed to run on.

CPU_COUNT: Maximum number of system processors that can be simultaneously managed by a single workgroup server.

USER_COUNT: Maximum number of client users that can simultaneously connect to a single workgroup server via the M6-WMQ Explorer, Message Server and/or AutoPilot M6.

AGENTLESS_CONNECTION_COUNT: Maximum number of concurrent Queue Manager connections allowed by Connection Manager.

EXPIRATION_DATE: Date on which the license expires and a new license will have to be obtained.

4.3.3 Checking Workgroup Server Limit Parameters

Each workgroup server takes a number of limit parameters. These parameters determine the maximum amount of activity the workgroup server can handle at one time. The limit parameters are displayed when you invoke the workgroup server from the command line.

4.3.3.1 Check Limit Parameters

From a command prompt, issue the command:

```
nsqmgr -console (Windows only) -mWorkgroupName
```

The workgroup server returns your licensing information as in the previous section. Then it displays the current values of the limit parameters, as in this example:

```
MAX_WORKGROUP_CLIENTS=2048
MAX_MANAGED_NODES=2048
MAX_SUPPORTED_CONNECTIONS=32767
MAX_Q_MGRS_PER_NODE=2048
```

Note that:

- **MAX_WORKGROUP_CLIENTS:** The maximum number of client services in M6-WMQ Explorer and Message Explorer that can be connected to the workgroup server at one time. The maximum value is **2048**.
- **MAX_QMGR_THREADS:** The maximum number of WebSphere MQ queue managers that can be managed. Its value is calculated by multiplying the value of MAX_MANAGED_NODES by the value of MAX_Q_MGRS_PER_NODE.
- **MAX_MANAGED_NODES:** The maximum number of WebSphere MQ nodes that can be managed. The maximum value is **2048**.
- **MAX_SUPPORTED_CONNECTIONS:** The predefined limit on the number of connections the workgroup server can handle. Its value must satisfy the following constraint:
$$\text{max_nodes} * (1 + \text{max_qmgrs_per_node}) + (\text{max_clients} + 1) < \text{max_supported_connections}$$
where max_supported_connections maximum value is set to **32767**.
- **MAX_Q_MGRS_PER_NODE:** The maximum number of queue managers that can be managed on each node. The maximum value is **100**.

**NOTE:**

For Linux on zSeries s390x platform, the maximum number for all TCP connections is 1024 and is driven by the operating system.

4.3.4 Modifying Limit Parameters using M6-WMQ Explorer

To modify the limit parameters for the workgroup server, work from M6-WMQ Explorer, a front-end application available on Windows platforms. For information about using the Explorer, refer to the application's on-line help. Note that to access the workgroup server through M6-WMQ Explorer, the workgroup server will have to be started first.

To modify limit parameters using M6-WMQ Explorer:

1. Start workgroup server from command line by issuing command:


```
nsqmgr -console -m[WorkgroupName]
```

 where *WorkgroupName* is name of your workgroup server (by default, MQM)
2. On a Windows platform, launch M6-WMQ explorer and navigate to workgroup server. Right-click on workgroup server and select *Properties*. The *M6-WMQ Workgroup Server Properties* dialog box is displayed.
3. Find *Workgroup Limits* section in the lower right corner. Using up/down arrows, modify settings for *Max Nodes*, *Max.Clients* and *QMgs/Node*. When completed, click **OK**.
For example, if there are 30 WebSphere MQ nodes (or plan to have), set *Max Nodes* to 30.

	NOTES:	<ol style="list-style-type: none"> 1. These limits cannot exceed those defined in the workgroup server license file. 2. Settings will not take effect until the workgroup server is restarted.
---	---------------	--

4. Using up/down arrows, modify setting for *Event Log*, *Acct Log* and *Stat Log*.

Event Log setting is the maximum number of events the workgroup server can store in event history logs, for each event category. The value entered must not be a negative number. When limit is exceeded, the oldest message is automatically purged from the system.

Acct Log setting is the maximum number of accounting messages allowed. When limit is exceeded, the oldest message is automatically purged from the system.

Stat Log setting is the maximum number of statistical messages allowed. When limit is exceeded, the oldest message is automatically purged from the system.

For these logs, a value of zero allows the log to grow unbounded when using an SQL database, or resets the log limit to the default value when using the Nastel database. When completed, click **OK**.

4.3.5 Configuring Workgroup Server Discovery Policy

The workgroup server can auto-discover WebSphere MQ nodes as they are added to your network. To use this feature, configure the workgroup server discover policy. As in the preceding section, use the M6-WMQ Explorer to set the policy.

To configure the workgroup server discovery policy:

1. In M6-WMQ Explorer, right-click on the workgroup server and select *Properties*.
2. The *M6-WMQ Workgroup Server Properties* dialog box is displayed.
3. Under *Discovery Policy*, enter the following settings:
 - **M6-WMQ Agent Port:** Number of TCP/IP ports your WMQ Agents listen on. Default value is 5010.
 - **IP Network Address:** IP address used by the workgroup server when sending a UDP broadcast packet in order to discover new WebSphere MQ nodes. The default value is 0.0.0.0. Change to another value if the network requires a unique, non-zero broadcast address (supplied by your local system administrator).
 - **Discovery:** Time interval (in minutes) between each auto-discovery attempt. The default value, 0 (zero), disables the auto-discovery process. An appropriate interval setting is 60 minutes; 1440 minutes is recommended.
4. Click **OK**.

4.3.6 Configuring Workgroup Server Event Policy

The Event Policy lets the users determine what types of events the user wants to receive from the workgroup server and display on the M6-WMQ Explorer screen. It is set using the M6-WMQ Explorer.

To configure the event policy:

1. In M6-WMQ Explorer, right-click on the workgroup server and select *Properties*. The *M6-WMQ Workgroup Server Properties* dialog box is displayed.
2. Under *Event Policy*, check each event type (*workgroup*, *queue manager*, *performance*, *alter*, *channel* and/or *command*) to be displayed or check **Event Policy** to automatically select all event types.
3. Click **OK**.

4.3.7 Configuring Workgroup Server Accounting and Statistics Log Sizes

The maximum accounting and statistics logs sizes should be configured based on the number of WMQ queues and channels and the number of days of historical information to maintain. There is one value to control the number of records in each accounting table, and another to control the number of records in each statistics table.

In general, the Accounting and Statistics messages are generated at fixed intervals (there are separate Queue Manager attributes for each interval). The interval granularity is seconds. The following equation provides an estimate of the number of intervals in a single day:

$$\text{NumIntervals} = (60 * 60 * 24) / \text{Interval}$$

The default interval size for each is 1800 seconds (30 minutes). Using this default value, there are about $(60 * 60 * 24) / 1800 = 48$ intervals (48 accounting intervals and 48 statistics intervals per day).

Estimating Accounting Log Size

In general, in each accounting interval there is one MQI Accounting record per queue manager (with accounting messages enabled) per application and one Queue Accounting record per queue (with accounting messages enabled) per application. The following equation gives an estimate of the number of records in a single accounting interval:

$$\text{NumAcctRecs} = (\text{NumQmgrs} + \text{NumQueues}) * \text{NumAppls}$$

where *NumAppl* is the number of WMQ Applications. This equation assumes that each queue manager has the same accounting interval. Combining this with the estimate of the number of intervals in a day, the following provides an estimate of the number of accounting records in a single day:

$$\text{NumAcctRecsPerDay} = \text{NumAcctRecs} * (60 * 60 * 24) / \text{AcctInterval}$$

To determine a reasonable Accounting Log size for the workgroup, use the following equation:

$$\text{AcctLogSize} = \text{NumAcctRecsPerDay} * \text{NumDaysOfHistoricalData}$$

Estimating Statistics Log Size

In general, in each statistics interval there is one MQI Statistics record per queue manager (with statistics messages enabled), one Queue Statistics record per queue (with statistics messages enabled), and one Channel Statistics record per active channel (with statistics messages enabled). The following equation gives an estimate of the number of records in a single statistics interval:

$$\text{NumStatRecs} = \text{NumQmgrs} + \text{NumQueues} + \text{NumChannels}$$

This equation assumes that each queue manager has the same statistics interval. Combining this with the estimate of the number of intervals in a day, the following provides an estimate of the number of statistics records generated in a single day:

$$\text{NumStatRecsPerDay} = \text{NumStatRecs} * (60 * 60 * 24) / \text{StatInterval}$$

To determine a reasonable Statistics Log size for the workgroup, use the following equation:

$$\text{StatLogSize} = \text{NumStatRecsPerDay} * \text{NumDaysOfHistoricalData}$$

4.4 M6-WMQ Configuration Files

M6-WMQ uses the configuration files listed in the Configuration Files table below:

	NOTE:	These configuration files are only needed for WebSphere MQ nodes that function below the Workgroup server level. A dedicated Workgroup server node does not need them.
---	--------------	--

Table 4-2. M6-WMQ Configuration Files

File	Description
Workgroup configuration file (mqgroup.ini)	Contains a list of workgroups defined on the network.
Return code configuration file (nsqrc.ini)	Contains WebSphere MQ reason codes, M6-WMQ reason codes and their explanations.
Security file (permits.ini)	Contains statements to allow workgroup server and WMQ Agent to check user authorization commands and WebSphere MQ objects.

4.4.1 Location of M6-WMQ Configuration Files

All configuration files should be stored in the directory:

```
[AUTOPILOT_WM] \config\groups
```

4.4.2 Workgroup Configuration File (mqgroup.ini)

The workgroup configuration file is named `mqgroup.ini`. It contains entries that identify each workgroup and workgroup server associated with it. The format of the workgroup configuration file is:

```
Group::AliasName=WorkgroupName
Group::Name::Node=NodeName WorkgroupName::Service=port
number | service name
```

	NOTE:	For more information on M6-WMQ security, see “Defining a Workgroup by Running the <code>nsqsqlmk</code> Utility” in this chapter.
---	--------------	---

As in the example:

```
Group::MQEVENT=MQTEST
Group::MQTEST=MQTEST

MQTEST::Node=alpha MQTEST::Service=4005
```

In the above example, the workgroup configuration file specifies a workgroup named MQTEST. The workgroup server for MQTEST runs on node alpha, listens on TCP/IP port 4005 and uses secured (“ta”) client connections.

	NOTE:	In the workgroup configuration file, entries are case-sensitive. There are no spaces allowed between the delimiters “::” and “=”
---	--------------	--

4.4.3 Verify Syntax of Configuration Files

The syntax of a configuration files should be checked after each edit. The **nsqinichk** (nsqinichkd for debug version) utility is used to check the syntax in the following files:

- mqgroup.ini
- permits.ini
- nsqrc.ini
- nsqsev.ini
- mqqual.ini

This utility is run from the command line. It accepts only one parameter, the .ini file name. The possible errors are:

- Invalid or unexpected symbol
- Missing key token
- Missing value token.

For example:

```
nsqinichk mqgroup.ini
```

If the syntax is correct, the following message is displayed:

```
>nsqinichk D:\nastel\apwmq\config\groups\mqgroup.ini
Checking file
D:\nastel\apwmq\config\groups\mqgroup.ini...
No errors
```

If an error is detected, the line number and position in the line is displayed. For example:

```
Checking mqgroup.ini...
Unexpected symbol(s) '\' at line 15, column 32
```

Automatic Configuration File Checking

When the AP-WMQ agent (nsqmq) starts, it will automatically verify the syntax of mqgroup.ini and, if used, permits.ini.

4.5 Configuring WebSphere MQ Nodes

Each WebSphere MQ node must:

- Run a WMQ Agent (unless agentless)
- Be supervised by at least one workgroup server

4.5.1 Setting M6-WMQ Environment Variable

Each node running M6-WMQ must define the M6-WMQ environment variable. The variable should be set to the M6-WMQ installation directory, as in the following examples:

```
[APWMQ_HOME]=c:\nastel\apwmq (Windows)
```

```
[APWMQ_HOME]=/opt/nastel/AutoPilot/WMQ (UNIX)
```

4.5.2 Including a WMQ Agent in a Workgroup

Once the workgroup servers are up and running, start adding nodes to each workgroup by configuring the file `mqgroup.ini`.

	NOTE:	If a node's WMQ Agent is needed to belong to more than one workgroup, the WMQ Agent must be included separately under each workgroup.
---	--------------	---

To include a WMQ Agent in a workgroup:

1. Open the workgroup configuration file `mqgroup.ini`, located in the directory `[APWMQ_HOME]\config\groups` on the WMQ Agent's node.
2. Specify the name of the workgroup(s) to which the WMQ Agent will belong.
3. Specify the name of the node on which the WMQ Agent resides. For more information about `mqgroup.ini`, see [section 4.4.2](#).
4. On UNIX command line, define an environmental variable named `APWMQ_MQM` and set it to the WebSphere MQ installation directory. On UNIX, the installation directory is usually `/var/mqm`. If it is unclear where the WebSphere MQ installation directory is, refer to the IBM WebSphere MQ documentation.

4.5.3 Registering WMQ Agents Under Workgroup Servers

Each independent WebSphere MQ node must be registered under one or more workgroup servers.

	NOTES:	Ensure the WMQ Agent has permission to access WebSphere MQ resources.
---	---------------	---

To register a WMQ Agent under a Workgroup server:

1. Ensure that the workgroup server is started. If the workgroup server is not active, the WMQ Agent will not be registered successfully.
2. Start the WMQ Agent from the command line:

```
nsqmq -console (Windows only) -mWorkgroupName -sPort
```

Do not omit the workgroup name: If the workgroup name is omitted, the WMQ Agent will not register, and will be excluded from the workgroup. The argument *port* should be the number of an unused TCP/IP server port that the WMQ Agent can bind to. (See the workgroup server command-line Argument table in [Appendix E](#) for a full list of command-line arguments for the WMQ Agent.) The default value is 5000. It should register with the workgroup.

4.5.4 Setting up WMQ Agents Configuration Parameters

To set up the WMQ Agent configuration parameters, unmanage the WMQ Agent and then double-click it.

Table 4-3. WMQ Agent Parameters

Policy	Parameter	Description
Communication	Heart Beat	Heart beat interval, in minutes, of the WMQ Agent. Default: 1 minute.
	Update Interval	Amount of time, in seconds, before the workgroup server updates information about the node's WebSphere MQ objects, such as queue managers, queues and channels. Default: 30 seconds.
	Request Timeout	Command time-out period, in seconds. Sets amount of time workgroup server waits for a response from the WMQ Agent running on the node. Default: 60 seconds.
	Command Limit	Specifies the number of commands that can be issued by the workgroup server to the WMQ Agent without acknowledgement from the agent. A value of zero indicates that there is no command limit. Default: 5000 commands.
	Event Counter	Number of workgroup events that the workgroup server generated since the last reset of the Event Counter parameter. The Event Counter parameter can be reset by editing the workgroup server's properties.
	Agent Version	The version of the WMQ Agent. This field cannot be changed.
	Use DNS	Select this checkbox if you want to use Domain Naming Service instead of WebSphere MQ node's IP address.
	CPU Count	Number of licensed CPUs.
Discovery	Queue Managers	Specifies which queue managers are to be automatically discovered by the workgroup server. This field accepts simple wildcards, meaning that characters can be followed by an asterisk (*).
	Cluster Queue Managers	Specifies which cluster queue managers are to be automatically discovered by the workgroup server. This field accepts simple wildcards, meaning that characters can be followed by an asterisk (*).
	Local Queues	Specifies which local queues are to be automatically discovered by the workgroup server. This field accepts simple wildcards, meaning that characters can be followed by an asterisk (*).
	Model Queues	Specifies which model queues are to be automatically discovered by the workgroup server. This field accepts simple wildcards, meaning that characters can be followed by an asterisk (*).
	Alias Queues	Specifies which alias queues are to be automatically discovered by the workgroup server. This field accepts simple wildcards, meaning that characters can be followed by an asterisk (*).
	Remote Queues	Specifies which remote queues are to be automatically discovered by the workgroup server. This field accepts simple wildcards, meaning that characters can be followed by an asterisk (*).
	Processes	Specifies which processes are to be automatically discovered by the workgroup server. This field accepts simple wildcards, meaning that characters can be followed by an asterisk (*).

Table 4-3. WMQ Agent Parameters

Policy	Parameter	Description
Discovery (Cont)	Channels	Specifies which channels are to be automatically discovered by the workgroup server. This field accepts simple wildcards, meaning that characters can be followed by an asterisk (*).
	Listeners	Specifies which listeners are to be automatically discovered by the workgroup server. This field accepts simple wildcards, meaning that characters can be followed by an asterisk (*).
	Services	Specifies which services are to be automatically discovered by the workgroup server. This field accepts simple wildcards, meaning that characters can be followed by an asterisk (*).
	Namelist	Specifies which namelists are to be automatically discovered by the workgroup server. This field accepts simple wildcards, meaning that characters can be followed by an asterisk (*).
	Authentication Informations	Specifies which authentication informations are to be automatically discovered by the workgroup server. This field accepts simple wildcards, meaning that characters can be followed by an asterisk (*).
	Topics	Specifies which topics are to be automatically discovered by the workgroup server. This field accepts simple wildcards, meaning that characters can be followed by an asterisk (*).
	Subscriptions	Specifies which subscriptions are to be automatically discovered by the workgroup server. This field accepts simple wildcards, meaning that characters can be followed by an asterisk (*).
	Discovery Period	Time interval, in minutes, at which the WMQ Agent discovers MQ objects and reports any changes to the workgroup server. Default: 720 minutes.

4.5.5 Configuring Remote Queue Manager Connections Using the Connection Manager

The M6-WMQ Connection Manager allows you to connect to queue managers running on remote WMQ servers without installing agents on the server. It uses Server-Conn channels to connect to queue managers and issue administration commands and receive replies and events. While the Server-Conn channel is available, you will be able to perform most of the MQ administrative functions. If you have multiple workgroup servers, you will need one Connection Manager per workgroup.

Installation Requirements

The Connection Manager is an MQ client application and needs the MQ client library installed in order to run on a non-MQ server machine.

The MCAUSER attribute of the SVRCONN channel

When using the Connection Manager, the connection to a remote queue manager is done via a SVRCONN (server-connection) channel on the remote server. By default, when the SYSTEM.DEF.SVRCONN channel is created, the MCAUSER attribute is set to blanks.

The user ID for connecting to a queue manager

On i5/OS, UNIX systems, and Windows systems, the user ID whose authority is checked when an MCA connects to a queue manager is the one under which the MCA is running. This is known as the *default user ID* of the MCA. The responder (svrconn) MCA will be started by the WebSphere MQ listener and will run under the same user ID as that of the listener, such as mqm on UNIX systems. The default user ID of the MCA must be a member of the Windows/UNIX mqm group or the Windows Administrators group or the QMQMADM group on i5/OS. This requirement is typically satisfied at WMQ install time.

The user ID for subsequent authority checks

- The user ID whose authority is checked when the MCA accesses queue manager resources subsequent to connecting to a queue manager might be different from the one that was checked when the MCA connected to the queue manager. User IDs that might be used are the value of the MCAUSER parameter in the channel definition.
- For a SVRCONN channel, the user ID that is received from a client system when a WebSphere MQ client application issues an MQCONN call.

MCAUSER Values

If the MCAUSER field is blank at connection time, the SRVRCONN MCA uses the user ID received from a client (such as Connection Manager) when attempting to access queue manager resources (e.g., objects) after connecting to the queue manager. If the client user ID is not a valid WMQ user on the target queue manager, then the resource access will fail with a 'not authorized' error.



NOTE:

If the user ID received from the client is blank, the user ID that started the server-connection channel is used. This case will not occur with the CM since the user ID is always non-blank.

Use of Model Queue

The Connection Manager uses a model queue for creation of temporary dynamic queues, used for command processing and message retrieval. The default is SYSTEM.DEFAULT.MODEL.QUEUE. Rather than alter the system queue if some parameter customization is needed, you can create a custom model queue to be used instead, say NSQCM.MODEL.QUEUE, with, for example, a very large MAXMSGL parameter value for retrieving very large messages. Enter the new queue name in the NASTEL.MODEL.QUEUE line in file [APWMQ_HOME]\config\groups\mqqual.ini. When the APWMQ agent starts, it will check the qualifiers file and use that model queue instead of the default.

Configuration Steps

1. Start your remote queue manager TCP listeners with unusual port numbers that unauthorized system users would not know and would be hard to guess.

2. Use those port numbers when configuring the CM RemoteQmgrs.
3. For improved security, Connection Manager should use a svrconn ChannelName other than the default SYSTEM.DEF.SVRCONN. The MQ admin should set the MCAUSER attribute in channel SYSTEM.DEF.SVRCONN to an invalid user ID, such as JUNK or DUMMY. This will block all usage of the default svrconn channel.
4. For initial Connection Manager testing, avoid user ID and security problems:
On the remote queue managers, set the Connection Manager-selected SRVCONN channel attribute MCAUSER to a user ID that has full access rights to the queue manager resources. Thus, use the user ID used to run the queue manager, such as mqm on UNIX; MUSR_MQADMIN or SYSTEM on Windows; QMQM on i5/iOS; or MQM or SYSTEM on OpenVMS. This user ID has full access rights to the queue manager resources.
5. After initial Connection Manager testing, use a higher level of security:
On the remote queue managers, set the selected svrconn channel attribute MCAUSER to blank and define the Connection Manager user ID on the target queue manager as a valid MQ user using the AP-WMQ Explorer (right-click *Qmgr* > **Commands** > **Security** > **Authority Records**) or the `setmqaut` command.

To Configure a Remote Queue Manager Connection for Windows:

1. From the M6-WMQ Explorer, right-click the workgroup manager icon and select **Remote Queue Managers** from the short-cut menu.
The *Remote Queue Manager Connections* dialog box is displayed.

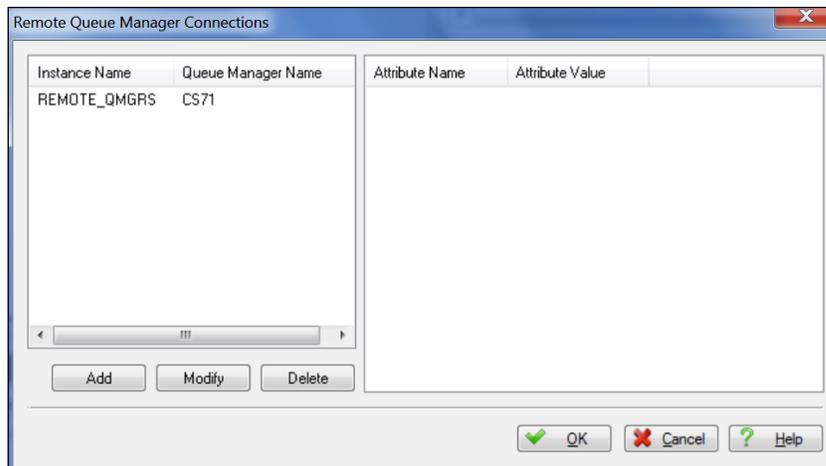


Figure 4-1. Remote Queue Manager Connections Dialog Box

2. Click **Add**. The *Add Queue Manager Connection* dialog box is displayed.

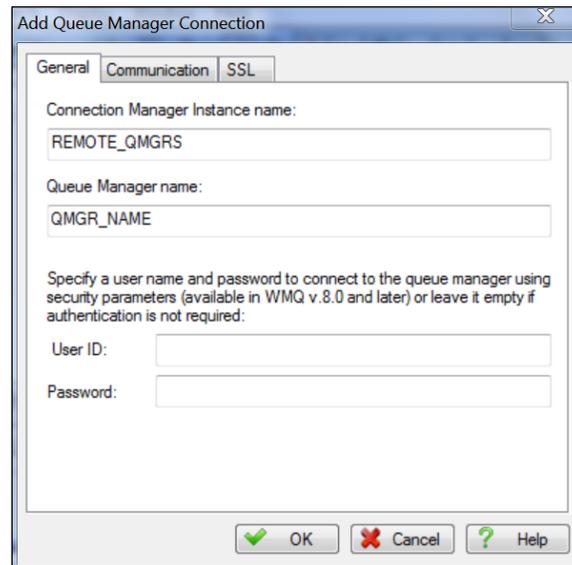


Figure 4-2. Add Queue Manager Connection Dialog Box – General

3. Enter the *Connection Manager instance* name. The default is REMOTE_QMGRS.
4. Enter the name of the new queue manager in the *Queue Manager name* field.
5. Enter a *User ID* and *Password* to connect to the queue manager using security parameters available in WebSphere MQ v 8.0 and later. Otherwise, leave blank if authentication is not required.
6. Click the **Communication** tab.
7. Enter the connection name of the new queue manager in the *Connection name* field.

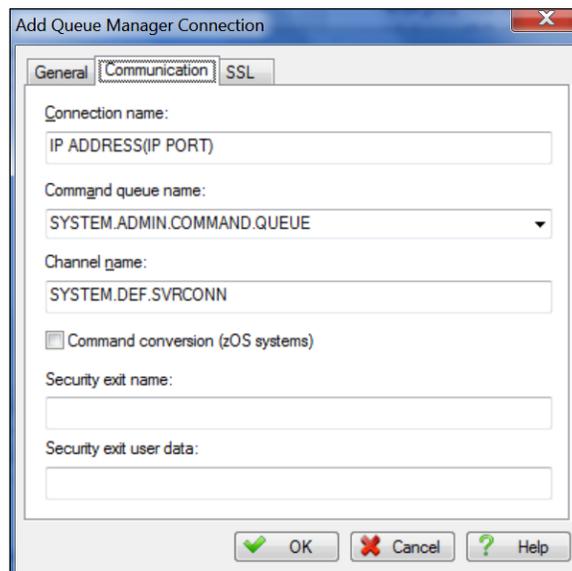


Figure 4-3. Add Queue Manager Connection Dialog Box – Communication

8. Enter the Command queue name in the *Command queue name* field. For most queue managers, use SYSTEM.ADMIN.COMMAND.QUEUE. For zOS, use SYSTEM.COMMAND.INPUT. Specify a local queue, not an alias queue.
9. Enter name of the server-connection (svrconn) channel to be used for connecting to the remote queue manager in the *Channel name* field. The default is SYSTEM.DEF.SVRCONN.
10. Check the *Command conversion (zOS systems)* box if this is a zOS queue manager connection that doesn't support PCF commands (V5 or lower).

11. If required, enter the *Security exit name* – the name of the channel exit to be used when connecting to the queue manager and the *Security exit user data* – up to 32 bytes of data passed to your SecurityExit.
12. Click the **SSL** (secure socket layer) tab and fill in required fields for your connection.

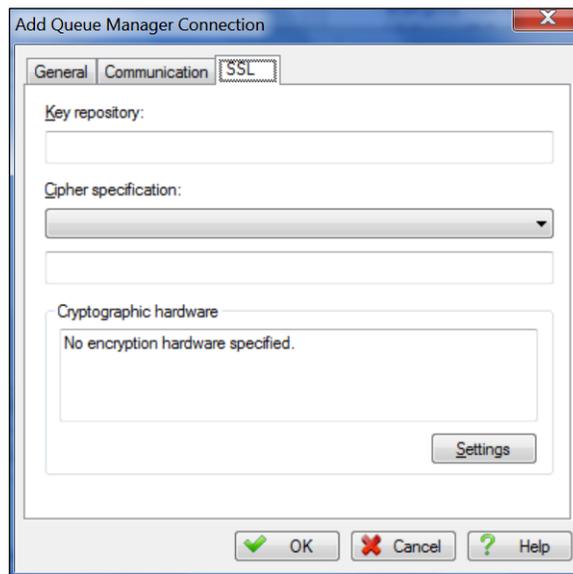


Figure 4-4. Add Queue Manager Connection Dialog Box – SSL

13. Click **OK**. The added information will be displayed on the *Remote Queue Manager Connections* dialog box.
A new node named *CM_instance_name*, such as REMOTE_QMGRS if the default instance name is used, is added to the M6-WMQ Explorer.
14. Configure the MCAUSER attribute of the SRVRCONN channels on the remote queue managers.
15. Repeat steps 1-14 to create a Connection Manager instance for each of your other workgroup servers, using a different Connection Manager instance name in step 3.

To Start Connection Manager Instance:

You can start Connection Manager instance from Windows services or the command line.

1. To start from Windows Services, first make sure the workgroup server is already running. Then navigate to the *Services* screen and start *M6-WMQ Connection Manager (workgroup_name)* service, where *workgroup_name* is the name of the workgroup server, such as MQM, that you chose during M6-WMQ for Windows installation.
2. To start from the command line:
 - a. Define an environment variable used by CM:
Set `APWMQ_CM_OPTIONS=-mworkgroup_name; -nREMOTE_QMGRS;`
`-sPort [: other_CM_options]`
where REMOTE_QMGRS is the name of the Connection Manager instance that works with the given workgroup_name.
 - b. To start the CM for a workgroup, enter the command
`nsqcm -console`
To see other CM options that can be incorporated into the environment variable:
`nsqcm -console -?`
The options are the same as for the APWMQ Agent.

Note: When connecting to a zOS queue manager, the `-N` option should be specified to prevent PCF buffer overflow.

3. If you have multiple workgroups, repeat steps 1-2 in a new command window for each workgroup.

If you don't have a CM service defined for a particular workgroup, create it from the Windows command line:

```
nsqcm -install workgroup_name
```

Starting the Connection Manager (CM) on Linux:

1. Edit script [APWMQ_HOME]/sbin/nsqcm_start.sh
 - a. Change the APWMQ_HOME environment variable setting /opt/nastel/apwmq if your installation directory is different.
 - b. Edit startup options -n<alias_node> and -s<CM_listener_port>, default 5011.
 - c. Add any other startup options as shown in the header of the script or in [section E.10](#), Connection Manager.
2. Run the script:
[APWMQ_HOME]/sbin/nsqcm_start.sh

Stopping the Connection Manager (CM) on Linux:

1. Edit script [APWMQ_HOME]/sbin/nsqcm_stop.sh
 - a. Change the APWMQ_HOME environment variable setting /opt/nastel/apwmq if your installation directory is different.
 - b. Edit options -n<alias_node> and -s<CM_listener_port>, default 5011.
 - c. Use the same values used in the start script.
2. Run the script:
[APWMQ_HOME]/sbin/nsqcm_stop.sh

This Page Intentionally Left Blank

Chapter 5: Deleted

This Page Intentionally Left Blank

Chapter 6: Managing the WMQ Network

Chapter 6 outlines the methods of using information gathered by M6-WMQ to manage the WebSphere MQ network.

6.1 Configuration Management

Configuration management is the ability to define, modify or delete:

- WebSphere MQ objects
- M6-WMQ objects

6.1.1 Defining, Modifying or Deleting Objects

The following objects can be defined, modified or deleted in M6-WMQ:

- **Workgroup Servers:** Because a Workgroup server can only oversee one active group at a time, you may want to create multiple Workgroup servers.
- **WebSphere MQ Nodes:** New WebSphere MQ nodes can be defined or deleted without interrupting individual Workgroup servers. Note that nodes are logical entries in a management group, which should be mapped to physical machines on the network. Each node should run WebSphere MQ and a WMQ agent.
- **Queue Managers:** Queue managers can be defined or deleted without interrupting individual Workgroup servers.
- **Queues, Channels, Cluster Queue Managers, Processes, Namelists, Listeners, Services, Authentication Information, Topics, and Subscriptions:** Queues (all types), channels, cluster queue managers, processes, namelists, listeners, services, authentication informations, topics, and subscriptions can be defined on any WebSphere MQ node registered under a Workgroup server (assuming the WebSphere MQ version installed on that node supports them).

6.1.2 Methods for Defining Objects

WebSphere MQ and M6-WMQ objects can be defined using the following methods:

- **Programmable Command Format:** PCF can be used to define groups, nodes and queue managers.
- **MQSC:** MQSC can be used to define queues, channels, processes, listeners, authentication informations, client connections and namelists.
- M6-WMQ Explorer and `nsqmqsc` utility.

6.1.3 Defining Objects with MQSC

There are two ways of defining objects with MQSC:

- **Interactive MQSC:** Users actively enter MQSC commands using the `nsqmqsc` utility. For more information on interactive MQSC, see the next section below.
- **MQSC Scripts:** Users write MQSC scripts that can be submitted in batches using the `nsqmqsc` utility.

6.1.4 Using Interactive MQSC

Interactive MQSC lets users issue real-time MQSC configuration management and monitoring commands. For a full set of supported MQSC commands, see the WebSphere MQ Command Reference, or type HELP at the command prompt.

The commands below are used to invoke the WMQ MQSC utility. In this example, the group is named GROUP, the node is named MYNODE, and the queue manager is named MYQMGR.

	NOTE:	The group name used in the <code>nsqmqsc</code> command must be configured in the <code>mqgroup.ini</code> file. For more information, see M6-WMQ Configuration Files in section 4-4 .
---	--------------	--

```
$ nsqmqsc -m<GROUP>
\GROUP:\>DIR
WebSphere MQ Nodes under [\GROUP]:
[MY_NODE] - STATUS(ACTIVE)
[TOVA]- STATUS(UNKNOWN)
\OMEGAX01:\>cd MY_NODE
\OMEGAX01\MY_NODE\:>dir
Queue Managers under
[\GROUP\MY_NODE]: [MY_QMGR] -
STATUS(ACTIVE) [OMEGAX.QMG] -
STATUS(UNKNOWN)
\GROUP\MYNODE\:>CD MYQMGR
...
dis QMGR
...
ping CHANNEL(MY_CHANNEL)
```

Figure 6-1. M6-WMQ MQSC Utility Start Commands

	NOTE:	MQSC commands can only be executed when the target queue manager is active.
---	--------------	---

6.1.5 Using MQSC Scripts for Queue Manager (Fault) Recovery

Hardware and software failure can corrupt or wipe out WebSphere MQ configurations. Hardware failures can result in a total or partial loss of local configurations; software failure can result in a corrupted queue manager.

To perform fault recovery after hardware or software failure, prepare MQSC scripts that contain configuration information for all WebSphere MQ nodes. Then these scripts can be run using the `nsqmqsc` utility.

The example below executes the MQSC script `mymqsc.tst` on a target queue manager. In this example, the group is named GROUP, the node is named MYNODE, and the queue manager is named MYQMGR.

```
$ nsqmqsc -mGroup -nMyNode -qMyQMGR < mymqsc.tst
```

6.1.6 Using MQSC Configuration Snapshots for (Fault) Recovery

The `nsqmqsc` utility can be used to generate MQSC scripts automatically. The utility generates scripts by taking an MQSC “snapshot” of each of your queue managers. In the event of hardware or software failure, the scripts can be used to recover your queue managers.

	NOTE: MQSC scripts use default objects. If you do not want to use the default object, you will need to change it.
---	--

The command below generates an MQSC script for a queue manager. It tells the Workgroup server to retrieve all configuration data for the queue manager `MYQMGR`, and then return the MQSC equivalent. In this example, the group is named `GROUP`, the node is named `MYNODE` and the queue manager is named `MYQMGR`.

```
$nsqmqsc -mGroup -nMy_Node -qMyQMgr -d -r > MyQMgr.tst
```

where

- **-d** means “generate MQSC snapshot”
- **-r** adds REPLACE token to each MQSC command

	NOTE: MQSC snapshots can be taken even when queue managers are not running. The data comes from the Workgroup server's database.
---	---

Later, you can use the MQSC snapshot to recover your queue managers. In this example the following command would be issued:

```
$nsqmqsc -mGroup -nNode -qMyQMgr < MyQMgr.tst
```

	NOTE: To prepare for fault recovery, you should create MQSC snapshots of all existing and new queue managers, and re-take a snapshot each time you add a queue, channel, or process to a queue manager. The <code>nsqmqsc</code> command can be used in a script to backup queue manager definitions on a daily basis. Script can be invoked using Windows AT command or UNIX cron.
---	--

6.1.7 Creating and Removing Queue Managers

Workgroup servers and WMQ agents allow the remote creation of queue managers.

	NOTE: When you create a new queue manager, all the default queues, channels, processes, listeners, authentication information, client connections, and namelists under it are created automatically.
---	---

To create a new queue manager named `NewQMgr` (where `group = Group` and `node = MyNode`), issue the following commands:

```
$nsqmqsc -mGroup
cd MyNode
define NewQMgr
. . . .
Exit
```

	NOTE: M6-WMQ will not delete a queue manager unless the queue manager is stopped.
---	--

To delete an old queue manager named `OldQMgr` (where `group = Group` and `node = MyNode`), issue the following commands:

```
$nsqmqsc -mGroup CD MyNode
delete OldQMgr
. . . .
Exit
```

6.1.8 Executing User Action as a Detached Process

To run a detached process for a user action in Windows by executing commands to start and stop a Windows service: In the following example, the Workgroup Server (nsqmgr) is being started and stopped.

To start a Windows service:

```
net start APWMQWorkgroup_MQMDEV
```

To stop a Windows service:

```
net stop APWMQWorkgroup_MQMDEV
```

Chapter 7: Maintaining M6-WMQ

When M6-WMQ has been correctly configured, it requires virtually no maintenance. There are three potential performance problems that M6-WMQ system administrators should prepare for. Potential problems and their solutions are presented in the *M6-WMQ Potential Problems and Solutions* table.

The solutions are further explained in the sections that follow.

Table 7-1. M6-WMQ Potential Problems and Solutions

Potential Problem	Solution
M6-WMQ management database may become damaged.	Administrators should keep a backup management database.
Event logs may grow too large and overflow.	Administrators should enable/disable trace as necessary.
Events may become too numerous and network traffic may slow down.	Administrators should optimize network speed by changing certain configuration settings.

7.1 Enabling/Disabling Trace for Workgroup and WMQ Agents

If your log files grow too large, you may want to disable the trace function of your workgroup server and WMQ Agents. Normally, the trace function stores the following information in log files:

- A full dump of all connections being established and their corresponding locations
- A full dump of all PCF messages received by the workgroup server
- A full dump of all PCF messages sent by the workgroup server
- A full report on all objects being discovered
- Information about the internal memory-allocation table

To enable the trace from the command line:

```
nsmgr [console, Windows only] -m<group> -t/-ts/-tc/-tu/-ta
```

where:

- t: enable tracing of PCF messages
- ts: enable tracing of SQL database operations
- tc: enable communication trace
- tu: enable authorization trace
- ta: enable user authentication trace (mainly for debugging user authentication problems.)

Alternately, add the trace option to the system environment variable: `APWMQ_GS_OPTIONS`. `APWMQ_GS_OPTIONS` contains Workgroup Server startup options, space or semicolon delimited. (For example: `-mMQM;-u;-a`). These options are overridden by options specified on the command line. If running multiple Workgroup Server services and each instance requires different options, you can define environment variables of the form `APWMQ_GS_OPTIONS_<wgrp>`, where `<wgrp>` is the name of the workgroup that the Workgroup Server is processing. These settings will override those defined in `APWMQ_GS_OPTIONS`.

To enable trace on a WMQ Agent, enter the following command:

```
nsqmq [console, Windows only] [-sPortNumber] [-mgroup] -t  
[console, Windows only]
```

where **-t** instructs the WMQ Agent to produce trace output.

Alternately, add the trace option to the system environment variable: `APWMQ_LS_OPTIONS`.

7.2 Optimizing Network Speed

If the WebSphere MQ network is running slowly, do one or more of the following:

- Create smaller groups; each consisting of 5 to 10 managed nodes.
- Receive only important events by turning off all unnecessary events at the level of queue managers and workgroup server.
- Enter a longer auto-discovery period, which will allow less frequent polling of WebSphere MQ configuration data.

Note that both the workgroup server and the WMQ Agent perform polling. It is recommended to have auto-discovery set to a longer interval at the workgroup server level (1440 minutes, or once a day), or to deactivate (0 minutes). The default WMQ Agent discovery time is 720 minutes.

Chapter 8: M6-WMQ Security

M6-WMQ security supports user authentication and user authorization:

- User Authentication is managed via operating system (Windows) or via AutoPilot M6 security server.
- User authorization supported via local permission database (permits.ini) on workgroup or local agent levels. (This functionality is being phased out and replaced with Security Manager.)

8.1 AutoPilot M6-based User Authentication

AutoPilot M6 security server user authentication is always enabled for all AutoPilot M6 consoles. By default, the administration applications (M6-WMQ Explorer, Message Explorer) user authentication is not enabled. To enable user authentication for these applications modify the workgroup server command line option. Instead of `-a` (default), add one of the following options:

- +a1 -- Authenticate user via local system security
- +a2 -- Authenticate user via AutoPilot M6 security

If workgroup server is running on MS Windows system as a service, the command line options are passed via environment variable - `APWMQ_GS_OPTIONS`. On every workgroup server system, modify [`Install_Dir`]/`Config/groups/mqgroup.ini` file by adding next two lines:

```
MQM: :ATPDSNode=localhost
MQM: :ATPDSPort=8889
```

Repeat this step for every workgroup (MQM, in this case, is a name of a workgroup). `ATPDSNode` must be equal to the hostname of the AutoPilot M6 domain server; `ATPDSPort` must be equal to the AutoPilot M6 domain server security gateway port. Restart workgroup server. Now, every session between M6-WMQ Explorer or Message Explorer and M6-WMQ environment will be authenticated.

To add, remove or maintain user's ID and password, use the AutoPilot M6 domain console.

8.2 OS-based User Authentication

OS authentication is supported on Windows 2000 or later operating systems. The M6-WMQ workgroup server must be activated under account which has "SYSTEM" privileges; (for example: system). The M6-WMQ workgroup server must be started with `+a1` command line option. If M6-WMQ workgroup server is started as a service, the environment variable `APWMQ_GS_OPTIONS` must have `+a1` option.

In order to enable the workgroup server to use `+a1` on any operating system, you can either:

- Change the permissions on the necessary files and directories
- or
- Add the user that the workgroup server runs as to the necessary user groups so that it can access these files.

8.3 User Authorization ("u" Parameter)

Upon request, M6-WMQ performs security checks at two points:

- When a Workgroup server receives a command submitted on behalf of an external user (for example, from the M6-WMQ Explorer).
- When a WMQ Agent receives the command from the Workgroup server. Security settings at the Workgroup server and the WMQ Agent levels are independent of each other, in the sense that they don't need to be both active to implement security. In order for M6-WMQ to do this security check at both the Workgroup server and WMQ Agent levels, both the Workgroup server and the WMQ Agent must be started with the `"u"` parameter (`"-u"` parameter disables security checking).

8.3.1 How it Works

When a PCF command is submitted on behalf of an external user, the user ID of the incoming command is that of the external user (in this case the M6-WMQ Explorer logon user). If security checking is turned on, M6-WMQ matches this external user ID with the permission settings specified in the `permits.ini` file (see the next section for more information about this file). Both Workgroup server and WMQ Agents have their own `permits.ini` files, which means that access to WebSphere MQ objects can be authorized or denied at both the Workgroup server and the WMQ Agent level.

The Workgroup server will check the user ID of the incoming command against the contents of its `permits.ini` file before sending the command on to the WMQ Agent. The WMQ Agent has its own `permits.ini` file (unless the Workgroup server and the WMQ Agent run on the same node, in which case they share the same `permits.ini`) and again it performs a security check. These two checks are independent of each other and either one may fail a command. Figure 8-1 shows the two-level security checking available in M6-WMQ.

The result of the two-level security match is either zero or one.

- If it's a match and the result is one, the authority is granted.
- If it's a match and the result is zero, the authority is denied.
- If there's no match, the command is rejected.
- If the value is not a number, it is assumed to be zero (for example, `UserAuthority::*=yes` is the same as `UserAuthority::*=0`).

An advantage of performing security checks at the Workgroup server level is there is a central point of control and just one `permits.ini` file to manage. On the other hand, specifying security (that is, starting the component with the "+u" parameter) at the WMQ Agent level allows the owners of the WMQ Agent to control their own security, which is appropriate for a decentralized scheme. As a third alternative, if the user names are under central control in the installation, it is possible to let the native security system completely control access to the WebSphere MQ resources by specifying `UserAuthority::*=1` as the only statement in the Group and WMQ Agent's `permits.ini` file.

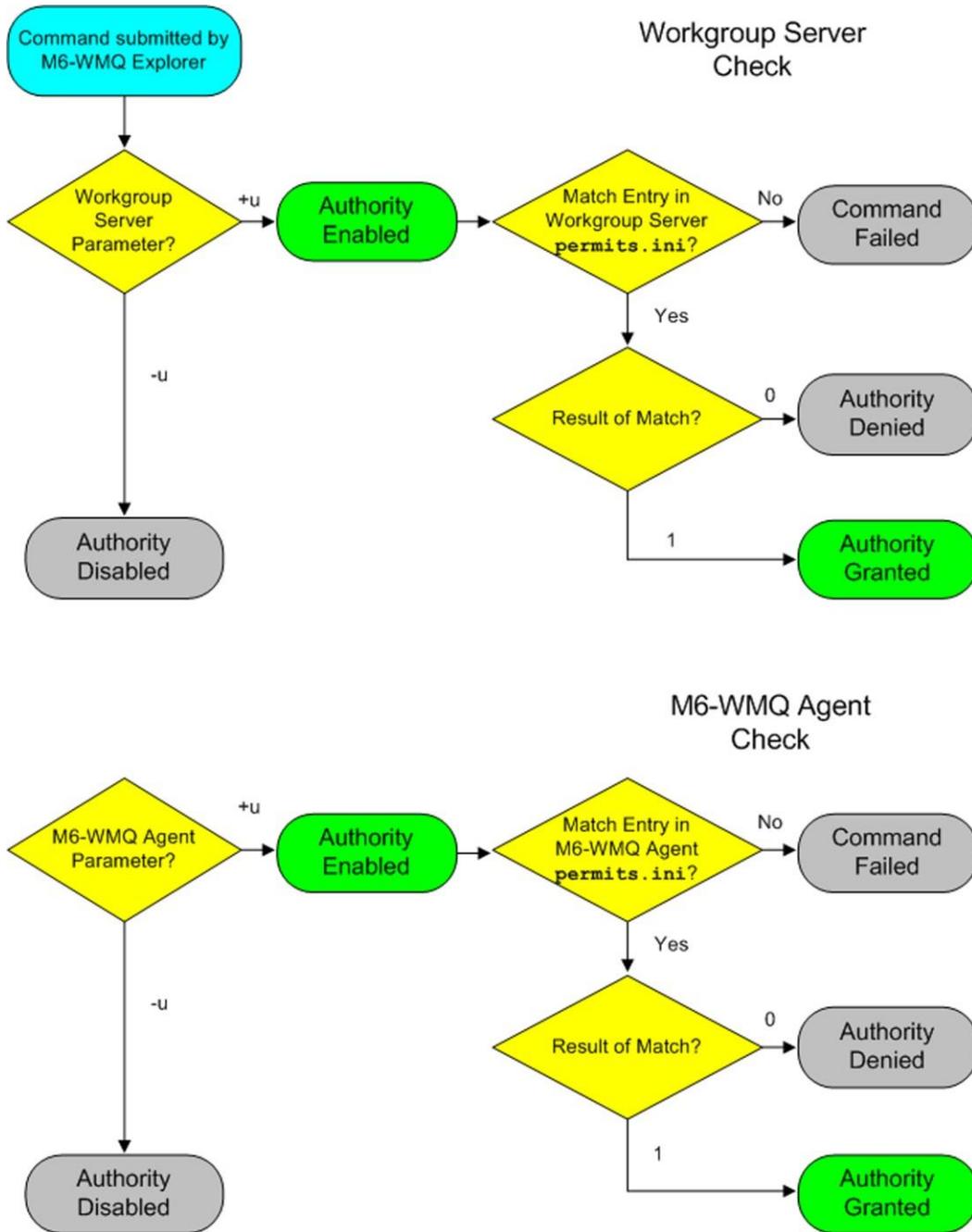


Figure 8-1. Two-level Security Checking by M6-WMQ

8.4 Editing Security Statements in `permits.ini` Files

The `permits.ini` file is a simple text file that specifies how a Workgroup server or a WMQ Agent checks incoming commands. A default copy of this file is provided with M6-WMQ installation media. There is one `permits.ini` located in the `[APWMQ_HOME]\config\groups` directory, where `[APWMQ_HOME]` is the M6-WMQ installation directory. If Workgroup server and WMQ Agent are running on the same system, they will share a common `permits.ini` file.

Anyone who has access to the computer where the Workgroup server or WMQ Agent is installed potentially has access to the `permits.ini`. Security features of the local system (example: RACF) must be used to control access to the `permits.ini` file.

Each `permits.ini` contains the following types of statements:

- Effective User Name statements
- User Authority statements
- TRUST statements
- Audit statements

The WMQ Agent and Workgroup server load the `permits.ini` file into memory at program startup and after each edit. This means that changes to `permits.ini` are effective immediately, without the need to stop and restart the WMQ Agent or Workgroup server. An exception to this is the z/OS systems where changes to the `permits.ini` file are immediately effective only if the dataset is a true sequential dataset. If the `permits.ini` file is a member of a partitioned dataset, changes may not be effective until the WMQ Agent is restarted.

The order of the Effective User Name and User Authority statements in the `permits.ini` affects how the Workgroup and WMQ Agents respond to each command. You should therefore understand how M6-WMQ checks each command against the statements found in `permits.ini`. Examples of `permits.ini` statements follow this section.

Note that if “+u” is not specified, none of the statements in the `permits.ini` file has any effect. In fact, in that case the `permits.ini` file is not read or otherwise accessed by M6-WMQ.

8.4.1 Object Specification

In `permits.ini` statements, where a fully qualified WebSphere MQ object specification is required, one of the following formats is required:

Workgroup Server Format:

```
<WORKGROUP>\<NODE>\<QMGR>\<OBJNAME>\<OBJTYPE>
```

WMQ Agent Format:

```
$SYSTEM$*\*\<QMGR>\<OBJNAME>\<OBJTYPE>
```

The Workgroup server and WMQ Agent formats are the same, except that the values for `<WORKGROUP>` and `<NODE>` at the WMQ Agent are replaced by wildcards and preceded by the marker `$SYSTEM$`.

In the descriptions of the statements of `permits.ini`, below, this fully qualified WebSphere MQ object is indicated by `<OBJECT>`:

The subfields of an `<OBJECT>` are:

`<WORKGROUP>` is the name of the M6-WMQ Workgroup server.

`<NODE>` is the name of the WMQ Agent under the control of the specified Workgroup server.

`$SYSTEM$` is a keyword required only to identify a WMQ Agent.

`<QMGR>` is the name of a WebSphere MQ queue manager under control of the specified WMQ Agent.

`<OBJNAME>` is the name of a queue, process, namelist, channel, service, authentication informations or listener.

`<OBJTYPE>` a text string `EXOT_xxx` representing the type of object specified in the `<OBJNAME>` field (see Table 8-1). In most cases, the object type can be specified with a “wildcard” place marker.

For example, a queue named `TEST.Q`, on queue manager `CSQ1`, managed by WMQ Agent `S390`, controlled by Workgroup server `MQM` would be specified in `permits.ini` as follows:

```
MQM\S390\CSQ1\TEST.Q\EXOT_QUEUE
```

All or part of an object specification may be replaced by an asterisk to select a range of objects. For example, to select all queues under the same queue manager, the following would be specified:

```
MQM\S390\CSQ1*\EXOT_QUEUE
```

For some commands, not all fields need to be specified. For example, to specify a queue manager in a command that involves only the queue manager, the `<OBJNAME>` field need not be present. To specify the `CSQ1` queue manager, use a wildcard place marker for the object name. The statement will look like:

```
MQM\S390\CSQ1*\EXOT_Q_MANAGER
```

Note that the `<OBJTYPE>` is still present in the specification.

The kind of object and the number of subfields to specify depends upon the command being validated. One way to determine the exact format of the object is to use the Audit tracing facility, described below. It allows you to inspect the output after executing a command to determine the object format for that command.

8.4.2 Other Field Specifications

The other fields used in `permits.ini` file specification statements are:

- `<USERNAME>` is the Windows `USERNAME` variable that defines the operator of the M6-WMQ Explorer or, in some cases; the user ID runs under the Workgroup server. This field must be in upper case, unless there is an Effective User Name replacement active. In that case, the name should be in the case of the Effective User Name (the name is the Effective User Name.)
- `<ALTUSER>` is the alternate or effective user name, in the same case as registered on the destination WMQ Agent node.
- `<PCFCMD>` is a text string representing either a WebSphere MQ PCF command or an AP M6 WMQ PCF command. Refer to the WebSphere MQ PCF Command Codes and M6-WMQ Extended PCF Command Codes tables, for a listing of relevant codes.
- `<IPADDR>` is the TCP/IP address of the connection where the command was originally submitted. This must be a decimal-dotted IP address. Names are not valid.
- `<APPLNAME>` is the name of the application attempting to make a connection. For example, the name of the M6-WMQ Explorer is `NSQEXPLR`.
- `{none|denied|granted|all}` is the Audit tracing level required; `none` if no tracing is required, `denied` if required when a request fails, `granted` if required when a request is allowed, or `all` if tracing is unconditional.
- `{1|0}` is the level of authority that you want to grant to the user: Number 1 (true, grant authority) or 0 (false, deny authority).

8.4.3 Effective User Name Statements

In a large distributed enterprise, the number of external users who may require some form of M6-WMQ access to local WebSphere MQ objects may be quite high, and may change frequently. M6-WMQ offers a feature that can simplify management of external user IDs: effective user name mapping. Through effective user name mapping, an external user ID can be mapped to any internal user ID.

For example, if remote user JSMITH submits a command, instruct the WMQ Agent to convert JSMITH to USER_A on all commands, so the command from JSMITH will be given to the queue manager as if it came from USER_A. Effective user name mapping allows you to define just a few local user IDs to which the large set of external IDs can be mapped. Remote user JSMITH does not need to know that this mapping is taking place, so the local user IDs such as USER_A can be maintained privately.

For example: a local user profile for USER_B, and assign only browse privileges to that user ID. USER_ADMIN might be a profile that has full administration capabilities, and profile USER_APPL might be an application user profile that has full access to a few application-specific objects. Effective user name mapping can be used to assign external users to one of the above-mentioned local user IDs, thus minimizing your security administration.

If User ID mapping is used at the WMQ Agent, replication of the `permits.ini` files to every WMQ Agent is required. In general, the usual place to do mapping is at the Workgroup server so you only have one file that has to be updated when user IDs change.

It is possible to set up a two-tier mapping scheme: Workgroup server can map its external users to group-specific (such as regional) user IDs, and these in turn can be mapped to different user IDs at each node in the Workgroup.

Effective user name mapping definitions are specified in the `permits.ini`. The format of the effective user name statements is as follows:

```
EffectiveUserName::<USERNAME>:::<OBJECT>=<ALTUSER>
```

The `<USERNAME>` in this statement is always uppercase; `<ALTUSER>` has the case as registered at the destination WMQ Agent.



NOTE:

You may use an asterisk (*) anywhere in the EffectiveUserName string after `::<USERNAME>::` as a wildcard match. For example, `EffectiveUserName::JSMITH::*=USER_A` maps all requests from JSMITH to USER_A. This meets Workgroup server syntax. To meet WMQ Agent syntax `$$SYSTEM$/` is required as prefix to `<OBJECT>`.

8.4.4 User Authority Statements

In addition to mapping external user identifiers to internal user identifiers (if “+u” parameter is specified), M6-WMQ will perform authorization checks on all commands received by the Workgroup server or WMQ Agent. The specifications in the `permits.ini` file will either grant or deny permission to submit a command to the WebSphere MQ command server.

To understand how permission to execute a command is defined, it is important to understand how commands are identified by M6-WMQ. Commands are classified by the type of object they target (for example, queue manager, queue, and channel). Presently there are thirteen object types supported by M6-WMQ. These are listed in the first column in the table below, together with their value; WS = Workgroup server and WA = WMQ Agent.

Object Type	Value	Applicable To
Queue Manager	EXOT_Q_MGR	WS, WA
Channel	EXOT_CHANNEL	WS, WA
Process	EXOT_PROCESS	WS, WA
Node	EXOT_MQNODE	WS
Workgroup server	EXOT_MANAGER	WS
Queue	EXOT_QUEUE	WS, WA
Cluster Queue	EXOT_Q_CLUSTER	WS, WA
Namelist	EXOT_NAMELIST	WS, WA
Cluster Queue Manager	EXOT_CLUSTER_QMGR	WS, WA
Cluster Channel	EXOT_CL_CHANNEL	WS, WA
Authentication Information	EXOT_AUTHINFO	WS, WA
Channel Listener	EXOT_LISTENER	WS, WA
Service	EXOT_SERVICE	WS, WA
Topic	EXOT_TOPIC	WS, WA
Subscription	EXOT_SUBSCRIPTION	WS, WA

Individual commands are defined by their PCF command code (`MQCMD_*`) value. For example, the PCF command code to alter a queue (`MQCMD_CHANGE_Q`) is 8. The PCF command code values are defined in the file `cmqfc.h`, and are included in Table 8-2 in this chapter. M6-WMQ extends the PCF command set with *Extended PCF Commands* (`EXCMD_*`). These codes are given in M6-WMQ Extended EPCF Command Codes table. (See Table 8-3 in this chapter.)

Because the M6-WMQ authority service identifies commands by both object type and individual PCF command code, a user can be granted or denied permission for all commands within an object type by specifying the object type value. The User Authority lines in the `permits.ini` file have the following format:

```
UserAuthority::::<OBJECT>::

```

Note that all parameters within angle brackets `<>` can be replaced by the wildcard value (`*`). With the use of the wildcard and the above formats, an authority specification can be defined to grant or deny (any or a specific user) permission to submit WebSphere MQ command server (any or a specific PCF command) which operates on a given OBJECT TYPE (any or a specific object type) on an OBJECT (any or a specific object of that type) on QUEUE MANAGER (any or a specific queue manager) on a NODE (any or a specific node) which is under control of a work GROUP (any or a specific Workgroup server).

Some commands are initiated by the Workgroup server rather than the M6-WMQ Explorer. For those commands, <USERNAME> will be the user under which the Workgroup server is running. Otherwise, <USERNAME> value that commands come in with is user ID under which M6-WMQ Explorer is running.

8.4.4.1 Setting-up permits.ini to Restrict Access

Precedence

Authority lines with more specific, finer permission rules (for example, those using few or no wildcards) should precede lines with coarser generic rules (for example, those using all wildcards). Any permission granted or denied in lines that come first in the file overrides any matching permission lines that appear later in the file. Hence, if you grant a user permission to delete a particular queue in line 20, denying this permission in a subsequent line will have no effect.

Generic lines

If these next lines are enabled, a match will be made on these lines before any specific EXCMD_INQUIRE_xxx and MQCMD_INQUIRE_xxx line below.

```
UserAuthority::*:*\\*\\*\\*::EXCMD_INQUIRE_*=1 # enable all M6-WMQ inquire cmds
UserAuthority::*:*\\*\\*\\*::MQCMD_INQUIRE_*=1 # enable all WebSphere MQ inquire cmds

# Workgroup server related cmds # UserAuthority::*:*\\*\\*\\*::EXCMD_INQUIRE_ACTIVE_MANAGER=1
UserAuthority::*:*\\*\\*\\*::EXCMD_INQUIRE_MANAGER=1
UserAuthority::MUSR_MQADMIN:*\\*\\*\\*::EXCMD_INQUIRE_LICENSE=1
UserAuthority::MUSR_MQADMIN:*\\*\\*\\*::EXCMD_CHANGE_MANAGER=1
UserAuthority::MUSR_MQADMIN:*\\*\\*\\*::EXCMD_DB_BACKUP=1
UserAuthority::MUSR_MQADMIN:*\\*\\*\\*::EXCMD_DB_RESTORE=1
UserAuthority::*:*\\*\\*\\*::EXCMD_REGISTER_MQNODE=1
UserAuthority::*:*\\*\\*\\*::EXCMD_SHUTDOWN_MANAGER=1

# Display of events to GUI is controlled by Workgroup server
# Executed as user SYSTEM UserAuthority::SYSTEM:*\\*\\*\\*::EXCMD_MANAGER_EVENT=1
UserAuthority:: SYSTEM:*\\*\\*\\*::EXCMD_ALTER_EVENT=1 UserAuthority::
SYSTEM:*\\*\\*\\*::EXCMD_STATE_EVENT=1 UserAuthority::
SYSTEM:*\\*\\*\\*::MQCMD_Q_MGR_EVENT=1 UserAuthority::
SYSTEM:*\\*\\*\\*::MQCMD_PERFM_EVENT=1 UserAuthority::
SYSTEM:*\\*\\*\\*::MQCMD_CHANNEL_EVENT=1

# Purge events granted to WMQ user MUSR MQADMIN
UserAuthority::MUSR_MQADMIN:*\\*\\*\\*::EXCMD_DELETE_MANAGER_EVENT=1
UserAuthority::MUSR_MQADMIN:*\\*\\*\\*::EXCMD_DELETE_ALTER_EVENT=1
UserAuthority::MUSR_MQADMIN:*\\*\\*\\*::EXCMD_DELETE_Q_MGR_EVENT=1
UserAuthority::MUSR_MQADMIN:*\\*\\*\\*::EXCMD_DELETE_PERFM_EVENT=1
UserAuthority::MUSR_MQADMIN:*\\*\\*\\*::EXCMD_DELETE_CHANNEL_EVENT=1
UserAuthority::MUSR_MQADMIN:*\\*\\*\\*::EXCMD_DELETE_CONFIG_EVENT=1

# Display of event details granted to any user
UserAuthority::*:*\\*\\*\\*::EXCMD_INQUIRE_MANAGER_EVENT=1
UserAuthority::*:*\\*\\*\\*::EXCMD_INQUIRE_ALTER_EVENT=1
UserAuthority::*:*\\*\\*\\*::EXCMD_INQUIRE_Q_MGR_EVENT=1
UserAuthority::*:*\\*\\*\\*::EXCMD_INQUIRE_PERFM_EVENT=1
UserAuthority::*:*\\*\\*\\*::EXCMD_INQUIRE_CHANNEL_EVENT=1
UserAuthority::*:*\\*\\*\\*::EXCMD_INQUIRE_CONFIG_EVENT=1
```

Any user needs the next three basic permissions to connect to a WMQ Agent.

```
UserAuthority::*:MQM\*\*\*:EXCMD_OPEN_EVENT=1
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_CLOSE_EVENT=1
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_MQ_OPEN=1
```

Node commands

```
UserAuthority::*:MQM\LOCALSUN\*:EXCMD_INQUIRE_MQNODE_NAMES=0
```

```
UserAuthority::*:MQM\STOWE\*\*\*:EXCMD_INQUIRE_MQNODE_NAMES=0
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_INQUIRE_MQNODE_NAMES=1
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_INQUIRE_MQNODE=1
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_MANAGE_MQNODE=1
```

example using single char '?' wildcard

```
UserAuthority::*:MQM\ST??E\*\*\*:EXCMD_UNMANAGE_MQNODE=0
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_UNMANAGE_MQNODE=1
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_CHANGE_MQNODE=1
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_DELETE_MQNODE=1
```

qmgr

```
UserAuthority::*:MQM\*\*\*:MQCMD_INQUIRE_NAMELIST_NAMES=1
```

```
UserAuthority::*:MQM\*\*\*:MQCMD_INQUIRE_Q_MGR=1
```

```
UserAuthority::*:MQM\D-HC00-28\TEST2\*\*\*:MQCMD_PING_Q_MGR=0
```

```
UserAuthority::*:MQM\*\*\*:MQCMD_PING_Q_MGR=1
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_START_Q_MGR=1
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_MQ_DISCOVER=1
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_INQUIRE_Q_MGR_NAMES=1
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_INQUIRE_CLUSQMGR_NAMES=1
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_DELETE_Q_MGR=1
```

```
UserAuthority::MUSR_MQADMIN::MQM\*\*\*:EXCMD_STOP_Q_MGR=1
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_INQUIRE_TEXT_FILE=1
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_DELETE_Q_MGR=1
```

queues

```
UserAuthority::*:MQM\D-HC\*\*\*\EXOT_Q_ALIAS::MQCMD_INQUIRE_Q_NAMES=0
```

```
UserAuthority::*:MQM\*\*\*:MQCMD_INQUIRE_Q_NAMES=1
```

```
UserAuthority::*:MQM\*\*\*:MQCMD_INQUIRE_Q_NAMES=1
```

```
UserAuthority::*:MQM\*\*\*:MQCMD_INQUIRE_Q=1
```

```
UserAuthority::*:MQM\*\*\*:EXCMD_INQUIRE_Q_RUNTIME=1
```

```
UserAuthority::*:MQM\*\*\*:MQCMD_CREATE_Q=1
```

```
UserAuthority::*:MQM\*\*\*:MQCMD_CHANGE_Q=1
```

```
UserAuthority::*:MQM\*\*\*:MQCMD_CLEAR_Q=0
```

```
UserAuthority::*:MQM\*\*\*:MQCMD_DELETE_Q=0
```

```
UserAuthority::*:MQM\*\*\*:MQCMD_COPY_Q=1
```

```
UserAuthority::*:MQM\*\*\*:MQCMD_RESET_Q_STATS=1
```



```
UserAuthority::MUSR_MQADMIN::*\*\*\*::EXCMD_MG_NEW=1
UserAuthority::MUSR_MQADMIN::*\*\*\*::EXCMD_MG_FIND=1
UserAuthority::MUSR_MQADMIN::*\*\*\*::EXCMD_MG_MODIFY=1

# authorization info cmds (MQ 5.3)
UserAuthority::MUSR_MQADMIN::MQM\*\*\*::MQCMD_CHANGE_AUTH_INFO=1
UserAuthority::MUSR_MQADMIN::MQM\*\*\*::MQCMD_COPY_AUTH_INFO=1
UserAuthority::MUSR_MQADMIN::MQM\*\*\*::MQCMD_CREATE_AUTH_INFO=1
UserAuthority::MUSR_MQADMIN::MQM\*\*\*::MQCMD_DELETE_AUTH_INFO=1
UserAuthority::MUSR_MQADMIN::MQM\*\*\*::MQCMD_INQUIRE_AUTH_INFO=1
UserAuthority::*::MQM\*\*\*::MQCMD_INQUIRE_AUTH_INFO_NAMES=1
UserAuthority::MUSR_MQADMIN::MQM\*\*\*::MQCMD_INQUIRE_AUTH_INFO_NAMES=1
Deny all other cmds to all users (this should be the last command)
UserAuthority::*\*\*\*\* =0
Trust all connecting users from any IP address perform any application

# TRUST::user_name::ip_adrs::app_name
TRUST::*::*=1

Allow all command audit
Audit::*=*all
```

8.4.5 TRUST Statements

The TRUST statement is used to enhance security by limiting the connections that can be made to the Workgroup server. The TRUST statement is effective only in the Workgroup server's `permits.ini` file and is ignored by the WMQ Agent. The `permits.ini` file for the Workgroup server must contain a TRUST statement or connections will be denied.

The TRUST lines in the `permits.ini` file have the following format:

```
TRUST::::<IPADDR>::={1|0}
```

A connection to the Workgroup server is allowed only if the first TRUST statement that is matched specifies 1 for the trust value. If no match is found, or the trust value of the first matching statement is 0 (zero), a connection is not allowed.

Any or all of the operands of the TRUST statement may contain mask specifications. For example, to allow any user in the 10.0.0 subnet to connect using any Nasted program, the following statement is used:

```
TRUST::*::10.0.0.*::NSQ*=1
```

8.4.6 Audit Statements

The Audit statement is used to trace the granting or denying of permissions. The Audit lines in the `permits.ini` file have the following format:

```
Audit::::<OBJECT>={denied|granted|all}
```

If you wish to trace TRUST statements, specify the `<IPADDR>` value or mask for the `<OBJECT>`. Depending on the platform, the output from the trace will be placed either in the `stdout` stream or in a file in the M6-WMQ installation directory with the name `<USER_NAME>_<ALTUSER>.ADT`. The platforms that store the trace in a file are Windows 2000 and XP and UNIX. All other supported platforms place the trace in the `stdout` stream. For example, if a Windows Workgroup server translates the user name EJACKSON to "ej", requests initiated by this user will be placed in a file named `EJACKSON_ej.ADT`.

8.5 Examples of permits.ini Statements

COMMENTS

General usage rules:

All EXCMD command/EXOT object type strings and MQCMD commands are listed in header files

APWMQ\tools\include\nexcmd.h and [WebSphereMQ_INSTALL]\tools\C\include\cmqcfh respectively.

1. This file is buffered in memory when Workgroup server and WMQ Agent start and the '+u' option is used.
2. Keep in-line comments short to minimize memory usage and parsing time. Do not use an '=' in comments; it will cause parsing errors..
3. Add lengthy user comments to bottom of the file (after the 'deny all other cmds' line) for faster parsing. Parsing will end on that line.
4. To make a production version of permits.ini, copy this sample file to use as a template, delete all comments and lines not needed, and add/edit permit lines as needed.
5. Syntax rules: TRUST::effective_user_name::ip_addr_or_mask:appl_name=1 or 0

Example:

```
TRUST::MUSR_MQADMIN::127.0.0.1::*=1
TRUST::*::123.45.67.*::NSQEXPLR=1
```

Trust only admin on localhost connections and from any user of application M6-WMQ Explorer with IP address 123.45.67*.

Audit::effective_user_name::ip_address_mask=1 or 0

Example:

```
Audit::MUSR_MQADMIN::123.0.0.*=1
```

Audit all commands by users mapped to MUSR_MQADMIN connected from 123.0.0.*

EffectiveUserName::user_name::obj_string=alt_user_name

Workgroup server maps the received client user name to an 'effective' user name. If not mapped, the client user name is used for further permission checks.

If the Workgroup server permits the command, the command is forwarded to the WMQ Agent with the 'effective' user name or original client name.

WMQ Agent does its own 'effective' user name lookup and permission checks.

WMQ Agent only checks permission lines which contain \$SYSTEM\$ as prefix of object string.

USER_NAME must be in upper case.

A Windows user name that contains a space must be truncated to the first name only.

Effective user name must either have explicit WebSphere MQ permissions for the object or be in the mqm group on the target node.

Simplest case is to set obj_string to '*' but one could map a user name based on a specific object string. The obj_string can be wildcard or group:obj_type or ip_adrs_or_mask.

Examples:

```
EffectiveUserName::HOWARD::123.0.5.246=MUSR_MQADMIN
```

Map user howard at IP address 123.0.5.246 to MUSR_MQADMIN

This format is used when M6-WMQ Explorer displays the group and nodes.

```
EffectiveUserName::HOWARD::*=MUSR_MQADMIN
```

Map user howard from any IP address to MUSR_MQADMIN

```
EffectiveUserName::HOWARD::MQM\SUNBOSS\*=MUSR_MQADMIN
```

Map user howard to MUSR_MQADMIN for any command for group MQM and node

SUNBOSS

This format is used when M6-WMQ Explorer or Workgroup server executes a command on an object.

```
EffectiveUserName::HOWARD::MQM\SUNBOSS\SUN1\EXOT_QUEUE=MUSR_MQADMIN  
(v2 format)
```

Similar to previous example but more specifically for a queue object command, for group MQM, node SUNBOSS and queue manager SUN1.

```
UserAuthority::eff_user_name::[$SYSTEM$\]group\node\qmgr\obj_name\obj_type::cmd_name=1 or 0
```

- a. In format v2 insert all attributes group, node, qmgr, obj_name, obj_type and cmd_name in every UserAuthority line. If a wildcard '*' is used and the attribute does not apply for matching the key, it will be ignored and considered a match. Attributes in the key that don't apply to the command will have a value "dc", meaning "don't care". A wildcard in the permits line will match on the "dc".

Example: If permit line is this:

```
UserAuthority::*:\*\*\*\*:EXCMD_INQUIRE_ACTIVE_MANAGE
```

R=1 the audit file will show this:

```
NSQPRMT(HOWARD[MUSR_MQADMIN]):
```

```
AUTH_TOKEN("UserAuthority::MUSR_MQADMIN::MQM\dc\dc\dc\
```

```
EXOT_MANAGER::EXCMD_INQUIRE_ACTIVE_MANAGER") defined -- GRANTED!
```

The dc's indicate node, queue manager and object name do not apply for this command.

In format v1, you may have less than or the exact attributes as the key, but may not have more than the key. The latter is not considered a match. See related item (e), Long vs Short Form.

- b. All fields in v1 and v2 syntax allow one or more wild card characters. See 6.
- c. No extra spaces are allowed after a field, or before and after the '='.
- d. "\$SYSTEM\$" string is required at the start of the permits object string for lines to be scanned only by WMQ Agent.

e. Long vs Short Form:

The long form of a permits line specifies each attribute of the object string expected in a command key to be matched. If a command is for a specific queue, such as change a queue, then the key will contain group\node\qmgr\q_name\obj_type and the permits line might look like this:

```
UserAuthority::*:**\*\*\*:8=1 # v1 format
```

```
UserAuthority::*:**\*\*\*:MQCMD_CHANGE_Q=1 # v2 format
```

You can also use a short form, which specifies only the first few attributes, followed by a '*' wildcard. At the minimum, you must specify "group*". If these attributes match the command key, then it's a match. Using the above example, you might specify only the group and the wildcard to match on everything that follows:

```
UserAuthority::*:**\*:8=1 # v1 format
```

```
UserAuthority::*:**\*:MQCMD_CHANGE_Q=1 # v2 format
```

or specify the group and node:

```
UserAuthority::*:**\*:8=1 # v1 format
```

```
UserAuthority::*:**\*:MQCMD_CHANGE_Q=1 # v2 format
```

f. obj_type EXCEPTION:

If command is

```
MQCMD_INQUIRE_CHANNEL_NAMES
```

the first instance of command key will have 3 attributes:

```
group\node\qmgr\1::20 # v1 format
```

```
group\node\qmgr\EXOT_QMGR::MQCMD_INQUIRE_CHANNEL_NAMES # v2 format
```

But instances for individual channels will have 4 attributes and the channel type instead of object type:

```
group\node\qmgr\chl_name\chl_type::20 # v1 format
```

```
group\node\qmgr\MQCHT_xxx::MQCMD_INQUIRE_CHANNEL_NAMES # v2 format
```

where:

```
xxx=ALL,SENDER,SERVER,RECEIVER,REQUESTER,CLNTCONN,SVRCONN,
CLUSRCVR, or CLUSSDR.
```

Unless absolutely necessary, avoid confusion and use short wildcard form:

```
group\node\qmgr\*:20 # v1 format
```

```
group\node\qmgr\*\*:MQCMD_INQUIRE_CHANNEL_NAMES # v2 format
```

where:

group, node, qmgr can also be wildcards.

6. Wildcards

- a. A wildcard '*' is allowed in place of any field, in the end or middle of any field.

Examples: * (a group, node, qmgr, object, object type or command)

\MQM* (groups that start with MQM)

\SUN* (nodes that start with SUN)

\SUN*NY (nodes that start with SUN and end with NY)

\QMP* (qmgrs that start with QMP)

\EXCMD_INQUIRE_* (all M6-WMQ inquire commands in v2 format)

\EXCMD_*_EVENT (all M6-WMQ event display commands in v2 format)

- b. A single character '?' wildcard means match on any one character.

Example: \SUN?NY will match nodes SUNANY or SUNBNY

- c. Using v2 format, you can use a wildcard to enable/disable a command set thereby avoiding the need for one permit line per command.

Examples:

```
UserAuthority::MQOP::*\*\*\*:EXCMD_INQUIRE*=1
```

Enable all M6-WMQ inquire commands for user MQOP.

```
UserAuthority::*:\*\*\*:MQCMD_INQUIRE*=1
```

Enable all WebSphere MQ inquire commands for all users.

```
UserAuthority::*:\*\*\*::EXCMD_INQUIRE_*_EVENT=0
```

Prohibit displaying of all events.

7. More specific permit lines must come before the more generic permit lines. Otherwise a command may be erroneously permitted or denied.
8. Enable authorization checks (+u option) by the Workgroup server for centralized access control, using one `permits.ini` file, with easier maintenance. Enable authorization checks by the WMQ Agent for distributed and more specific access control (using one or more `permit.ini` files), but harder to maintain.
9. Audit .ADT file messages format:

An audit file, if generated for a user, has the name

```
<clientUserName>_<effectiveUserName>.ADT.
```

The .ADT files contain audit trace messages from the Workgroup server or WMQ Agent that show whether a command was denied or granted. In format v2, a group, node, queue manager, object name or object_type attribute with value "dc" means "don't care", that is, the attribute does not pertain to the command. "dc" will match the wildcard character "*" in a permits line attribute.

In the old v1 format, "dc" is not used. You will see a text interpretation of the object type and command within brackets.

Examples Format v2:

```
NSQPRMT(HOWARD[MUSR_MQADMIN]):
```

```
AUTH_TOKEN("UserAuthority::MUSR_MQADMIN::MQM\dc\dc\dc\
```

```
EXOT_MANAGER::EXCMD_OPEN_EVENT") defined -- GRANTED!
```

```
NSQPRMT(HOWARD[MUSR_MQADMIN]):
```

```
AUTH_TOKEN("UserAuthority::MUSR_MQADMIN::MQM\dc\dc\dc\
```

```
EXOT_MANAGER::EXCMD_INQUIRE_ACTIVE_MANAGER") defined -- GRANTED!
```

```
NSQPRMT(MUSR_MQADMIN[MUSR_MQADMIN]):
AUTH_TOKEN("UserAuthority::MUSR_MQADMIN::$SYSTEM$\\
dc\\dc\\QM_d_hc00_28\\dc\\EXOT_Q_MGR::MQCMD_PING_Q_MGR") defined -- GRANTED!
```

```
NSQPRMT(HOWARD[MUSR_MQADMIN]):
AUTH_TOKEN("UserAuthority::MUSR_MQADMIN::MQM\\D-HC00-28\\
QM_d_hc00_28\\S_d_hc00_28\\EXOT_Q_CHANNEL::MQCMD_DELETE_CHANNEL") defined --
DENIED!
```

```
Example Format v1: NSQPRMT(HOWARD[queueadmin]):
AUTH_TOKEN("UserAuthority::queueadmin::MQM\\STOWE\\HP1\\
NASTEL.ADMIN.AUDIT.QUEUE\\11::8") [EXOT_QUEUE:MQCMD_CHANGE_Q] defined --
DENIED!
```

10. WMQ Agent and Workgroup server Restart

You may edit the `permits.ini` file without restarting WMQ Agent or Workgroup server. The file modification will be detected and the file reloaded while WMQ Agent and Workgroup server are running.

11. Do I have to enter one line for each MQCMD and EXCMD command with deny or grant?

No. If the number of deny commands is much less than the number of grant commands for a user or user group, then create specific "deny" permission lines and one generic "grant" line to permit everything else. If the opposite is true, create specific "grant" permission lines and one generic "deny" line to prevent everything else.

You can structure the permits lines as follows:

Audit lines

Trust lines

EffectiveUserName lines

UA denies for all users

UA grants for all users

UA permits lines for effective user 1:

Denies for user 1

(select from cmd for WMQ Agent, Workgroup server, Node, Qmgr, Queue, Process, channel, MMF, etc)

Grants for user 1 (select from same set)

Generic deny all else for user 1

Repeat above for each effective user.

UA generic deny line all other cmds to all users.

8.6 Use Case Examples

The following two examples identify user authorization supported via a local permission database (permits.ini) that has been taken from actual users. They have been verified by Nastel support.

These examples show three categories of authorizations:

- **WMQADMIN** - grants access to all MQ objects; user can access and configure all MQ objects.
- **OPRUSER** - grants access to all MQ objects; but user cannot delete any MQ objects.
- **DEVUSER** - grants limited access to MQ objects and MMF commands.

Although generic, the user can substitute their information into these examples as required.

8.6.1 Use Case Example 1

```
#SYNTAX_V2                                     # Required format version ID
TRUST::*:*:*=1
Audit::*:*=denied

## Admin Group
EffectiveUserName::MQM::*=WMQADMIN
EffectiveUserName::SYSTEM::*=WMQADMIN
EffectiveUserName::ADMIN::*=WMQADMIN
EffectiveUserName::MUSR_MQADMIN::*=WMQADMIN

# Admin authorizations
UserAuthority::WMQADMIN::*=1                    #Grant all access to all objects

## Operations Group (limited access)
EffectiveUserName::OPERATIONS::*=OPRUSER

## EffectiveUserName::ARTHUR::*=OPRUSER
# OPRUSER authorizations
# Restrict deletion
UserAuthority::OPRUSER::*\*\*::EXCMD_DELETE*=0
UserAuthority::OPRUSER::*\*\*::MQCMD_DELETE*=0

# Grant all other commands
UserAuthority::OPRUSER::*=1

## Developer Group (limited access to MQ objects, some MMF commands)
EffectiveUserName::DEVELOPER::*=DEVUSER
## EffectiveUserName::ARTHUR::*=DEVUSER

# DEVUSER authorizations
```

Display of events to GUI is controlled by Workgroup

```
UserAuthority::DEVUSER::*\*\*::EXCMD_MANAGER_EVENT=1
UserAuthority::DEVUSER::*\*\*::EXCMD_ALTER_EVENT=1
UserAuthority::DEVUSER::*\*\*::EXCMD_STATE_EVENT=1
UserAuthority::DEVUSER::*\*\*::MQCMD_Q_MGR_EVENT=1
UserAuthority::DEVUSER::*\*\*::MQCMD_PERFM_EVENT=1
UserAuthority::DEVUSER::*\*\*::MQCMD_CHANNEL_EVENT=1
```

Generic commands that do not use node name.

```
UserAuthority::DEVUSER::*\*\*\*::EXCMD_OPEN_EVENT=1
UserAuthority::DEVUSER::*\*\*\*::EXCMD_INQUIRE_ACTIVE_MANAGER=1
UserAuthority::DEVUSER::*\*\*\*::EXCMD_CLOSE_EVENT=1
UserAuthority::DEVUSER::*\*\*\*::EXCMD_MQ_OPEN=1
```

General inquiry without node name

```
UserAuthority::DEVUSER::*\*\*\*\*::EXCMD_INQUIRE_*=1
UserAuthority::DEVUSER::*\*\*\*\*::MQCMD_INQUIRE_*=1
```

Various authorizations

```
UserAuthority::DEVUSER::*\*\*\*\*::EXCMD_MANAGE_MQNODE=1
UserAuthority::DEVUSER::*\*\*\*\*::MQCMD_PING_CHANNEL=1
UserAuthority::DEVUSER::*\*\*\*\*::MQCMD_PING_Q_MGR=1
UserAuthority::DEVUSER::*\NODE1\*\*\*\*::MQCMD_CREATE_*=1
UserAuthority::DEVUSER::*\NODE2\*\*\*\*::MQCMD_CREATE_*=0
```

MMF authorizations

```
UserAuthority::DEVUSER::*\*\*\SYSTEM.*\*\*::EXCMD_MG_BROWSE=0
UserAuthority::DEVUSER::*\*\*\NASTEL.*\*\*::EXCMD_MG_BROWSE=0
UserAuthority::DEVUSER::*\*\*\*\*::EXCMD_MG_BROWSE=1
UserAuthority::DEVUSER::*\*\*\*\*::EXCMD_MG_FIND=1
UserAuthority::DEVUSER::*\*\*\SYSTEM.*\*\*::EXCMD_MG_NEW=0
UserAuthority::DEVUSER::*\*\*\NASTEL.*\*\*::EXCMD_MG_NEW=0
UserAuthority::DEVUSER::*\*\*\*\*::EXCMD_MG_NEW=1
UserAuthority::DEVUSER::*\*\*\*\*::EXCMD_MG_DELETE=0
UserAuthority::DEVUSER::*\*\*\*\*::EXCMD_MG_COPY=0
UserAuthority::DEVUSER::*\*\*\*\*::EXCMD_MG_MOVE=0
UserAuthority::DEVUSER::*\*\*\*\*::EXCMD_MG_MODIFY=0
```

Most restrictive

```
UserAuthority::DEVUSER::*=0
```

Deny all other cmds

Customer Group (limited access to MQ objects, some MMF commands)

```
EffectiveUserName::CUSTOMER::*=CMRUSER
EffectiveUserName::ARTHUR::*=CMRUSER
```

CMRUSER authorizations

Display of events to GUI is controlled by Workgroup

UserAuthority::CMRUSER::****:EXCMD_MANAGER_EVENT=1

UserAuthority::CMRUSER::****:EXCMD_ALTER_EVENT=1

UserAuthority::CMRUSER::****:EXCMD_STATE_EVENT=1

UserAuthority::CMRUSER::****:MQCMD_Q_MGR_EVENT=1

UserAuthority::CMRUSER::****:MQCMD_PERFM_EVENT=1

UserAuthority::CMRUSER::****:MQCMD_CHANNEL_EVENT=1

Generic commands that do not use node name.

UserAuthority::CMRUSER::****:EXCMD_OPEN_EVENT=1

UserAuthority::CMRUSER::****:EXCMD_INQUIRE_ACTIVE_MANAGER=1

UserAuthority::CMRUSER::****:EXCMD_CLOSE_EVENT=1

UserAuthority::CMRUSER::****:EXCMD_MQ_OPEN=1

General inquiry without node name

UserAuthority::CMRUSER::****:EXCMD_INQUIRE_*=1

UserAuthority::CMRUSER::****:MQCMD_INQUIRE_*=1

Various authorizations

UserAuthority::CMRUSER::****:EXCMD_MANAGE_MQNODE=1

UserAuthority::CMRUSER::****:MQCMD_PING_CHANNEL=1

UserAuthority::CMRUSER::****:MQCMD_PING_Q_MGR=1

UserAuthority::CMRUSER::****\CMRNODE***:MQCMD_CREATE_*=1

UserAuthority::CMRUSER::******:MQCMD_CREATE_*=0

MMF authorizations

UserAuthority::CMRUSER::****\CMRQ***:EXCMD_MG_BROWSE=1

UserAuthority::CMRUSER::******:EXCMD_MG_BROWSE=0

UserAuthority::CMRUSER::****\CMRQ***:EXCMD_MG_FIND=1

UserAuthority::CMRUSER::******:EXCMD_MG_FIND=0

UserAuthority::CMRUSER::****\CMRQ***:EXCMD_MG_NEW=1

UserAuthority::CMRUSER::******:EXCMD_MG_NEW=0

UserAuthority::CMRUSER::******:EXCMD_MG_DELETE=0

UserAuthority::CMRUSER::******:EXCMD_MG_COPY=0

UserAuthority::CMRUSER::******:EXCMD_MG_MOVE=0

UserAuthority::CMRUSER::******:EXCMD_MG_MODIFY=0

Most restrictive

UserAuthority::CMRUSER::*=0

Deny all other cmds

All other users (no access to MQ objects)

EffectiveUserName::*=NOACUSER

NOACUSER authorizations

UserAuthority::NOACUSER::*=0

Deny all cmds to invalid user

8.6.2 Use Case Example 2

```
#SYNTAX_V2
```

```
# Required format version ID
```

```
# This file is the security file for the dev1 iteration of the M6-WMQ.#
```

```
#####
```

```
# Nastel Administrators
```

```
# Created : 09.22.07 by XYZ Company
```

```
#####
```

```
TRUST::*:*:*=1
```

```
Audit::*:*=denied
```

```
# Mapping the Admin names to be equivalent to the USR_ADMIN_1 ID below:
```

```
EffectiveUserName::USER1::*=usr_admin_1           # USER1 - Grant all access to all objects
```

```
EffectiveUserName::LC44NAS::*=usr_admin_1         # USER1 UserID
```

```
EffectiveUserName::MQM::*=usr_admin_1             # MQM- Grant all access to all objects
```

```
EffectiveUserName::mqm::*=usr_admin_1
```

```
EffectiveUserName::SYSTEM::*=usr_admin_1          # SYSTEM- Grant access to all
objects
```

```
EffectiveUserName::ADMIN::*=usr_admin_1           # AutoPilot M6 ID
```

```
EffectiveUserName::MUSR_MQADMIN::*=usr_admin_1    # Win WebSphere/MQ UserID
```

```
EffectiveUserName::Admin::*=usr_admin_1          # AutoPilot M6 ID
```

```
UserAuthority::usr_admin_1::*=1                  # Grant all access to all objects
```

```
#####
### Common commands for all Workgroup server's and all users #####
#####

UserAuthority::*:*|*|*|*|*::EXCMD_OPEN_EVENT=1
UserAuthority::*:*|*|*|*|*::EXCMD_INQUIRE_ACTIVE_MANAGER=1
UserAuthority::*:*|*|*|*|*|EXOT_MANAGER::EXCMD_INQUIRE_MQNODE_NAMES=1
UserAuthority::*:*|*|*|*|*|EXOT_MANAGER::EXCMD_INQUIRE_MANAGER=1
UserAuthority::*:*|*|*|*|*|*::EXCMD_INQUIRE_*=1
UserAuthority::*:*|*|*|*|*|*::MQCMD_INQUIRE_*=1

# Allow all WMQ Agent's to register with Workgroup server
UserAuthority::*:*|*|*|*|*|*::EXCMD_REGISTER_MQNODE=1

# Display of events to GUI is controlled by Workgroup server
UserAuthority::*:*|*|*|*|*|*::EXCMD_MANAGER_EVENT=1
UserAuthority::*:*|*|*|*|*|*::EXCMD_ALTER_EVENT=1
UserAuthority::*:*|*|*|*|*|*::EXCMD_STATE_EVENT=1
UserAuthority::*:*|*|*|*|*|*::MQCMD_Q_MGR_EVENT=1
UserAuthority::*:*|*|*|*|*|*::MQCMD_PERFM_EVENT=1
UserAuthority::*:*|*|*|*|*|*::MQCMD_CHANNEL_EVENT=1

# Any user needs the next three basic permissions to connect to a WMQ Agent.

UserAuthority::*:*|*|*|*|*|*::EXCMD_OPEN_EVENT=1           # Opens WMQ Agent event queue
UserAuthority::*:*|*|*|*|*|*::EXCMD_CLOSE_EVENT=1         # Closes WMQ Agent event queue
UserAuthority::*:*|*|*|*|*|*::EXCMD_MQ_OPEN=1             # Connect to a qmgr

#####
# Mapping the Admin names to be equivalent to the USR_ADMIN_4 ID below:

EffectiveUserName::ABC123::*=usr_admin_4                   # Limited access to all objects
EffectiveUserName::DEF345::*=usr_admin_4                   # Limited access to all objects
EffectiveUserName::GHI678::*=usr_admin_4                   # Limited access to all objects
EffectiveUserName::JKL123::*=usr_admin_4                   # Limited access to all objects
EffectiveUserName::MNO456::*=usr_admin_4                   # Limited access to all objects
EffectiveUserName::PQR789::*=usr_admin_4                   # Limited access to all objects
```


8.7 Firewall Administration



IMPORTANT!

Firewall administrators must allow incoming TCP connection requests on port 5000. UDP port can remain closed; it is used only for automatically discovering new agents on the network.

For server to agent communications, the following is done automatically:

1. The WMQ Agent is started and creates TCP and UDP Listeners on the specified port. The default port is 5010 and is configurable.
2. WMQ Agent reads local server configuration and sends UDP registration to Workgroup server (server) – default is 4010 and is configurable.
3. Server responds to registration and sends a request to establish TCP connection to port 5010.
4. WMQ Agent responds and starts new threads for the server, one per QMGR. TCP/IP service allocates a new port number for each thread – this is transparent to the user and handled automatically by the firewall.
5. If the connection is broken the server tries to re-establish connection. If not successful, it sends a TCP request every 10 min (default can be changed and is set for every node) to re-establish connection.



NOTE:

If the agent cannot successfully establish a connection on startup, the node can be manually added to the configuration and server will send a TCP request, agent will reply with registration response, and steps 3 and 4 are repeated.

6. The server also sends UDP broadcasts on a pre-configured port (5010) to request registrations from any new agents every 1440 minutes; both port and time are configurable.

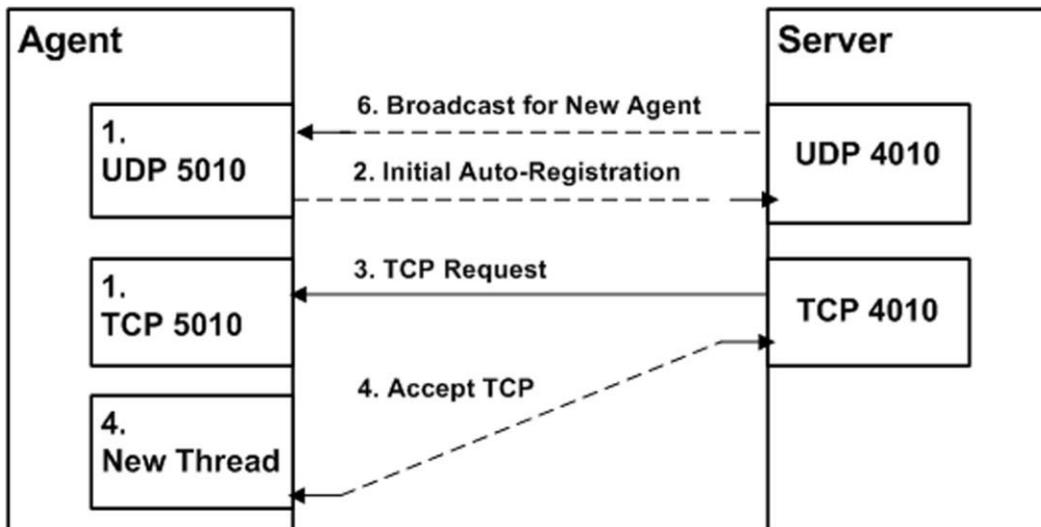


Figure 8-2. Server to Agent Communication

8.8 WebSphere MQ Access Requirements

WebSphere MQ provides various security mechanisms through which access to local WebSphere MQ objects can be restricted. M6-WMQ's security methods may also be used to control access to WebSphere MQ objects. The following sections describe how to edit M6-WMQ security profiles to restrict access to WebSphere MQ objects.

There are three areas of WebSphere MQ access:

- Access to object properties
- Access to commands
- Access to messages

M6-WMQ accesses WebSphere MQ objects in one of two contexts:

- **External user context:** On behalf of an external user who has submitted a command (from M6-WMQ Explorer, for example)
- **Local context:** As a local application, M6-WMQ must open, put and get to certain queues.

8.9 Local Context Access to WebSphere MQ Objects

M6-WMQ processes require full access to certain WebSphere MQ objects. Full access means access to commands, such as Alter, Browse, Put, Get, Delete, Define, etc. With respect to WebSphere MQ Object Authority Manager (OAM) `setmqaut` commands, it means "ALL".

M6-WMQ requires full access to the following local queues (the M6-WMQ applications which access the named object are given in parenthesis):

<code>NASTEL.EVENT.QUEUE</code>	(WMQ Agent)
<code>NASTEL.MMF.AUDIT.QUEUE</code>	(WMQ Agent)
<code>NASTEL.MMF.ADMIN.COMMAND.QUEUE</code>	(WMQ Agent, MS)
<code>NASTEL.PUBSUB.EVENT.QUEUE</code>	(WMQ Agent, PS)
<code>SYSTEM.ADMIN.COMMAND.QUEUE</code>	(WMQ Agent)
<code>SYSTEM.COMMAND.INPUT</code>	(WMQ Agent)

Table 8-2. WebSphere MQ PCF Codes

Command Category	Command Code Name	Value	Command Code Name	Value
QMGRs	MQCMD_CHANGE_Q_MGR	1	MQCMD_INQUIRE_Q_MGR	2
	MQCMD_REFRESH_Q_MGR	16	MQCMD_RESET_Q_MGR	92
	MQCMD_PING_Q_MGR	40	MQCMD_Q_MGR_EVENT	44
	MQCMD_RESUME_Q_MGR	132	MQCMD_START_Q_MGR	139
	MQCMD_STOP_Q_MGR	144	MQCMD_SUSPEND_Q_MGR	146
	MQCMD_INQUIRE_Q_MGR_STATUS	161	MQCMD_INQUIRE_CMD_SERVER	117
	MQCMD_INQUIRE_QSG	119	MQCMD_START_CMD_SERVER	138
	MQCMD_STOP_CMD_SERVER	143		
Processes	MQCMD_CHANGE_PROCESS	3	MQCMD_COPY_PROCESS	4
	MQCMD_CREATE_PROCESS	5	MQCMD_DELETE_PROCESS	6
	MQCMD_INQUIRE_PROCESS	7	MQCMD_INQUIRE_PROCESS_NAMES	19
Queues	MQCMD_CHANGE_Q	8	MQCMD_CLEAR_Q	9
	MQCMD_COPY_Q	10	MQCMD_CREATE_Q	11
	MQCMD_DELETE_Q	12	MQCMD_INQUIRE_Q	13
	MQCMD_RESET_Q_STATS	17	MQCMD_INQUIRE_Q_STATUS	41
	MQCMD_INQUIRE_Q_NAMES	18	MQCMD_STATISTICS_Q	165
	MQCMD_ACCOUNTING_Q	168	MQCMD_MOVE_Q	127
	MQCMD_STATISTICS_MQI	164	MQCMD_ACCOUNTING_MQI	167
Channels	MQCMD_INQUIRE_CHANNEL_NAMES	20	MQCMD_CHANGE_CHANNEL	21
	MQCMD_COPY_CHANNEL	22	MQCMD_CREATE_CHANNEL	23
	MQCMD_DELETE_CHANNEL	24	MQCMD_INQUIRE_CHANNEL	25
	MQCMD_PING_CHANNEL	26	MQCMD_RESET_CHANNEL	27
	MQCMD_START_CHANNEL	28	MQCMD_STOP_CHANNEL	29
	MQCMD_RESOLVE_CHANNEL	39	MQCMD_INQUIRE_CHANNEL_STATUS	42
	MQCMD_START_CHANNEL_INIT	30	MQCMD_INQUIRE_CHANNEL_INIT	118
	MQCMD_STOP_CHANNEL_INIT	141	MQCMD_STATISTICS_CHANNEL	166
Topics	MQCMD_CHANGE_TOPIC	170	MQCMD_COPY_TOPIC	171
	MQCMD_CREATE_TOPIC	172	MQCMD_DELETE_TOPIC	173
	MQCMD_INQUIRE_TOPIC	174	MQCMD_INQUIRE_TOPIC_NAMES	175
	MQCMD_INQUIRE_TOPIC_STATUS	183	MQCMD_CLEAR_TOPIC_STRING	184
Subscriptions	MQCMD_INQUIRE_SUBSCRIPTION	176	MQCMD_CREATE_SUBSCRIPTION	177
	MQCMD_CHANGE_SUBSCRIPTION	178	MQCMD_DELETE_SUBSCRIPTION	179
	MQCMD_COPY_SUBSCRIPTION	181	MQCMD_INQUIRE_SUB_STATUS	182
	MQCMD_INQUIRE_PUBSUB_STATUS	185		
Namelists	MQCMD_CHANGE_NAMELIST	32	MQCMD_COPY_NAMELIST	33
	MQCMD_CREATE_NAMELIST	34	MQCMD_DELETE_NAMELIST	35
	MQCMD_INQUIRE_NAMELIST	36	MQCMD_INQUIRE_NAMELIST_NAMES	37
Events	MQCMD_CONFIG_EVENT	43	MQCMD_PERFM_EVENT	45
	MQCMD_CHANNEL_EVENT	46	MQCMD_COMMAND_EVENT	99

Table 8-2. WebSphere MQ PCF Codes

Command Category	Command Code Name	Value	Command Code Name	Value
Publish/ Subscribe	MQCMD_DELETE_PUBLICATION	60	MQCMD_DEREGISTER_PUBLISHER	61
	MQCMD_DEREGISTER_SUBSCRIBER	62	MQCMD_PUBLISH	63
	MQCMD_REGISTER_PUBLISHER	64	MQCMD_REGISTER_SUBSCRIBER	65
Clusters	MQCMD_INQUIRE_CLUSTER_Q_MGR	70	MQCMD_RESUME_Q_MGR_CLUSTER	71
	MQCMD_SUSPEND_Q_MGR_CLUSTER	72	MQCMD_REFRESH_CLUSTER	73
	MQCMD_RESET_CLUSTER	74		
Services	MQCMD_CHANGE_SERVICE	149	MQCMD_COPY_SERVICE	150
	MQCMD_CREATE_SERVICE	151	MQCMD_DELETE_SERVICE	152
	MQCMD_INQUIRE_SERVICE	153	MQCMD_START_SERVICE	155
	MQCMD_INQUIRE_SERVICE_STATUS	154		
	MQCMD_STOP_SERVICE	156		
Channel Listeners	MQCMD_CHANGE_LISTENER	93	MQCMD_COPY_LISTENER	94
	MQCMD_CREATE_LISTENER	95	MQCMD_DELETE_LISTENER	96
	MQCMD_INQUIRE_LISTENER	97	MQCMD_START_CHANNEL_LISTENER	31
	MQCMD_INQUIRE_LISTENER_STATUS	98	MQCMD_STOP_CHANNEL_LISTENER	142
Security/ Authentication	MQCMD_REFRESH_SECURITY	78	MQCMD_CHANGE_AUTH_INFO	79
	MQCMD_COPY_AUTH_INFO	80	MQCMD_CREATE_AUTH_INFO	81
	MQCMD_DELETE_AUTH_INFO	82	MQCMD_INQUIRE_AUTH_INFO	83
	MQCMD_INQUIRE_AUTH_INFO_NAMES	84	MQCMD_CHANGE_SECURITY	100
	MQCMD_INQUIRE_SECURITY	121	MQCMD_REVERIFY_SECURITY	133
Miscellaneous	MQCMD_ESCAPE	38	MQCMD_REQUEST_UPDATE	66
	MQCMD_BROKER_INTERNAL	67	MQCMD_ACTIVITY_MSG	69
	MQCMD_TRACE_ROUTE	75	MQCMD_INQUIRE_CONNECTION	85
	MQCMD_STOP_CONNECTION	86	MQCMD_INQUIRE_AUTH_RECS	87
	MQCMD_INQUIRE_ENTITY_AUTH	88	MQCMD_DELETE_AUTH_REC	89
	MQCMD_SET_AUTH_REC	90	MQCMD_LOGGER_EVENT	91
	MQCMD_CHANGE_TRACE	103	MQCMD_ARCHIVE_LOG	104
	MQCMD_INQUIRE_ARCHIVE	114	MQCMD_INQUIRE_LOG	120
	MQCMD_INQUIRE_SYSTEM	123	MQCMD_INQUIRE_THREAD	124
	MQCMD_INQUIRE_TRACE	125	MQCMD_INQUIRE_USAGE	126
	MQCMD_RECOVER_BSDS	128	MQCMD_RESET_TPIPE	130
	MQCMD_RESOLVE_INDOUBT	131	MQCMD_SET_ARCHIVE	134
	MQCMD_SET_LOG	136	MQCMD_SET_SYSTEM	137
	MQCMD_START_TRACE	140	MQCMD_STOP_TRACE	145
	MQCMD_CREATE_LOG	162	MQCMD_INQUIRE_AUTH_SERVICE	169
CF Structs	MQCMD_CHANGE_CF_STRUCT	101	MQCMD_BACKUP_CF_STRUCT	105
	MQCMD_CREATE_CF_STRUCT	108	MQCMD_COPY_CF_STRUCT	110
	MQCMD_DELETE_CF_STRUCT	112	MQCMD_INQUIRE_CF_STRUCT	115
	MQCMD_INQUIRE_CF_STRUCT_STATUS	116	MQCMD_RECOVER_CF_STRUCT	129
	MQCMD_INQUIRE_CF_STRUCT_NAMES	147		

Table 8-2. WebSphere MQ PCF Codes

Command Category	Command Code Name	Value	Command Code Name	Value
Storage Classes	MQCMD_CHANGE_STG_CLASS	102	MQCMD_INQUIRE_STG_CLASS_NAMES	148
	MQCMD_CREATE_STG_CLASS	109	MQCMD_COPY_STG_CLASS	111
	MQCMD_DELETE_STG_CLASS	113	MQCMD_INQUIRE_STG_CLASS	122
Buffer Pools	MQCMD_CREATE_BUFFER_POOL	106	MQCMD_DELETE_BUFFER_POOL	157
	MQCMD_CHANGE_BUFFER_POOL	159		
Page Sets	MQCMD_CREATE_PAGE_SET	107	MQCMD_DELETE_PAGE_SET	158
	MQCMD_CHANGE_PAGE_SET	160		

Table 8-3. M6-WMQ Extended PCF Codes

Command Category	Command Name	Value	Command Name	Value
QMGRs	EXCMD_START_Q_MGR	10003	EXCMD_STOP_Q_MGR	10004
	EXCMD_INQUIRE_Q_MGR_NAMES	10007	EXCMD_CREATE_Q_MGR	10012
	EXCMD_CHANGE_Q_MGR_STAT	10011	EXCMD_TEST_Q_MGR	10999
	EXCMD_DELETE_Q_MGR	10013		
Processes	EXCMD_START_PROCESS	10998		
Queues	EXCMD_INQUIRE_Q_RUNTIME	10017	EXCMD_MOVE_QLOCAL	10611
Events	EXCMD_INQUIRE_Q_MGR_EVENT	10160	EXCMD_DELETE_Q_MGR_EVENT	10161
	EXCMD_INQUIRE_CHANNEL_EVENT	10170	EXCMD_DELETE_CHANNEL_EVENT	10171
	EXCMD_INQUIRE_MANAGER_EVENT	10180	EXCMD_DELETE_MANAGER_EVENT	10181
	EXCMD_INQUIRE_PERFM_EVENT	10190	EXCMD_DELETE_PERFM_EVENT	10191
	EXCMD_INQUIRE_ALTER_EVENT	10200	EXCMD_DELETE_ALTER_EVENT	10201
	EXCMD_INQUIRE_CONFIG_EVENT	10202	EXCMD_DELETE_CONFIG_EVENT	10203
	EXCMD_OPEN_EVENT	10005	EXCMD_CLOSE_EVENT	10006
	EXCMD_ALTER_EVENT	8001	EXCMD_MANAGER_EVENT	8002
	EXCMD_STATE_EVENT	8003		
Clusters	EXCMD_INQUIRE_CLUSQMGR_NAMES	10020		
Security/ Authentication	EXCMD_LOGIN_CHALLENGE	12001	EXCMD_LOGIN_AUTHENTICATE	12002
	EXCMD_LOGIN_VERIFY	12003	EXCMD_CHANGE_PASSWORD	12004
	EXCMD_DISPLAY_SECURITY	10681	EXCMD_CHANGE_SECURITY	10682
	EXCMD_VERIFY_SECURITY	10683	EXCMD_REFRESH_SECURITY	10684
	EXCMD_RELOAD_PERMITS	10208	EXCMD_INQUIRE_SEC_REALMS	12005
Workgroup	EXCMD_INQUIRE_MANAGER_NAMES	10100	EXCMD_INQUIRE_MANAGER	10101
	EXCMD_CHANGE_MANAGER	10102	EXCMD_DELETE_MANAGER	10103
	EXCMD_CREATE_MANAGER	10104	EXCMD_START_MANAGER	10105
	EXCMD_STOP_MANAGER	10106	EXCMD_CHANGE_MANAGER_STAT	10107
	EXCMD_INQUIRE_ACTIVE_MANAGER	10108	EXCMD_SET_ACTIVE_MANAGER	10109
	EXCMD_SHUTDOWN_MANAGER	10110	EXCMD_DB_BACKUP	11100
	EXCMD_DB_RESTORE	11101	EXCMD_INQUIRE_LICENSE	11001
	EXCMD_SQLDB_CONNECT	11102		
MQ Nodes	EXCMD_INQUIRE_MQNODE_NAMES	10120	EXCMD_INQUIRE_MQNODE	10121
	EXCMD_CHANGE_MQNODE	10122	EXCMD_DELETE_MQNODE	10123
	EXCMD_CREATE_MQNODE	10124	EXCMD_START_MQNODE	10125
	EXCMD_STOP_MQNODE	10126	EXCMD_CHANGE_MQNODE_STAT	10127
	EXCMD_REGISTER_MQNODE	10128	EXCMD_MANAGE_MQNODE	10129
	EXCMD_UNMANAGE_MQNODE	10130	EXCMD_HEART_BEAT	10015
MCAs	EXCMD_INQUIRE_MCA_NAMES*	10140	EXCMD_INQUIRE_MCA*	10141
	EXCMD_CHANGE_MCA*	10142	EXCMD_START_MCA*	10001
	EXCMD_STOP_MCA*	10002		
TA Configuration	EXCMD_TA_CONFIG	10231		

Table 8-3. M6-WMQ Extended PCF Codes

Command Category	Command Name	Value	Command Name	Value
Message Management	EXCMD_MG_BROWSE	10300	EXCMD_MG_NEW	10301
	EXCMD_MG_MOVE	10302	EXCMD_MG_DELETE	10303
	EXCMD_MG_COPY	10304	EXCMD_MG_FIND	10305
	EXCMD_MG_MODIFY	10306		
Listener	EXCMD_INQUIRE_LISTENER_NAMES	10021	EXCMD_STOP_CHANNEL_LISTENER	10601
Channel	EXCMD_STOP_CHANNEL_INIT	10602	EXCMD_INQUIRE_CHANNEL_INIT	10603
Subscription	EXCMD_INQUIRE_SUBSCRIPTION_NAME	10024		
Logs	EXCMD_ARCHIVE_LOG	10631	EXCMD_DISPLAY_LOG	10632
	EXCMD_SET_LOG	10633	EXCMD_INQUIRE_LOG_NAMES	10207
Archive	EXCMD_DISPLAY_ARCHIVE	10641	EXCMD_SET_ARCHIVE	10642
CFSTRUCT	EXCMD_DISPLAY_CF_STRUC	10691	EXCMD_CREATE_CF_STRUC	10692
	EXCMD_COPY_CF_STRUC	10693	EXCMD_CHANGE_CF_STRUC	10694
	EXCMD_DELETE_CF_STRUC	10695	EXCMD_BACKUP_CF_STRUC	10696
	EXCMD_RECOVER_CF_STRUC	10697	EXCMD_DISPLAY_CF_STATUS	10698
Accounting	EXCMD_INQUIRE_ACCT_MQI	10400	EXCMD_DELETE_ACCT_MQI	10401
	EXCMD_INQUIRE_ACCT_Q	10402	EXCMD_DELETE_ACCT_Q	10403
Statistics	EXCMD_INQUIRE_STAT_MQI	10410	EXCMD_DELETE_STAT_MQI	10411
	EXCMD_INQUIRE_STAT_Q	10412	EXCMD_DELETE_STAT_Q	10413
	EXCMD_INQUIRE_STAT_CHANNEL	10414	EXCMD_DELETE_STAT_CHANNEL	10415
Miscellaneous	EXCMD_MQ_OPEN	10008	EXCMD_MQ_CLOSE	10009
	EXCMD_STOP_PCFEM	10010	EXCMD_MQSC_SCRIPT	10014
	EXCMD_DELETE_SESSION	10019	EXCMD_LOCATE_OBJECT	10016
	EXCMD_INQUIRE_SESSION	10018	EXCMD_INQUIRE_SERVICE_NAMES	10022
	EXCMD_INQUIRE_TEXT_FILE	10205	EXCMD_CREATE_TEXT_FILE	10206
	EXCMD_MQ_DISCOVER	11000	EXCMD_EXECUTE_SHELL_CMD	11002
	EXCMD_DISPLAY_SYSTEM	10671	EXCMD_SET_SYSTEM	10672
	EXCMD_DISPLAY_GROUP	10901		

**NOTE:**

* Not applicable. Command is obsolete.

This Page Intentionally Left Blank

Chapter 9: Using M6-WMQ API

You can extend and enhance M6-WMQ functionality using the M6-WMQ Application Programming Interface (API).

- Connectivity services (for connection to TCP/IP and WebSphere MQ)
- Data manipulation services
- Common API services.

The relation between these groups is shown in Figure 9-1. Note that outside services can interact with the full set of API services, or with a smaller subset of them.

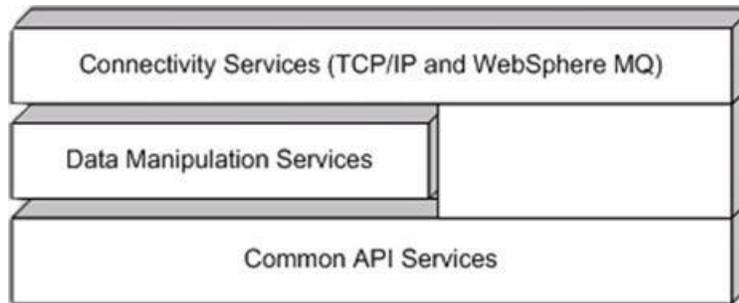


Figure 9-1. Relation between M6-WMQ and API Services

9.1 M6-WMQ API Services

Table 9-1. M6-WMQ API, Services Available

Service Group	Service	Functionality
Data Manipulation Services	Message API (MsgAPI)	Lets applications organize data into exchangeable messages for TCP/IP or WebSphere MQ.
	PSM Message API (PsmAPI)	Uses the message API to create and send Programmable System Management (PSM) messages.
Connectivity Services	Node Connectivity API (NconnAPI)	Allows inter-process communication among remote nodes. Uses TCP/IP stream-oriented sockets for transport.
	TCP/IP Connectivity API (SockAPI)	Used by the node connectivity API as an interface to the TCP/IP stack. Shields applications from low-level TCP/IP details.
	WebSphere MQ Connectivity API (MQConnAPI)	Allows access to WebSphere MQ resources, such as queue managers and queues. Applications can manage WebSphere MQ by using MQConnAPI in conjunction with PCFAPI.
Common API Services	Data Conversion API (DcAPI)	Allows data conversion from one format to another. For example, performs byte swapping when dealing with two different binary formats.
	Memory Management API (MemAPI)	Handles memory allocation and management. Allows applications to track memory usage, leaks, and total size of allocated memory (including the memory owner).
	Assertion and Debugging API (DbgAPI)	Lets developers track bugs by asserting Boolean conditions in their code. Provides several ways of handling assertion failures.



NOTE:

Not all of M6-WMQ's API services are independent. The connectivity services, for example, use the common services to manage memory and data conversion.

9.1.1 Coordinating Data Formats

Because there are so many binary data formats, establish conversion routines whenever using distributed applications across a network.

There are three ways to coordinate the use of data formats:

- Use one format for all applications. Configure each application to receive data in the agreed format. The disadvantage of this approach is that encoding and decoding must take place on every platform, even when the sending and receiving formats are the same.
- Have each application transmit data in its native format. In this arrangement, the receiving application is responsible for converting messages to its own native format. This arrangement is sometimes called receiver makes it right. The disadvantage of this approach is that the binary encoding and character set must be sent in the header of each message.
- Have each application transmit data in the format of the receiving application. In this arrangement, the sending application attaches a header that tells the receiver about its own binary encoding and character set. The advantage of this approach is that the receiver does not have to convert data if the sender's data format is the same.

9.1.1.1 Handling Events

The M6-WMQ API can recognize and react to specific system events. Using the event-handling functions, a developer can develop event-driven applications that avoid complex loops and event-handling logic.

When it recognizes events, the API can communicate these events to:

- **MQ Nodes:** The API can select a set of nodes that hold the data for an event.
- **Sockets:** The API can select a socket that holds the data for an event.
- **Queue Managers:** The API can retrieve queue manager events from the event queue.

Applications that require multiple event-handling loops can do either of the following:

- Periodically check for events on specific objects, using a time-out technique. For example, an application can check for MQManager events and WebSphere MQ node events every five seconds.

	NOTE:	Although checking for events in this manner does work, it can cause event loops to become overly complex. As a result, event handling may be slowed.
---	--------------	--

- Allocate a separate thread to handle each object type. For example, an application can allocate two event threads: one to handle manager events, the other to handle node events. The advantage of this approach is that each thread's event-loop becomes more specialized and less complex.

9.1.1.2 Memory Management

If an application uses dynamic memory allocation, memory leaks may occur. The API's memory management functions let an application monitor its own memory consumption and clean up unallocated memory blocks. This feature is especially useful during development and debugging, when hidden memory leaks can crash an application and possibly the whole system.

The API can also pinpoint the module where a memory block is allocated. The block is specified in one of the parameters of the `NewMemory` function. When memory-management problems occur, the problem module can be easily identified.

A memory allocation request actually returns two allocations:

- One to satisfy the caller's request
- Another to allocate an `InfoBlock` that holds information about each allocation.

Afterward, the `InfoBlock` structure is inserted into a linked list of all allocated blocks. An application can print the list using the function `DumpMemoryLog`.

	NOTE:	The memory management feature is intended only for the debugging and testing phase. The debugging libraries contain extra code used to manage the memory blocks. For your release version, link to the release libraries, instead of the debug libraries.
---	--------------	---

9.1.1.3 Error Handling

Most of the functions in M6-WMQ are Boolean, meaning that they return true when they succeed, false when they fail. Functions that require more elaborate error handling return special codes, as in this example:

```
IBOOL bCode; ILONG lReason;
```

```
....
```

```
bCode = bFunc(param1, param2, &lReason);
if (!bCode)
{
    /* check the value of lReason */
    switch(lReason)
    {
```

Certain structure-allocation functions (which usually start with `Create<Object>`) return the pointer to the corresponding structure. In these functions, a null value is returned when an error occurs.

There is one other type of error-detection. Some functions that use `Open`, `Close`, `Read` and `Write` verbs return a Boolean value, but do not provide a matching reason code to explain the error. For these functions, each structure maintains a special field. The field contains the error code for the last call to that object.

9.2 Developing Management Applications

To develop management applications in the M6-WMQ API:

- Choose the desired API service
- Use the correct header files for implementing that service
- Use the functions in the API service
- Link with the library that implements the API service

Using the M6-WMQ API does not bind your application to M6-WMQ run-time components such as the Workgroup server. Rather, the API services let applications access resources such as M6-WMQ's resources, such as TCP/IP, WebSphere MQ, PCF messages, and data conversion.

The role of the M6-WMQ API is illustrated in the Figure below:

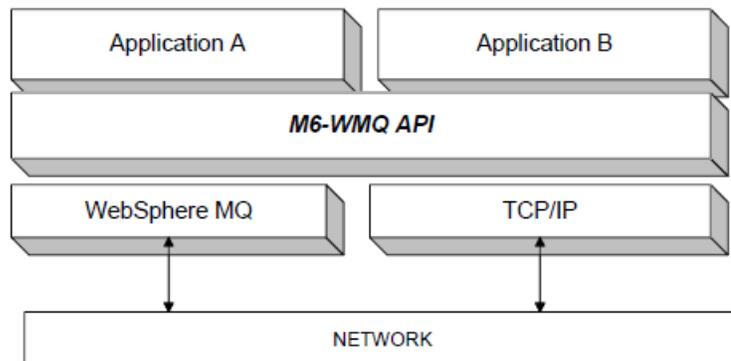


Figure 9-2. Role of the M6-WMQ API

9.2.1 Building Client/Server Applications

The M6-WMQ API is well suited for building client/server applications over TCP/IP networks. The API introduces an `MQNode` object, which represents the remote node.

Any application that wants to establish contact with the managed node creates an instance of the WebSphere MQ node structure, using the function `CreateMQNode`. The application makes calls such as `Open`, `Close`, `Read` and `Write` to communicate with the node.

	NOTE:	The <code>MQNode</code> object is built to exchange PSM messages only. The PSM format creates messages that can be easily recognized and converted to the format of the native machine.
---	--------------	---

9.2.1 WebSphere MQ-based Applications

The M6-WMQ API also provides direct access to WebSphere MQ components. For example, when an MQManager object is created in the API, the command queue for that queue manager opens automatically. Any PCF commands issued to the API object are sent to the queue manager's command queue.

9.2.1.1 Management Applications for WebSphere MQ

Using the services PsmAPI and NConnAPI, a developer can write applications that connect directly to M6-WMQ components, including Workgroup servers and WMQ Agents. This functionality is particularly useful for developers who want to create front-end management applications for WebSphere MQ.

The management applications written may, if preferred, communicate with the workgroup server only. Communicating with the workgroup server only provides access to all M6-WMQ functionality. However, in some instances the management application will need to connect directly to WebSphere MQ nodes.

For example, in some cases the workgroup server may be unavailable due to a system crash or communication problem. In these cases, it is preferred that the management application communicates directly with WebSphere MQ, as shown in the Figure below.

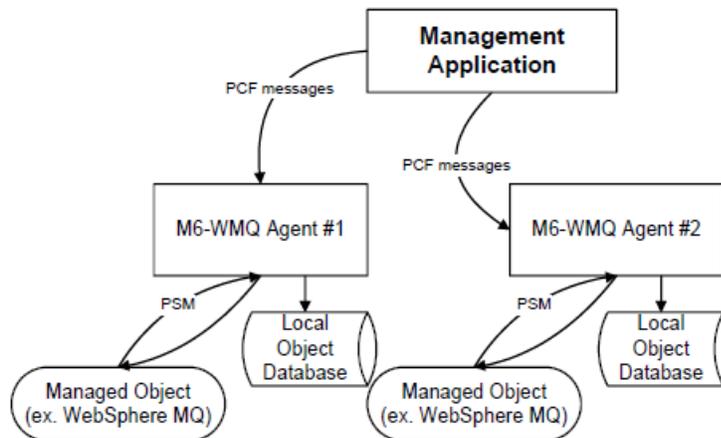


Figure 9-3. Managing WebSphere MQ Directly Using Application Developed in M6-WMQ API

The M6-WMQ API enables communication between the management application, the workgroup server, the WMQ Agents and WebSphere MQ. In Figure 10-3, the arrangement includes:

- **WMQ Agent:** running on each WebSphere MQ node and interacting with queue managers. The WMQ Agent also acts as a server that can handle management commands from the management application.
- **The Management Application:** Interacting with the WMQ Agent to collect information about WebSphere MQ objects and perform operations.



NOTE:

For more information on the M6-WMQ API, see the *M6-WMQ API Reference Guide* included in the Documentation Library folder on your M6-WMQ installation media.

This Page Intentionally Left Blank

Chapter 10: Integrating the M6-WMQ Ping Utility

M6-WMQ ping utility (**NSQPING** can be integrated into AutoPilot M6 to provide detailed message performance metrics).

The following diagram depicts the integration path:

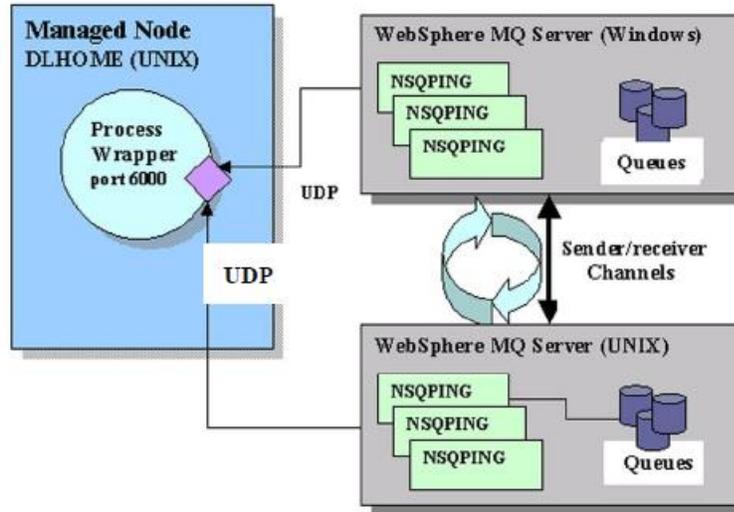


Figure 10-1. Queue Ping Utility Integration Path (NSQPING)

Integrate as follows:

1. **AutoPilot M6 Configuration:** deploy an instance of AutoPilot M6 process wrapper on a managed node close or same machine as the source of the metrics. See *AutoPilot M6 Configuration* section below.
2. **Configuring Process Wrapper Options:** configure process wrapper to accept WebSphere MQ performance metrics from M6-WMQ **NSQPING** utility. See *Configuring Process Wrapper* section below.

	NOTE:	Due to unreliable nature of UDP, it is highly recommended that process wrapper be deployed on the same machine as NSQPING . This also means that managed node must be running on the same machine as NSQPING .
---	--------------	--

Once AutoPilot M6 process wrapper is deployed and configured:

1. **Measuring Message Performance:** run **NSQPING** as a daemon to collect message performance metrics. See *Measuring Messaging Flow Performance* section below.
2. **Monitor Message Metrics:** create business view to monitor collected metrics. See *Monitor Message Metrics* section below.

10.1 AutoPilot M6 Configuration

The following steps are performed to accept WebSphere MQ message performance metrics from M6-WMQ.

1. Deploy process wrapper: Process wrapper is a generic AutoPilot M6 monitor capable of accepting facts from other applications via UDP or WebSphere MQ. It can also start processes, restart processes and record published information into a JDBC database.
2. Configure process wrapper: Setting options required to accept information from M6-WMQ ping utility. This step is required to monitor WebSphere MQ message performance metrics.

10.1.1 Deploying Process Wrapper

To deploy and configure an instance of a process wrapper:

1. Right click on the desired managed node.
2. Select **Deploy Expert > Wrappers > Process Wrapper**
3. Configure Process Wrapper for context, name and to accept fact on the desired UDP port (6000). (See *Configuring Process Wrapper* section below.)
4. Click **Deploy** to deploy the expert on the managed node.

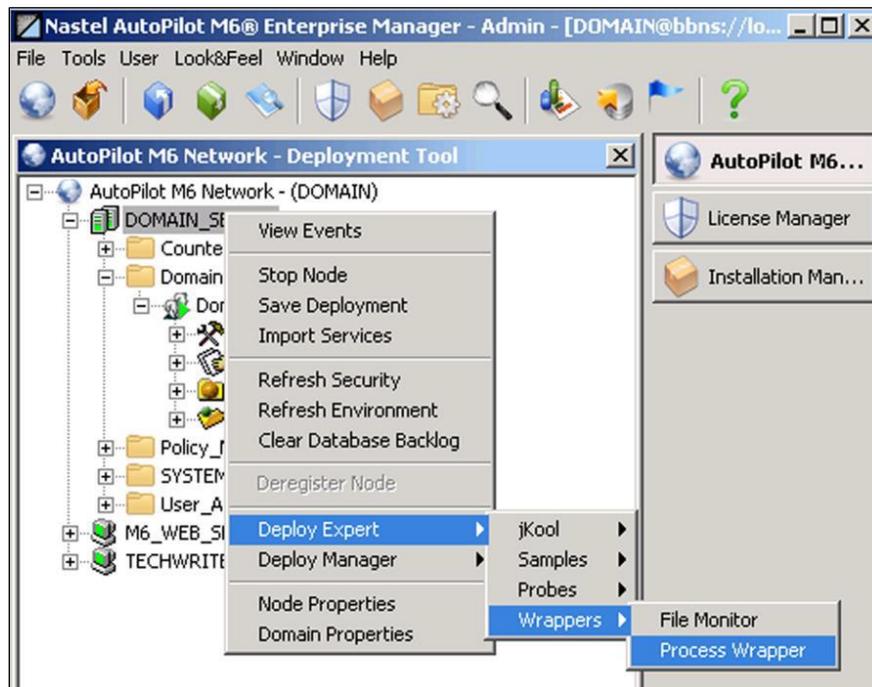


Figure 10-2. Deploying Process Wrappers

Once selected the process wrapper configuration properties will be displayed, as shown below.

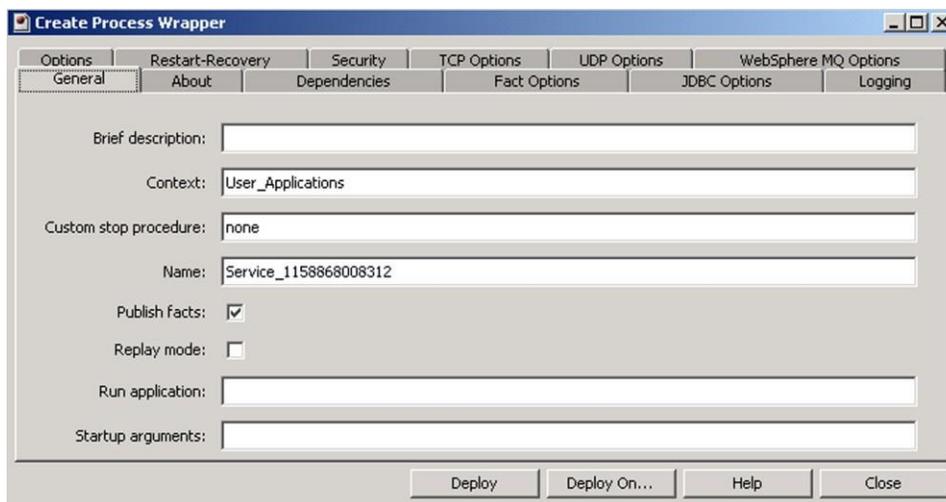


Figure 10-3. Process Wrapper Configuration Properties

10.1.2 Configuring Process Wrapper

The following process wrapper properties must be configured:

1. **General:** At a minimum *Context* and *Name* must be configured.

Table 10-1. Process Wrapper Expert: General

Property	Description
Brief Description	A short, user defined, description of the service
Context	A user define category that will be registered with the domain server. The default is: User_Applications
Custom Stop Procedure	Application or script that gracefully shuts down started applications. The default is: none
Name	Name that uniquely identifies the service in the domain. The default name system assigned with the word service and 12 random digits (example: service_123456789012)
Publish facts	Check to enable/disable. Publishes received facts locally when enabled.
Replay mode	Check to enable/disable. Emulates fact source and replays all received facts.
Run Application	Fully qualified name of application/script/process to run (example: usr/bin/tar)
Startup Arguments	Start-up parameters that are passed to the application (example: cvf mytar/opt/nastel)

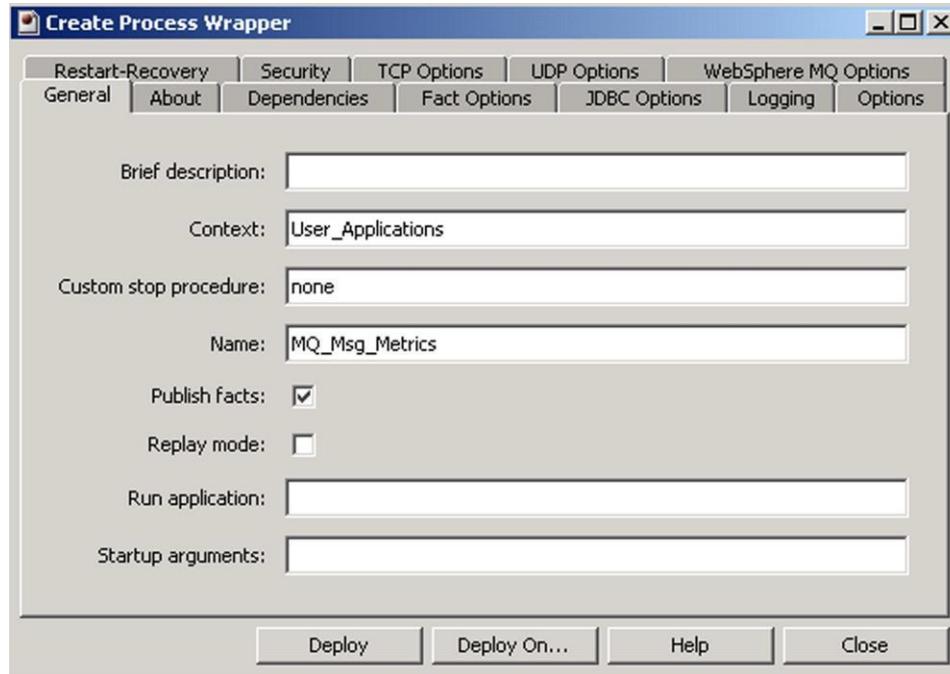


Figure 10-4. Process Wrapper General Configuration Properties

2. **UDP Options:** Both properties on this screen must be configured.

Table 10-2. Process Wrapper Expert: UDP Options	
Property	Description
Accept UDP facts	Enable by clicking in the checkbox.
UDP port	Unique port on which process wrapper will accept the incoming performance metrics. Default is 6000.

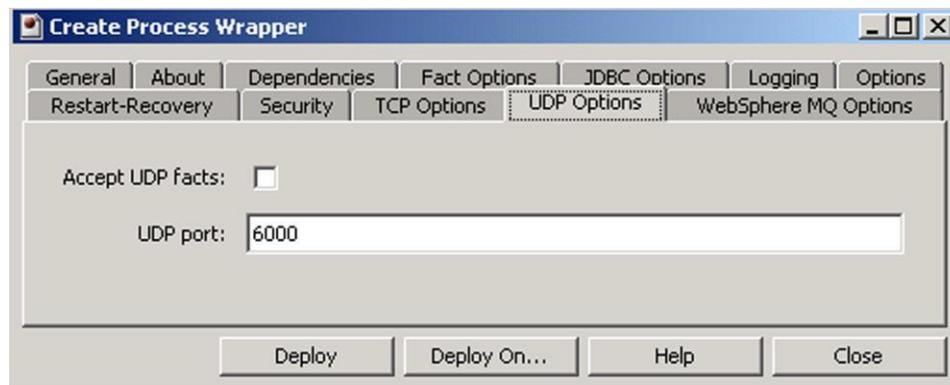


Figure 10-5. Process Wrapper UDP Option Configuration Properties

10.2 Measuring Message Flow Performance

To measure message performance:

- Run **NSQPING** at the command prompt.

	IMPORTANT!	When measuring message performance over remote queues: local and remote queue managers must have pair of sender/receiver channels defined and active that allows local and remote queue manager to exchange messages.
---	-------------------	---

NSQPING requires:

- Target queue manager running
- WebSphere MQ command server or PCF capable application running on the specified queue manager.

	NOTE:	PCF is IBM's Programmable Command Format.
---	--------------	---

NSQPING uses a batch of PCF messages and submits them to the WebSphere MQ command server over a user specified queue. This queue can be configured as local, alias, cluster or remote queue as long as it points eventually to a WebSphere MQ command server or an application capable of handling MQCMD_PING_Q_MGR PCF command.

To publish message performance metrics using **NSQPING**, execute at the command prompt:

```
nsqping
```

```
-mQmgr_name -qQueueName -bBatch_size -hHost -pPort -dSampling_rate
```

- **Example 1:** Measure message performance every ten seconds on queue manager **QM1** over **SYSTEM.ADMIN.COMMAND.QUEUE** and forward the ping test messages to a process wrapper running on **DLHOME** and listening on port **6000**:

```
nsqping
```

```
-mQM1 -qSYSTEM.ADMIN.COMMAND.QUEUE -b100 -hDLHOME -p6000 -d10
```

- **Example 2:** Measure message performance every 20 seconds on queue manager **QM2** over **REMOTE.Q** and forward ping test messages to a process wrapper running on **DLHOME** and listening on port 6060, perform confirm on arrival (**-cco**) and confirm on delivery (**-ccod**) measurements:

```
nsqping
```

```
-mQM2 -qREMOTE.Q -cco -ccod -b100 -hDLHOME -p6060 -d20
```

NSQPING Example

Below is a sample output of **NSQPING** when executed in interactive mode:

```
nsqping -mQM_ramses -qSYSTEM.ADMIN.COMMAND.QUEUE -ccoa -ccod -b10 Pinging
QM_ramses(SYSTEM.ADMIN.COMMAND.QUEUE) using 36 byte 10(msgs) batch
```

Statistics for queue QM_ramses(SYSTEM.ADMIN.COMMAND.QUEUE):

Summary Performance Indicators:

```
MINIMUM_ROUND_TRIP(0.0000
sec/msg)
MAXIMUM_ROUND_TRIP(9.7050
sec/msg)
AVERAGE_ROUND_TRIP(6.4672
sec/msg) MESSAGES_SENT(10)
CONFIRMED_EXPIRIES(0)
CONFIRMED_DELIVERIES(10)
CONFIRMED_ARRIVALS(10)
CONFIRMED_EXCEPTIONS(0)
REPORTS_RECEIVED(20)
RESPONSES_RECEIVED(10)
MESSAGES_RECEIVED(30)
BYTES_SENT(360)
BYTES_RECEIVED(360)
RESPONSE_REQUEST_RATIO(100.000
0%)
```

General Performance Indicators:

```
TOTAL_PUT_TIME(1.5770 sec)
TOTAL_GET_TIME(12.8740 sec)

AVERAGE_PUT_RATE(6.3412 msg/sec [228.28 bytes/sec])
AVERAGE_GET_RATE(2.3303 msg/sec [27.96 bytes/sec])
PUT_GET_RATIO(272.1201% [2.72])
```

Message Performance Indicators:

```
GROSS_ROUND_TRIP_RATE(2.7680 msg/sec [49.82
bytes/sec]) EFFECTIVE_ROUND_TRIP_RATE(1.3840
msg/sec) CONFIRMATION_OVERHEAD(50.0000% [0.50])
AVERAGE_ARRIVAL_RATE(0.0000 msg/sec)
AVERAGE_DELIVERY_RATE(100.0000 msg/sec)
AVERAGE_MSG_LATENCY(0.0000 sec) WITH QDEPTH(0)
AXIMUM_MSG_LATENCY(0.1000 sec) WITH QDEPTH(10)

TEST_COMPLETION_CODE(0)
```

NSQPING Command Line Options

Below is a table of NSQPING options and their usage:

Table 10-3. NSQPING Command Line Options and Usages	
Property	Description
-m<QmgrName>	Specifies the name of the local queue manager.
-q<QueueName>	Queue name of any type.
-b<Batch>	Number of messages in a batch.
-s<Bytes>	Size, in bytes, of generated messages.
-d<Interval>	Interval, in seconds, that ping messages are generated. An interval of 0 (zero) indicates to ping only once.
-w<Wait>	Wait interval, in milliseconds, before timeout.
-e<Expiry>	Message expires after Expiry time in 0.1 seconds.
-u<AltUser>	Issue ping on behalf of another user.
-cco	Generate confirm-on-arrival report.
-ccod	Generate confirm-on-delivery report.
-cexp	Generate confirm-on-expiry report.
-cexc	Generate confirm-on-exception report.
-cdlq	Generate message that will go onto DLQ if undelivered.
-h<Host>	Host name of the target AutoPilot M6 Process Wrapper.
-p<Port>	Port number of the target AutoPilot M6 Process Wrapper.
-v<0 1 2>	Set verbose mode to 0 – summary; 1- reports; 2- all

10.3 Monitor Message Metrics

All message performance metrics are available under the process wrapper where they are published by **NSQPING** as shown below:

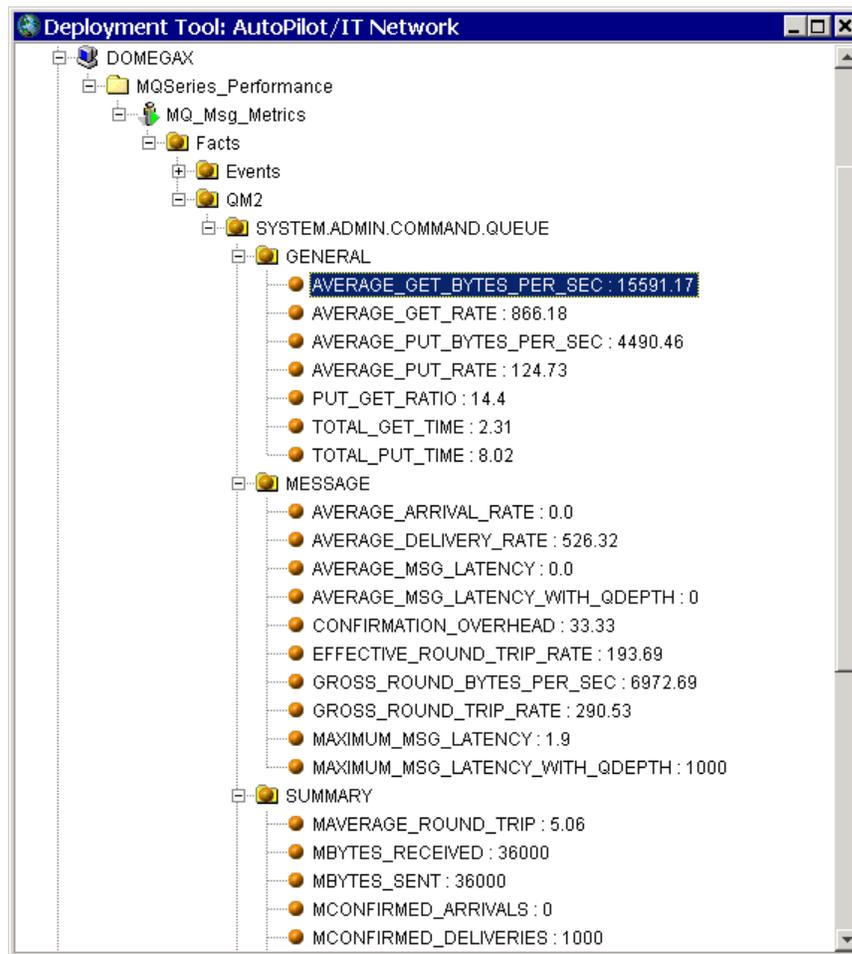


Figure 10-6. Published Message Performance Metrics

All metrics published by **NSQPING** can be included in user defined business views with rules, automation, alerts and notifications.

	NOTE:	Refer to <i>AutoPilot M6 User's Guide</i> for more information about business views.
---	--------------	--

Chapter 11: Performance Tuning

Workgroup server and managed nodes will use progressively higher CPU/memory with the increase of the AutoPilot M6 polling experts such as Queue Monitors, Channel Monitors, Queue Manager Monitor, and Node Monitors.

The number of active experts at any given moment must be kept to a minimum to reduce CPU usage. Reduce or limit CPU usage as follows:

1. Use event based monitoring using M6-WMQ Event Monitor.
2. Use fewer experts with smaller discovery scope.
3. Eliminate Channel Monitors since channel events can be used to determine channel status.
4. Increase discovery interval for each expert.
5. Use Scheduling to schedule several services one after the other. This will load balance the experts and the load they put on the system.

	NOTE:	The more polling experts there are and the less discovery interval, the more CPU it will take out of AutoPilot M6 and workgroup server. Use polling experts only when absolutely necessary.
---	--------------	---

Below are some performance tuning suggestions:

Table 11-1. Performance Tuning Suggestion	
Property	Suggestion
M6-WMQ Performance Tuning	
Node Discovery	600 or more (workgroup server properties)
WMQ Object Discovery	Interval is less than Update Interval. Should be 2/3 of the Update Interval. (node properties)
Command Limit	Should be number of MQ objects on the queue managers + 1/3 of that number. (node properties)
Request Timeout	Should be within 60-120 seconds (node properties), depending on speed/load of the slowest WebSphere MQ server machine. The slower the machine, the higher the timeout value should be.
WebSphere Plug-in Performance Tuning	
Discovery Period	120 seconds for up to 1000 objects, 300 for up to 5000 objects, 500+ seconds for 5000+ objects.
Group Connection Pool	Should not be greater than maximum number of licensed Workgroup server connections (node-based licensing only.) It also should allow room for other client connections such as M6-WMQ applications.

This Page Intentionally Left Blank

Chapter 12: Z/OS Accounting and Statistics (ZAS) Collector

Architecture Description

On z/OS, SMF (System Management Facility) is used to record information about WebSphere MQ when WMQ tracing is activated. (See Figure 12-1.) The SMF data set contains one or more files configured at the customer's site. Tracing starts with the first file in the data set. When it fills, the state of the file changes from ACTIVE to DUMP REQUIRED, an alternate SMF data set will become active, and an SMF exit occurs which activates the JCL NSQSMF in data set USER.PROCLIB.NSQSMF receives as input the name of the data set file that filled. NSQSMF then causes the dump to a temporary file.

After the dump, NSQSMF activates program NSQZAS, which reads SMF dump records from the temporary file and looks for records with type 116. These record types contain WMQ-related data as a sequence of subrecords. The type 116 records are written after an application disconnects from a queue manager or terminates. (In certain cases -- long running transactions -- records are written before the application ends.)

A subrecord type identifies the data as one of the following:

- WMQ SMF header
- Thread ID
- Thread-level accounting
- Queue-level accounting.

The type 116 records usually contain the above subrecords in that order. At times, when a record cannot hold the data for all the queues involved in an application connection to the queue manager, additional records may be written that contain the following:

- WMQ SMF header
- Thread ID
- Queue-level accounting

NSQZAS writes the ZAS PCF messages to a set of alias queues, NASTEL.ADMIN.XX.QUEUE (XX = ACCOUNTING OR STATISTICS) from which the messages are picked up by the M6-WMQ agent. The agent (possibly receiving PCF messages from the publisher NSQPUB, depending upon the configuration) sends the messages to the workgroup server, which writes them to a database. M6-WMQ Explorer can then be used to retrieve and display the data.

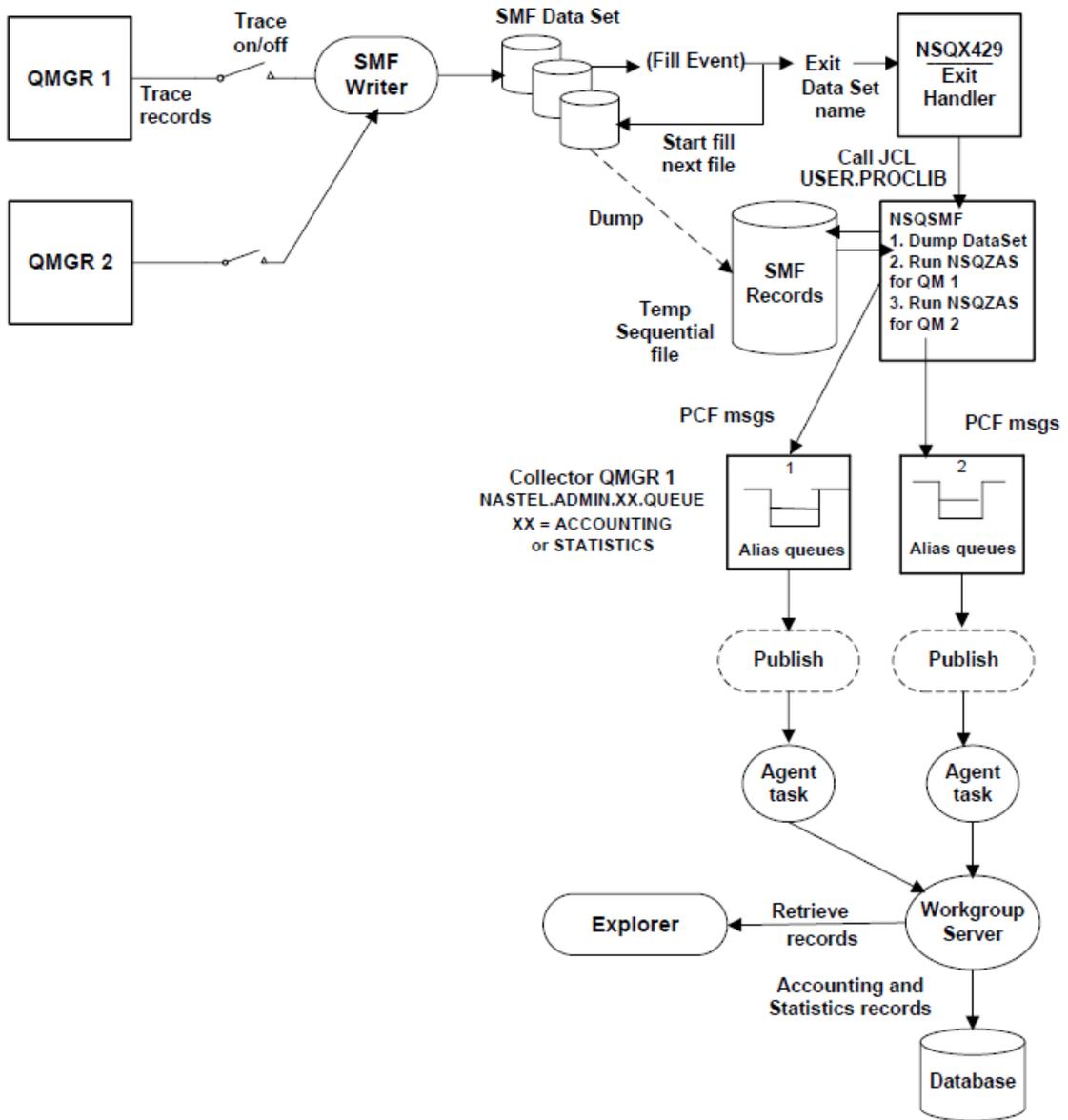


Figure 12-1. ZAS Flow Diagram

12.1 Configuring NSQZAS

	NOTE:	Steps 1 and 2 are both required.
---	--------------	----------------------------------

1. To control the SMF Exit through the Dynamic Exits Facility:
 - a. Edit the system PARMLIB(PROGxx) file used for IPL and add the lines below (typically at the end of the file) that define the SMF IEFU29 exit module name and where the exit module resides.
 - b. Replace ++hlq++ with the high level qualifier for the AutoPilot M6-WMQ installation library. The 'FIRST' qualifier means that the AutoPilot exit must be first in a chain of vendor IEFU29 exits.

```

/*-----
/* DYNAMIC EXITS
/*-----

EXIT ADD EXITNAME(SYS.IEFU29) MODNAME(NSQXU29)
DSNAME(++hlq++.LOAD) FIRST

EXIT ADD EXITNAME(SYSSTC.IEFU29) MODNAME(NSQXU29)
DSNAME(++hlq++.LOAD) FIRST

```

2. To define the SMF Exit in SMFPRMxx":
 - a. To allow the system to invoke SMF exit IEFU29, define the exit in the SMF parmlib member PARMLIB(SMFPRMxx), where xx = 00 or other ID

	NOTE:	Member PARMLIB(IEASYyy) should have a line SMF=nn, where nn is the suffix of the SMFPRMxx member in 1b above, and a line PROG=zz, where zz is the suffix of the PROGxx member in 1a above.
--	--------------	--

- b. Insert the range of SMF record types that are not to be written in the NOTYPE option of the SYS parameter. Ensure not to exclude type 116.
- c. Insert IEFU29 in the EXITS option of the SUBSYS parameter for STC.
- d. If your installation chooses not to define a SUBSYS parameter for STC, specify IEFU29 in the EXITS option of the SYS parameter. The data set names used by SMF may also be customized. The ACTIVE option should be defined versus NOACTIVE. The member should look similar to the Figure below.

```

ACTIVE                /* ACTIVE SMF RECORDING                */
DSNAME(SYS1.MAN1,
        SYS1.MAN2,
        SYS1.MAN3)
PROMPT                /* PROMPT OPERATOR                */
REC(PERM)              /* TYPE 17 PERM RECORDS ONLY      */
MAXDORM(3000)         /* WRITE IDLE BUFFER AFTER 30 MIN  */
STATUS(010000)        /* WRITE SMF STATS AFTER 1 HOUR    */
JWT(0400)             /* 522 AFTER 30 MINUTES           */
SID(&SYSNAME)
LISTDSN               /* LIST DATA SET STATUS AT IPL    */
SYS(NOTYPE(14:19,62:69),EXITS(IEFU83,IEFU84,IEFACTRT,
                             IEFUSI,IEFUJI,IEFU29),NOINTERVAL,NODETAIL)
SUBSYS(STC,EXITS(IEFU29,IEFU83,IEFU84,IEFUJP,IEFUSO,IEFACTRT))

```

Figure 12-2. SMFPRMxx member

3. Insert Exit JCL into Procedure Library as follows:

JCL NSQSMF is called by the exit assembly language program NSQXU29. To find the JCL at run time, the JCL must be found in a searchable procedure library list.

- a. Edit JCL member ++hlq++.JCL (NSQSMF).
- b. Change the M6WMQ high level qualifier (HLQ) ++hlq++ and the WMQ HLQ ++mqser++.
- c. Add NSQZAS job sections, one per queue manager being monitored and edit the PARM options. These options are:
 - m List of qmgrs for which to collect acct/stats msgs
 - mc Qmgr to use by the collector for writing the acct/stats messages to forward to M6WMQ workgroup server

Note: -m and -mc values should be the same. To process multiple qmgrs, one NSQZAS job per qmgr is required. That is, replicate lines NSQZAS through SMFIN and edit the copied lines as required. All the NSQZAS jobs will process from the same temp SMF dump file.

- qa Collect Queue Accounting on|off
- ma Collect MQI Accounting on|off
- ms Collect MQI Statistics on|off
- qs Collect Queue Statistics on|off
- cs Collect Channel Statistics on|off
- st n_secs Statistics time: aggregate over n_secs. Default: 1800.
- t Trace functions (stdout is file SYSPRINT)
- d Dump SMF records in ascii/hex
- ds Dump output to stdout. Default: file SMFDUMP.

- d. Copy it into a procedure library used on your system, such as USER.PROCLIB.

4. Configure queues used by Accounting/Statistics Collector as follows:

- a. Program NSQZAS writes PCF accounting and statistics messages to alias queues NASTEL.ADMIN.ACCOUNTING.QUEUE and NASTEL.ADMIN.STATISTICS.QUEUE.
- b. Configure the queues using MQSC member ++hlq++.JCL (MQMZAS) and command CSQUTIL. Refer to the *M6 for WMQ Installation Guide, z/OS Chapter* for details of the command. Edit the MQSC command file first if you are using the M6-MQ publisher.

	NOTE:	If you configure the M6-WMQ publisher and later decide not to use it, you must delete queue NASTEL.PUBSUB.EVENT.QUEUE, change the alias queue to point to NASTEL.EVENT.QUEUE and restart the agent.
---	--------------	---

5. Customize JCL procedures as follows:

There are several NSQZAS-related JCL procedures that need to be customized before they will work properly. Edit these JCL's in the ++hlq++.JCL data set:

- **NSQZAS** – select an SMF data set to dump and process with program NSQZAS
- **RUNZAS** – runs stand-alone test of program NSQZAS; variant of NSQAZS
- **AMQSMON** – runs modified IBM utility to do formatted browse/dump of messages on local queues NASTEL.ADMIN.ACCOUNTING (or STATISTICS).QUEUE
- **RUNCDMP** – runs IBM utility to do a formatted dump of type 115 and 116 records in an SMF dump file.

6. Create special datasets:
 - a. Create data set NASTEL.MQSMF.CNTL and copy into it JCL member ++hlq++.JCL (MQSMF). That file contains control information for the SMF dump processing:


```
INDD (DUMPIN, OPTIONS (DUMP) )
OUTDD (DUMPOUT, TYPE (116) )
```
 - b. Create GDG Data Set

Create data set NASTEL.MQSMF.DATA used to hold GDG (Generation Data Group) files that will be created by JCL NSQZAS:

```
SUBMIT ++hlq++.JCL(SMFGDG)
```
7. Configure system parameter as follows:

You can specify the interval at which WMQ collects statistics and accounting data in one of two ways (Refer to the *Using System Management Facility in InfoCenter for WMQ 6* weblink in Appendix A):

 - a. Specify a value of STATIME in the system parameters using CSQ654SP.
 - b. Specify a value for STATIME and use the SMF global accounting interval (described in *OS/390 MVS Integration and Tuning Reference*).
8. Control SMF Accounting Recording as follows:

Turn on or off recording of accounting information at the queue or queue manager level using the ACCTQ parameter of the commands DEFINE QLOCAL/QMODEL, ALTER QLOCAL/QMODEL, or DEFINE/ALTER QMGR.

The AutoPilot M6-WMQ Explorer can also be used to alter the settings. Right-click on a queue or channel and select **Properties>Monitoring** to control an individual queue or channel. For a queue manager, you can affect accounting for all queues. See the *AutoPilot M6 for WMQ Administrator's Guide* for details.

12.2 Starting NSQZAS

If the configuration steps above were performed, then NSQZAS processing should start automatically each time an SMF data set is full with data collected from WMQ applications.

12.3 Stopping/Restarting NSQZAS

To stop collection of accounting and statistics, use the Stop Trace command.

```
%qmgr stop trace(A) class(3)
```

To restart data collection, use the **Start Trace** command to start WMQ tracing using SMF.

```
%qmgr start trace(A) class(3)
```

A response will be displayed similar to the following:

```
RESPONSE=ADCD CSQW130I %CSQ7 'A' TRACE STARTED, ASSIGNED TRACE NUMBER 02
RESPONSE=ADCD CSQ9022I %CSQ7 CSQWVCM1 ' START TRACE' NORMAL COMPLETION
```

These commands must be repeated for each queue manager.

To trigger ZAS message generation manually before an SMF data set is full, use the **SWITCH SMF** command. (Refer to the *SMF Commands* Section.)



NOTE:

To collect channel statistics, you must first stop the channels.

12.4 Mapping Between Accounting and Statistics PCF Messages on Z/OS and Distributed Platforms

The following tables show the mapping from accounting and statistics message parameters on distributed platforms to the equivalent parameter in SMF records on z/OS.

The z/OS parameter names are from the various table layouts in *WebSphere MQ, Version 6.0, Information Center*, chapters *z/OS: System Setup Guide > Monitoring Performance and Resource Usage > Interpreting WebSphere MQ accounting data*,

<http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/index.jsp>. Distributed platform parameter names and descriptions are from *Monitoring WebSphere MQ, Version 6.0, Part 4. Accounting and statistics messages*, <http://publibfp.boulder.ibm.com/epubs/pdf/csqzax05.pdf>.

In the following table:

- **X** indicates *no equivalent mapping*.
- **Req** indicates *required*
- **WA** indicates *when parameter is available*

Table 12-1. MQI Accounting Message Parameters			
Distributed	REQ or WA	z/OS	z/OS Description
QueueManager	Req	Y	Available in the program loop
IntervalStartDate Date of the start of the monitoring period.	Req	WTASINTS	Interval start - for post processing. Includes date and time. [5] Might be same as or calculated from IntervalEndDate
IntervalStartTime Time of start of the monitoring period.	Req	WTASINTS	[5]
IntervalEndDate	Req	WTASINTE or SM116DTE	Interval end - for post processing. Includes date and time [5] Date when SMF moved record.
IntervalEndTime	Req	WTASINTE or SM116TME	Time when SMF moved record.
CommandLevel Queue manager command level.	Req	Y	Available from qmgr attributes.
ConnectionId	Req	QWHCCN or WTIDCHLC	Connection name. CONNNAME of the Channel. For example: For a sender WINMVS2B(2162) For a receiver, where channel came from: 9.20.101.14 or NYCHOST
SeqNumber Sequence number. This value is incremented for each subsequent record for long running connections.	Req	X	Use constant 1 if it cannot be detected on z/OS.
ApplicationName	Req	??	Transaction name
ApplicationPid	Req	X	Use constant 1
ApplicationTid	Req	QWHCTNO - for CICS only, otherwise X	CICS thread number.
UserId	Req	QWHCAID or QWHCOPID	User ID associated with the OS/390 job; or User ID associated with the transaction.
ConnDate	WA	X	

Table 12-1. MQI Accounting Message Parameters

Distributed	REQ or WA	z/OS	z/OS Description
ConnTime	WA	X	
ChannelName	WA	WTIDCHL Channel name	
DiscDate	WA	X	
DiscTime	WA	X	
DiscType	WA	X	
OpenCount The number of objects opened. This parameter is an integer list indexed by object type.	WA	X	QMSTOPEN is number of MQOPEN requests, independent of object type.
OpenFail	WA	X	
CloseCount The number of objects closed. This parameter is an integer list indexed by object type.	WA	X	QMSTCLOS is number of MQCLOSE requests, independent of object type.
CloseFailCount	WA	X	
PutCount The number of persistent and nonpersistent messages successfully put to a queue, with the exception of messages put using the MQPUT1 call. This parameter is an integer list indexed by persistence value.	WA	X	QMSTPUT is number of MQPUT requests, independent of persistence value. QMACPUTA + QMACPUTB + QMACPUTC + QMACPUTD Number of MQPUT requests for messages of length 0 through 99 bytes + lengths 100-999 + lengths 1000-9999 + lengths > 10,000, All independent of persistence value. QMAC* data available from message manager accounting record QMAC
PutFailCount	WA	X	
Put1Count The number of persistent and nonpersistent messages successfully put to the queue using MQPUT1 calls. This parameter is an integer list indexed by persistence value.	WA	X	QMSTPUT1 is number of MQPUT1 requests, independent of persistence value.
Put1FailCount	WA	X	
GetCount The number of successful destructive gets for persistent and nonpersistent messages. This parameter is an integer list indexed by persistence value.	WA	X	QMSTGET Number of MQGET requests, independent of persistence value. Also: QMACGETA, QMACGETB, QMACGETC, QMACGETD Number of MQGET requests for messages of length 0 through 99 bytes + Lengths 100-999 + Lengths 1000-9999 + Lengths > 10,000, all independent of persistence value.
GetFailCount The number of unsuccessful destructive gets.	WA	X	

Table 12-1. MQI Accounting Message Parameters

Distributed	REQ or WA	z/OS	z/OS Description
<i>GetBytes</i> Total number of bytes retrieved for persistent and nonpersistent messages. This parameter is an integer list indexed by persistence value.	WA	X	
<i>BrowseCount</i>	WA	X	
<i>BrowseFailCount</i>	WA	X	
<i>BrowseBytes</i>	WA	X	
<i>CommitCount</i>	WA	X	
<i>CommitFailCount</i>	WA	X	
<i>BackCount</i>	WA	X	
<i>InqCount</i> The number of successful objects inquired upon. This parameter is an integer list indexed by object type.	WA	X	QMSTINQ is number of MQINQ requests, independent of object type.
<i>InqFailCount</i>	WA	X	
<i>SetCount</i> The number of successful MQSET calls. This parameter is an integer list indexed by object type.	WA	X	QMSTSET is number of MQSET requests, independent of object type.
<i>SetFailCount</i>	WA	X	

Table 12-2. Queue Accounting Message Parameters

Distributed	REQ or WA	z/OS	z/OS Description
<i>QueueManager</i>	Req	Y	Available in the program loop
<i>IntervalStartDate</i> The date of the start of the monitoring period.	Req	WTASINTS	Interval start - for post processing. Includes date and time. [5] Might be same as or calculated from IntervalEndDate
<i>IntervalStartTime</i> The time of the start of the monitoring period.	Req	WTASINTS	Might be calculated from IntervalEndTime
<i>IntervalEndDate</i>	Req	WTASINTE or SM116DTE	Interval end - for post processing Date when SMF moved record.
<i>IntervalEndTime</i>	Req	WTASINTE or SM116TME	Time when SMF moved record.
<i>CommandLevel</i> The queue manager command level.	Req	Y	Available from qmgr attributes
<i>ConnectionId</i>	Req	QWHCCN or WTIDCHLC	Connection name. CONNNAME of the Channel For example: WINMVS2B(2162)
<i>SeqNumber [3]</i> The sequence number. This value is incremented for each subsequent record for long running connections.	Req	X	Use constant 0
<i>ApplicationName</i>	Req	Y	Transaction name, CCBNAME, Connection name
<i>ApplicationPid</i>	Req	X	Use constant 1
<i>ApplicationTid</i>	Req	QWHCTNO - for CICS only, otherwise X	CICS thread number.
<i>UserId</i>	Req	QWHCAID QWHCOPID	User ID associated with the OS/390 job; or User ID associated with the transaction.
<i>ObjectCount</i> The number of queues accessed in the interval for which accounting data has been recorded. This value is set to the number of <i>QAccountingData</i> PCF groups contained in message (1 to 100) [3]	Req	Y	WQ structure can have data for more than 1 queue, so count must be available some place. WTASWQCT From version 6 – Count of WQ blocks for this thread.
<i>QAccountingData</i> Grouped parameters specifying accounting details for a queue.	Req	Y	
NOTE: One or more of the following data appear in the <i>QAccountingData</i> group			NOTE: The data below is in a structure called the WQ in SMF 116 subtype 1 records.
<i>QName</i>	WA	OBJNAME	
<i>CreateDate</i>	WA	Y	Available from q attributes.
<i>CreateTime</i>	WA	Y	Avail from q attributes
<i>QType</i>	WA	QTYPE	Avail from q attributes
<i>QDefinitionType</i>	WA	Y	Avail from q attributes
<i>OpenCount</i>	WA	OPENN	

Table 12-2. Queue Accounting Message Parameters

Distributed	REQ or WA	z/OS	z/OS Description
<i>OpenDate</i>	WA	OPENTIME	IBM STCK format
<i>OpenTime</i>	WA	OPENTIME	
<i>CloseDate</i>	WA	CLOSTIME	IBM STCK format
<i>CloseTime</i>	WA	CLOSTIME	
<i>PutCount</i> The number of persistent and nonpersistent msgs successfully put to the queue, with the exception of MQPUT1 calls. This parameter is an integer list indexed by persistence value.	WA	Y	PUTN=Number of MQPUT requests PUTPMSG=Number of persistent messages retrieved using MQPUT. PUTN – PUTPMSG = Number of non-persistent messages.
<i>PutFailCount</i>	WA	X	
<i>Put1Count</i> The number of persistent and nonpersistent msgs successfully put to the queue using MQPUT1 calls. This parameter is an integer list indexed by persistence value.	WA	Y	Number of MQPUT1 requests PUT1PMSG= Number of persistent messages created using MQPUT1. PUT1N - PUT1PMSG = Number of non-persistent messages.
<i>Put1FailCount</i> The number of unsuccessful attempts to put a message using MQPUT1 calls.	WA	X	
<i>PutBytes</i> The total number of bytes put for persistent and nonpersistent messages. This parameter is an integer list indexed by persistence value.	WA	PUTBYTES	Total number of bytes put successfully. Assume all persistent; non-persistent = 0.
<i>PutMinBytes</i> Smallest persistent and nonpersistent message size placed on the queue. This parameter is an integer list indexed by persistence value.	WA	PUTMINMS	Put minimum message size Assume all persistent; non-persistent = 0.
<i>PutMaxBytes</i> Largest persistent and nonpersistent message size placed on the queue. This parameter is an integer list indexed by persistence value.	WA	PUTMAXMS	Put maximum message size. Assume all persistent; non-persistent = 0.
<i>GetCount</i> The number of successful destructive MQGET calls for persistent and nonpersistent messages. This parameter is an integer list indexed by persistence value.	WA	Y	GETDVAL (from v6) = Number of successful destructive MQGET Calls. - OR - Number persistent messages retrieved using MQGET = GETPMSG. Non-persistence msgs = VALIDGET-GETPMSG where VALIDGET = Number of MQGETs with data NOTE: GETN = GETA + GETS + GETBRWA + GETBRWS
<i>GetFailCount</i> The number of unsuccessful destructive gets.	WA	X	

Table 12-2. Queue Accounting Message Parameters

Distributed	REQ or WA	z/OS	z/OS Description
<i>GetBytes</i> The number of successful destructive MQGET calls for persistent and nonpersistent messages. This parameter is an integer list indexed by persistence value.	WA	GETBYTES	Total number of bytes gotten successfully. Assume all persistent; non-persistent=0.
<i>GetMinBytes</i> This parameter is an integer list indexed by persistence value.	WA	GETMINMS	Get minimum message size
<i>GetMaxBytes</i> This parameter is an integer list indexed by persistence value.	WA	GETMAXMS	Get maximum message size
<i>BrowseCount</i>	WA	Y	MQGET Browse Any + Browse Specific = GETBRWA + GETBRWS
<i>BrowseFailCount</i>	WA	X	
<i>BrowseBytes</i> The number of bytes read in non-destructive gets that returned persistent messages.	WA MQCFIL64	X	
<i>BrowseMinBytes</i> This parameter is an integer list indexed by persistence value.	WA	X	
<i>BrowseMaxBytes</i> This parameter is an integer list indexed by persistence value.	WA	X	
<i>TimeOnQMin</i> The shortest time a persistent and nonpersistent message remained on the queue before being retrieved, in microseconds. This parameter is an integer list indexed by persistence value.	WA INTEGER	MINLATNT	Minimum latency of message IBM STCK format, long long. Must convert to INTEGER Microsecs. [5] Assume all persistent; non-persistent=0.
<i>TimeOnQAvg</i>	WA INTEGER	X	
<i>TimeOnQMax</i>	WA INTEGER	MAXLATNT	Maximum latency of message IBM STCK format, long long. Must convert to INTEGER Microsecs. [5] Assume all persistent; non-persistent=0.

[1] Intentionally deleted.

[2] Intentionally deleted.

[3] “Queue accounting message data consists of a number of PCF parameters, and between one and one hundred *QAccountingData* PCF groups. There is one *QAccountingData* PCF group for every queue that had accounting data collected. If an application accesses more than 100 queues, multiple accounting messages are generated. Each message has the *SeqNumber* in the MQCFH (PCF header) updated accordingly, and the last message in the sequence has the *Control* parameter in the MQCFH specified as MQCFC_LAST.”

[4] “The SMF statistics broadcast occurs, and it is a long running application, such as a channel. This means it was running prior to the last SMF statistics broadcast. Before Version 5.2 the records were produced only when the job or application ended. So for a channel that was active for a year, you

would get one record. In Version 5.2 you can request that records are produced at the SMF broadcast, typically every 30 minutes, by setting STATIME=0 in the CSQ6SYSP system parameter macro.”

- [5] “The time in the SMF record (SM116TME) is the time the record was produced (the number of hundredths of a second since midnight). It is usually close (within a second) to the time the transaction or channel ended. If you need more accurate times, you should use the time interval end time (WTASINTE), which is in STCK (Store Clock) format.”

“S/390 Store Clock format (STCK) is a double word where bit 53 is a microsecond. To convert a STCK value to microseconds ignore the bottom 12 bits. In C this can be done by treating the values as long long and dividing by 4096 to get to microseconds.”

- [6] Other z/OS statistics available are: Data Manager, Buffer Manager, Lock Manager, DB2 Manager, CF Manager.

Table 12-3. Queue Statistics Message

Distributed	REQ or WA	z/OS	z/OS Description
<i>QueueManager</i>	Req	Y	
<i>IntervalStartDate</i> The date of the start of the monitoring period.	Req	WTASINTS	Interval start - for post processing. Includes date and time. [5]
<i>IntervalStartTime</i> The time of the start of the monitoring period.	Req	WTASINTS	Might be calculated from IntervalEndTime
<i>IntervalEndDate</i>	Req	WTASINTE OR SM116DTE	Interval end - for post processing Date when SMF moved record.
<i>IntervalEndTime</i>	Req	WTASINTE OR SM116TME	Time when SMF moved record.
<i>CommandLevel</i> The queue manager command level.	Req	Y	Available from qmgr attributes
<i>ObjectCount</i> The number of queues accessed in the interval for which accounting data has been recorded. This value is set to the number of <i>QAccountingData</i> PCF groups contained in the message (1 to 100) [3]	Req	Y	WTASWQCT From version 6 – Count of WQ blocks for this thread.
<i>QStatisticsData</i> Grouped parameters specifying statistics for a queue.	Req	Y	
NOTE: One or more of the following data appear in the QStatisticsData group			
<i>QName</i>	WA	OBJNAME	
<i>CreateDate</i>	WA	Y	Avail from q attributes
<i>CreateTime</i>	WA	Y	Avail from q attributes
<i>QType</i>	WA	QTYPE	Avail from q attributes
<i>QDefinitionType</i>	WA	Y	Avail from q attributes
<i>QMinDepth</i>	WA	X	
<i>QMaxDepth</i>	WA	Y	In WQ block
<i>TimeOnQAvg</i>	WA INTEGER	X	
<i>PutCount</i>	WA The number of persistent and nonpersistent msgs successfully put to the queue, with the exception of MQPUT1 calls. This parameter is an integer list indexed by persistence value.	Y	PUTN=Number of MQPUT requests PUTPMSG=Number of persistent msgs retrieved using MQPUT. PUTN – PUTPMSG = nbr of non-persistent msgs.
<i>PutFailCount</i>	WA	X	

Table 12-3. Queue Statistics Message

Distributed	REQ or WA	z/OS	z/OS Description
<i>Put1Count</i>	WA The number of persistent and nonpersistent msgs successfully put to the queue using MQPUT1 calls. This parameter is an integer list indexed by persistence value.	Y	Number of MQPUT1 requests PUT1PMSG= Number of persistent msgs created using MQPUT1. PUT1N - PUT1PMSG = non-persistent msgs.
<i>Put1FailCount</i>	WA The number of unsuccessful attempts to put a message using MQPUT1 calls.	X	
<i>PutBytes</i>	WA The total number of bytes put for persistent and nonpersistent messages. This parameter is an integer list indexed by persistence value.	PUTBYTES	Total number of bytes put successfully. Assume all persistent; non-persistent = 0.
<i>GetCount</i>	WA The number of successful destructive MQGET calls for persistent and nonpersistent messages. This parameter is an integer list indexed by persistence value.	Y	GETDVAL (from v6) = Number of successful destructive MQGET Calls. - OR - Nbr persistent messages retrieved using MQGET = GETPMSG. Non-persistence msgs = VALIDGET-GETPMSG where VALIDGET = Number of MQGET call with data NOTE: GETN = GETA + GETS + GETBRWA + GETBRWS
<i>GetFailCount</i>	WA The number of unsuccessful destructive gets.	X	
<i>GetBytes</i>	WA The number of successful destructive MQGET calls for persistent and nonpersistent messages. This parameter is an integer list indexed by persistence value.	GETBYTES	Total number of bytes got successfully. Assume all persistent; non-persistent=0.
<i>BrowseCount</i>	WA	Y	MQGET Browse Any + Browse Specific = GETBRWA + GETBRWS
<i>BrowseFailCount</i>	WA	X	
<i>BrowseBytes</i>	WA The number of bytes read in non-destructive gets that returned persistent messages. MQCFIL64.	Y	

Table 12-3. Queue Statistics Message

Distributed	REQ or WA	z/OS	z/OS Description
<i>PurgeCount</i>	WA	X	
<i>ExpiredMsgCount</i>	WA	X	
<i>NonQueuedMsgCount</i>	WA	X	

Table 12-4. MQI Statistics Parameters

Distributed	REQ or WA	z/OS	z/OS Description
<i>QueueManager</i>	Req	See previous table	
<i>IntervalStartDate</i> The date of the start of the monitoring period.	Req	See previous table	
<i>IntervalStartTime</i> The time of the start of the monitoring period.	Req	See previous table	
<i>IntervalEndDate</i>	Req	See previous table	
<i>IntervalEndTime</i>	Req	See previous table	
<i>CommandLevel</i> The queue manager command level.	Req	See previous table	
			Following are message manager statistics in QMST record [6]. Message manager processes the MQI verbs.
<i>ConnCount</i>	WA	X	
<i>ConnFailCount</i>	WA	X	
<i>ConnsMax</i>	WA	X	
<i>DiscCount</i>	WA	X	
<i>OpenCount</i> The number of objects successfully opened. This parameter is an integer list indexed by object type.	WA	QMSTOPEN	Number of MQOPEN requests. Not provided by object type. May have to assume all for QLOCAL.
<i>OpenFailCount</i>	WA	X	
<i>CloseCount</i> The number of objects successfully closed. This parameter is an integer list indexed by object type.	WA	QMSTCLOS	Number of MQCLOSE requests. Not provided by object type. May have to assume all for QLOCAL
<i>CloseFailCount</i>	WA	X	
<i>InqCount</i>	WA	X	
<i>InqFailCount</i>	WA	X	
<i>SetCount</i>	WA	QMSTSET	Number of MQSET requests
<i>SetFailCount</i>	WA	X	
<i>PutCount</i> The number of persistent and nonpersistent messages successfully put to a queue using MQPUT requests. This parameter is an integer list indexed by persistence value.	WA	QMSTPUT	Number of MQPUT requests. Assume all persistent; non-persistent=0.
<i>PutFailCount</i>	WA	X	
<i>Put1Count</i>	WA	QMSTPUT1	Number of MQPUT1 requests. Assume all persistent; non-persistent=0.
<i>Put1FailCount</i>	WA	X	

Table 12-4. MQI Statistics Parameters

Distributed	REQ or WA	z/OS	z/OS Description
<i>PutBytes</i>	WA	X	
<i>GetCount</i>	WA	QMSTGET	Number of MQGET requests
<i>GetFailCount</i>	WA	X	
<i>GetBytes</i> The number of successful destructive MQGET calls for persistent and nonpersistent messages. This parameter is an integer list indexed by persistence value	WA	X	
<i>BrowseCount</i>	WA	X	
<i>BrowseFailCount</i>	WA	X	
<i>BrowseBytes</i>	WA	X	
<i>CommitCount</i>	WA	X	
<i>CommitFailCount</i>	WA	X	
<i>BackCount</i>	WA	X	
<i>ExpiredMsgCount</i>	WA	X	

Table 12-5. Queue Statistics Message

Distributed	REQ or WA	z/OS	z/OS Description
<i>QueueManager</i>	Req	Y	
<i>IntervalStartDate</i> The date of the start of the monitoring period.	Req	WTASINTS	Interval start - for post processing. Includes date and time. [5]
<i>IntervalStartTime</i> The time of the start of the monitoring period.	Req	WTASINTS	Might be calculated from IntervalEndTime
<i>IntervalEndDate</i>	Req	WTASINTE OR SM116DTE	Interval end - for post processing Date when SMF moved record.
<i>IntervalEndTime</i>	Req	WTASINTE OR SM116TME	Time when SMF moved record.
<i>CommandLevel</i> The queue manager command level.	Req	Y	Available from qmgr attributes
<i>ObjectCount</i> The number of queues accessed in the interval for which accounting data has been recorded. This value is set to the number of <i>QAccountingData</i> PCF groups contained in the message (1 to 100) [3]	Req	Y	WTASWQCT From version 6 – Count of WQ blocks for this thread.
<i>QStatisticsData</i> Grouped parameters specifying statistics for a queue.	Req	Y	
NOTE: One or more of the following data appear in the QStatisticsData group			
<i>QName</i>	WA	OBJNAME	
<i>CreateDate</i>	WA	Y	Avail from q attributes
<i>CreateTime</i>	WA	Y	Avail from q attributes
<i>QType</i>	WA	QTYPE	Avail from q attributes
<i>QDefinitionType</i>	WA	Y	Avail from q attributes
<i>QMinDepth</i>	WA	X	
<i>QMaxDepth</i>	WA	Y	In WQ block
<i>TimeOnQAvg</i>	WA INTEGER	X	

Table 12-5. Queue Statistics Message

Distributed	REQ or WA	z/OS	z/OS Description
<i>PutCount</i>	WA The number of persistent and nonpersistent msgs successfully put to the queue, with the exception of MQPUT1 calls. This parameter is an integer list indexed by persistence value	Y	PUTN=Number of MQPUT requests PUTPMSG=Number of persistent msgs retrieved using MQPUT. PUTN – PUTPMSG = nbr of non-persistent msgs.
<i>PutFailCount</i>	WA	X	
<i>Put1Count</i>	WA The number of persistent and nonpersistent msgs successfully put to the queue using MQPUT1 calls. This parameter is a list indexed by persistence value.	Y	Number of MQPUT1 requests PUT1PMSG= Number of persistent msgs created using MQPUT1. PUT1N - PUT1PMSG = non-persistent msgs
<i>Put1FailCount</i>	WA The number of unsuccessful attempts to put a message using MQPUT1 calls.	X	
<i>PutBytes</i>	WA The total number of bytes put for persistent and nonpersistent messages. This parameter is an integer list indexed by persistence value.	PUTBYTES	Total number of bytes put successfully. Assume all persistent; non-persistent = 0.
<i>GetCount</i>	WA The number of successful destructive MQGET calls for persistent and nonpersistent messages. This parameter is an integer list indexed by persistence value.	Y	GETDVAL (from v6) = Number of successful destructive MQGET Calls. - OR - Nbr persistent messages retrieved using MQGET = GETPMSG. Non-persistence msgs = VALIDGET-GETPMSG where VALIDGET = Number of MQGET call with data NOTE: GETN = GETA + GETS + GETBRWA + GETBRWS
<i>GetFailCount</i>	WA The number of unsuccessful destructive gets.	X	
<i>GetBytes</i>	WA The number of successful destructive MQGET calls for persistent and nonpersistent messages. This parameter is an integer list indexed by persistence value.	GETBYTES	Total number of bytes got successfully. Assume all persistent; non-persistent=0.
<i>BrowseCount</i>	WA	Y	MQGET Browse Any + Browse Specific = GETBRWA + GETBRWS

Table 12-5. Queue Statistics Message

Distributed	REQ or WA	z/OS	z/OS Description
<i>BrowseFailCount</i>	WA	X	
<i>BrowseBytes</i>	WA The number of bytes read in non-destructive gets that returned persistent messages. MQCFIL64.	Y	
<i>PurgeCount</i>	WA	X	
<i>ExpiredMsgCount</i>	WA	X	
<i>NonQueuedMsgCount</i>	WA	X	

Table 12-6. Channel Statistics Message

Distributed	REQ or WA	z/OS	z/OS Description
<i>QueueManager</i>	Req	See previous table	
<i>IntervalStartDate</i> The date of the start of the monitoring period.	Req	See previous table	
<i>IntervalStartTime</i> The time of the start of the monitoring period.	Req	See previous table	
<i>IntervalEndDate</i>	Req	See previous table	
<i>IntervalEndTime</i>	Req	See previous table	
<i>CommandLevel</i> The queue manager command level.	Req	See previous table	
			Following data is in Task Id structure WTID
<i>ChannelName</i>	Req	WTIDCHL	Channel name Sender: The channel name For example: V521.V52A Rcvr: The job name For example: V52ACHIN
<i>ChannelType</i>	Req	Y	Avail from channel attributes
<i>ChlStatisticsData</i> Grouped parameters specifying statistics details for a channel.	Req		
<i>RemoteQmgr</i>	WA	X	
<i>ConnectionName</i>	WA	WTIDCHLC	Channel connection name Sender: This is defined in the CONNAME attribute of the Channel For example: WINMVS2B(2162) Rcvr: Where the channel came from For example: 9.20.101.14
<i>MsgCount</i> The number of [persistent + nonpersistent] messages sent or received.	WA	Y	Derived from WQ data
<i>TotalBytes</i> The number of [persistent + nonpersistent] bytes sent or received.	WA	Y	Derived from WQ data
<i>NetTimeMin</i>	WA	X	
<i>NetTimeAvg</i>	WA	X	
<i>NetTimeMax</i>	WA	X	
<i>ExitTimeMin</i>	WA	X	
<i>ExitTimeAvg</i>	WA	X	

Table 12-6. Channel Statistics Message			
Distributed	REQ or WA	z/OS	z/OS Description
<i>ExitTimeMax</i>	WA	X	
<i>FullBatchCount</i>	WA	X	
<i>IcmlBatchCount</i>	WA	X	
<i>AverageBatchSize</i>	WA	X	
<i>PutRetryCount</i>	WA	X	

12.5 Best Practices for ZAS Collection

For efficient collection of data, note the following configuration points:

1. Minimize the number of types of SMF records collected.

The only SMF records needed are accounting, record type 116. In SMFPRMxx member of SYS1.PARMLIB, specify:

```
SYS(TYPE(116, nnn,...))
```

where nnn are other record types that you want collected for other vendor products. The more record types that are written by SMF, the greater the system overhead.

2. Provide minimally 3 SMF datasets for collection.

In SMFPRxx, define the names of the SMF data sets to be used, for example:

```
DSNAME(SYS1.MAN1,
        SYS1.MAN2,
        SYS1.MAN3)
```

Three or more data sets allow time for the SMF writer to cycle through the data sets without overwriting any one data set too soon, in case you want to view or re-process it.

3. Size the SMF datasets to minimize frequent SMF exits.

The SMF IEFU29 exit is triggered and invokes NSQZAS each time an SMF data set fills. Making the data set sizes too small will cause them to fill more frequently, especially if there are multiple vendor products that do SMF data collection. This will cause more frequent exits and processing program startups. Making the data sets too large will reduce the fill rate and the frequency at which the SMF data is processed and converted to accounting and statistics data. In the latter case, if the fill rate is taking too long, a JCL to *SWITCH SMF* can be scheduled to execute more frequent SMF dumps and NSQZAS processing.

An SMF data set sizing estimation formula follows.

Standard SMF 116 header record:	SMFHdr_size	28 bytes
WMQ SMF header:	QWHS_size	128
TaskID record:	WTID_size	208
Thread/task level accounting record:	WTAS_size	700
Queue accounting record:	WQ_size	592

nq = avg. number of queues accessed during an application connection

nc = avg. number of application connections to be handled in an SMF data set

For each application connection to a queue manager, the SMF data size is:

$$\text{SMFHdr_size} + \text{QWHS_size} + \text{WTID_size} + \text{WTAS_size} + \text{nq} * \text{WQ_size} = 1064 + \text{nq} * 592$$

Total estimated SMF data set storage required for WMQ =

$$\text{nc} * (1064 + \text{nq} * 592) =$$

$$\text{nc} * 1064 + \text{nc} * \text{nq} * 592 \text{ bytes}$$

4. Maximize NSQZAS statistics aggregation interval to reduce network traffic.

Set the NSQZAS startup option `-t` value (the aggregation interval for collecting SMF WMQ message data into PCF statistics messages) to the maximum possible for your data analysis purposes. This will minimize the number of PCF statistics generated and reduce network traffic overhead from the z/OS machine to the AutoPilot workgroup server.

For example:

If an SMF data set, after filling, contains 30 minutes of uniformly distributed WMQ messages, a `-t` value of 1800 secs (30 mins) would cause the generation of 1 MQI statistics message covering that entire 30 minute interval; `m` queue statistics messages (`m` = number of queues accessed by all WMQ applications in the 30 minute interval) and `n` channel statistics message (`n` = number of active channels in the interval), or a total of $(1+m+n)$ messages. A `-t` value of 300 (5 mins) would cause approximately 6 times as many messages.

12.6 Initializing SMF Data Sets

To start off a test from a known state, the existing SMF data sets should be cleared as follows:

1. Use SDSF command **/D SMF** to display the SMF data sets and their state.
2. Open `++hlq++JCL(SMFCLEAN)`.
3. Edit the DUMPIN line and change the MANnn number to that of a data set in the DUMP REQUIRED state.
4. Repeat for each such data set.
5. If a data set is in the active state, use the **switch smf** command to switch the active SMF data set to the next available member.

switch SMF

```
RESPONSE=ADCD      IEE360I SMF NOW RECORDING ON SYS1.MAN2 ON
Z7SYS1 RESPONSE=TIME=15:59:28
```

There must be at least one data set in the ALTERNATE state. For example, the D SMF command might show this before the switch command:

```
RESPONSE=ADCD
IEE949I 15.53.15 SMF DATA SETS 209
      NAME                VOLSER    SIZE (BLKS)  %FULL
      STATUS P-SYS1.MAN1  Z7SYS1
      7200          5
      S-SYS1.MAN2        Z7SYS1    1800          0      ALTERNATE
      S-SYS1.MAN3        Z7SYS1    1800          0      ALTERNATE
```

After the switch command, the display SMF response shows the active data set is now the first alternate, SYS1.MAN2:

```
RESPONSE=ADCD
IEE949I 16.02.08 SMF DATA SETS 252
      NAME                VOLSER    SIZE (BLKS)  %FULL  STATUS
      P-SYS1.MAN1        Z7SYS1    7200          7      DUMP REQUIRED
      S-SYS1.MAN2        Z7SYS1    1800          2      ACTIVE
      S-SYS1.MAN3        Z7SYS1    1800          0      ALTERNATE
```

The data set will now be in DUMP REQUIRED state and SMFCLEAN can be run for that data set.

12.7 SMF Commands

12.7.1 Running SMF Command from SDSF

1. Go to the SDSF panel by logging in and entering command “SDSF” or “=m.5” on an Option or Command Input line.
2. On the command line, enter `/command` using one of the commands below. The command may be entered in upper or lower case.
3. On the SDSF panel you may:
 - a. Enter a slash alone (/) on the COMMAND INPUT line to see a history of previous commands.
 - b. Page up and down through the command history.
 - c. Move the cursor to one of the commands and press **Enter** to recall that command to the command line. Then press **Enter** again to execute it.

12.7.2 Displaying SMF Status

To display SMF status, type

```
D SMF
```

Possible responses are:

- a. RESPONSE=ADCD IEE351I SMF SYS1.MAN RECORDING NOT BEING USED
This response indicates SMF has not been activated. Use the Start SMF command.

- b. RESPONSE=ADCD
IEE949I 14.46.02 SMF DATA SETS 029

NAME	VOLSER	SIZE (BLKS)	%FULL	STATUS
P-SYS1.MAN1	Z7SYS1	7200	28	ACTIVE
S-SYS1.MAN2	Z7SYS1	1800	49	DUMP REQUIRED
S-SYS1.MAN3	Z7SYS1	1800	12	DUMP REQUIRED

This response shows which data set is active and which require dumping.

12.7.3 Activating SMF for Recording Record Types 116

```
SETSMF SYS (TYPE 116)
```

```
RESPONSE=ADCD IEE712I SETSMF PROCESSING COMPLETE
```

12.7.4 Starting SMF

Use this command to start WMQ tracing using SMF.

```
set SMF=PQ
```

PQ in this example is the suffix of the PARMLIB(PROC.xx) file. (See [section 12.1](#).) You will see a message that requires you to respond as follows:

1. On the command line, enter LOG. At the bottom of the LOG screen you will see a message similar to
0000000 ADCD 14.41.53 *03 IEE357A REPLY WITH SMF VALUES OR U
2. At the top of the screen, on the COMMAND INPUT line, respond with /03U
Use the exact sequence number appearing at the beginning of the REPLY prompt. In this example, it is 03.
3. Repeat the display SMF status command 'D SMF'. You should now see a reply similar to this:

```
RESPONSE=ADCD
      IEE949I 14.46.02 SMF DATA SETS 029
      NAME              VOLSER      SIZE (BLKS)  %FULL  STATUS
      P-SYS1.MAN1      7SYS1       7200         28     ACTIVE
      S-SYS1.MAN2      Z7SYS1       1800         49     DUMP REQUIRED
      S-SYS1.MAN3      Z7SYS1       1800         12     DUMP REQUIRED
```

12.7.5 Displaying Trace Status

Use this command to see the status of WMQ tracing.

```
%qmgr display trace
```

```
RESPONSE=ADCD
      CSQW127I %CSQ7 CURRENT TRACE ACTIVITY IS
      - TNO TYPE CLASS DEST USERID RMID
      1 GLOBAL 01 RES * *
      2 ACCTG 03 SMF * *
      00 CHINIT * RES * * END OF TRACE REPORT
      CSQ9022I %CSQ7 CSQWVCM1 ' DISPLAY TRACE ' NORMAL COMPLETION
```

The second line in the trace shows that SMF trace is active for accounting, class 3. This is what is needed for NSQZAS to be able to collect WMQ accounting and statistics. If this line is not seen, use the `Start Trace` command.

12.7.6 Starting Trace

Use this command to start WMQ tracing using SMF.

```
%qmgr start trace(A) class(3)
```

```
RESPONSE=ADCD CSQW130I %CSQ7 'A' TRACE STARTED, ASSIGNED TRACE NUMBER 02
RESPONSE=ADCD CSQ9022I %CSQ7 CSQWVCM1 ' START TRACE ' NORMAL COMPLETION
```

12.7.7 Stopping Trace

Use this command to stop WMQ tracing using SMF.

```
%qmgr stop trace(A) class(3)
```

```
RESPONSE=ADCD CSQW131I %CSQ7 STOP TRACE SUCCESSFUL FOR TRACE NUMBER(S)
RESPONSE=02
RESPONSE=ADCD CSQ9022I %CSQ7 CSQWVCM1 ' STOP TRACE ' NORMAL COMPLETION
```

12.7.8 Switching the Active SMF Data Set

Use this command to switch the active SMF data set to the next available member.

switch SMF

```
RESPONSE=ADCD      IEE360I SMF NOW RECORDING ON SYS1.MAN2 ON Z7SYS1
RESPONSE=TIME=15:59:28
```

There must be at least one data set in the ALTERNATE state. For example, the D SMF command might show this before the switch command:

```
RESPONSE=ADCD
IEE949I 15.53.15 SMF DATA SETS 209
      NAME          VOLSER    SIZE (BLKS)  FULL STATUS
      P-SYS1.MAN1  Z7SYS1    7200        5    ACTIVE
      S-SYS1.MAN2  Z7SYS1    1800        0    ALTERNATE
      S-SYS1.MAN3  Z7SYS1    1800        0    ALTERNATE
```

After the switch command, the display SMF response shows the active data set is now the first alternate, SYS1.MAN2:

```
RESPONSE=ADCD
IEE949I 16.02.08 SMF DATA SETS 252
      NAME          VOLSER    SIZE (BLKS)  %FULL  STATUS
      P-SYS1.MAN1  Z7SYS1    7200        7    DUMP REQUIRED
      S-SYS1.MAN2  Z7SYS1    1800        2    ACTIVE
      S-SYS1.MAN3  Z7SYS1    1800        0    ALTERNATE
```

12.7.9 Running the SMF Exit Manually

Use this command to run the same JCL that is executed when the SMF exit occurs, when an SMF data set is full.

```
START NSQSMF,MAN=smf_dataset
```

where `smf_dataset` is the name of a full data set, such as **SYS1.MAN1**

The NSQSMF JCL will automatically be found in data set USER.PROCLIB. The screen will show this top line, which can be ignored:

```
HQX7720 ----- SDSF PRIMARY OPTION MENU -- NO RESPONSE RECEIVED
```

Enter LOG, find the last NSQSMF job and display it to check that the job ran properly.

Alternately, use the **SWITCH SMF** command.

12.8 Useful WMQ Commands

These commands are run from the SDSF panel

To start channel initiator:

1. Go to SDSF panel: =m.5
2. Enter **`/%qmgr start chinit`**

To display channel initiator:

1. Go to SDSF panel: =m.5
2. Enter **`/%qmgr display chinit`**

To start queue manager:

1. Go to SDSF panel: =m.5
2. Enter **`/%qmgr start qmgr`**

To display queue manager:

1. Go to SDSF panel: =m.5
2. Enter **`/%qmgr display qmgr`**

To start listener:

1. Go to SDSF panel: =m.5
2. Enter **`/%qmgr start listener port_nbr`** (for example: 1418)

12.9 Utilities

12.9.1 How to Process an SMF Dump File

This procedure takes an existing SMF dump file and processes it to form PCF accounting and statistics messages.

1. Go to *naslib.JCL*
2. Find RUNZAS and enter option **SUB** to submit the job. Leave the `-t` trace option enabled.
3. Go to SDSF panel, enter ST (status of jobs), find and select job NSQZAS.
4. Examine SYSOUT file for error messages
5. Examine SYSPRINT trace file. First few lines should look like this:

```
NSQZAS, z/OS Accounting and Statistics Collector, version 1.1.1 Nov
8 2007 starting...
```

```
z/OS Accounting and Statistics Collector start-up parameters:
STARTUP OPTIONS:
```

```
  QMgr List                V22F CSQ6 CSQ7
  Collector QMgr           CSQ7
  MQI Accounting           on
  Queue Accounting         on
  MQI Statistics           on
  Queue Statistics         on
  Channel Statistics       on
  Statistics Interval      300
  secs Trace functions     on

  Dump SMF records        on

  Dump output to:         stdout

Tue Dec 4 17:01:28 2007    main: after initialization
Tue Dec 4 17:01:28 2007    zasConnQMgr: entry: CSQ7
Tue Dec 4 17:01:28 2007    zasConnQMgr: exit, hConn 1dc97ce0
```

Last 3 lines should like this:

```
Tue Dec 4 17:01:29 2007    zasFreeStatsBlocks:   Free   QStats
                           block malloc'd memory

Tue Dec 4 17:01:29 2007    main: close SMF input file and dump
                           file

Tue Dec 4 17:01:29 2007    main: exit
```

12.9.2 How to Use AMQSMON to Print Formatted Accounting and Statistics PCF Messages

When an SMF data set is dumped and processed during production, generated PCF messages are enqueued to alias queues NASTEL.ADMIN.STATISTICS.QUEUE and NASTEL.ADMIN.ACCOUNTING.QUEUE and sent to a workgroup server by the publisher or agent. In order to examine PCF messages before they leave the z/OS system, the queues have to be changed to local queues. Then utility `amqsmon` (the customized version that reads from those queues) can be run to analyze and print the formatted PCF messages.

1. Go to ISPF Primary Option Menu.
2. Go to IBM WebSphere MQ for z/OS – Main Menu. Enter on command either: M followed by 12 on the IBM Products Panel; or =m.12
3. Next to queue names NASTEL.ADMIN.STATISTICS.QUEUE and NASTEL.ADMIN.ACCOUNTING.QUEUE, enter Action 1, Object QALIAS, Name NAS*.
4. On the next screen, next to the queue names, enter option 4 followed by 1 in order to delete the definitions.
5. On the Main Menu, use option 2, Define Like, with Object QLOCAL to create the two local queues with the names defined in step 3 above.
6. Run the JCL's to dump and process an SMF data set. To run `amqsmon`:
 - a. Open for edit data set `naslib.JCL.AMQSMON`. Edit the queue manager name `-m CSQ7` to suit the test. Optionally append `'-b'` to the PARAM line to browse the queue without deleting the PCF message.
 - b. Edit the PARAM line to format the MQI Accounting messages and submit:
`PARM='-m CSQ7 -t accounting -a'`
 - c. Edit the PARAM line to format the Queue Accounting messages and submit:
`PARM='-m CSQ7 -t accounting -q'`
 - d. Edit the PARAM line to format the MQI Statistics messages and submit:
`PARM='-m CSQ7 -t statistics -a'`
 - e. Edit the PARAM line to format the Queue Statistics messages and submit:
`PARM='-m CSQ7 -t statistics -q'`
 - f. Edit the PARAM line to format the Channel Statistics messages and submit:
`PARM='-m CSQ7 -t statistics -c'`
7. To view the output, go to the SDSF panel: =m.5
8. Enter ST (status of jobs) and page down to the AMQSMON jobs; there should be 5 of them. Select each job in turn with the '?' option and then select the SYSPRINT log with the s' option. They should look like the following samples:

```
MonitoringType: MQIAccounting QueueManager: 'V22F' IntervalStartDate:
'2000-11-24'

IntervalStartTime: '10.03.35'
IntervalEndDate: '2000-11-24'
IntervalEndTime: '10.03.50'
CommandLevel: 0
ConnectionId:      x'c4e4d4d4e840404040404083f285f98484f4f6f2f0f0f7f5f5f083'
SeqNumber: 0

ApplicationName: 'PAICEB' ApplicationPid: 0
ApplicationTid: 0 UserId: 'PAICE'

OpenCount: [0, 2000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 2000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [0, 1000]
Put1Count: [0, 0]
PutBytes: [0, 1000000]
GetCount: [0, 2000]
GetBytes: [0, 1000000]
CommitCount: 2000
BackCount: 0

MonitoringType: QueueAccounting QueueManager: 'V22F' IntervalStartDate:
'2000-11-24'

IntervalStartTime: '10.03.35'
IntervalEndDate: '2000-11-24'
IntervalEndTime: '10.03.50'
CommandLevel: 0
ConnectionId:      x'c4e4d4d4e840404040404083f285f98484f4f6f2f0f0f7f5f5f083'
SeqNumber: 0

ApplicationName: 'PAICEB' ApplicationPid: 0
ApplicationTid: 0 UserId: 'PAICE'

ObjectCount: 2
QueueAccounting: 0 QueueName: 'CP0000'

. . .
```

MonitoringType: **MQIStatistics** QueueManager: 'V22F' IntervalStartDate: '2000-11-24'

IntervalStartTime: '14.54.13'

IntervalEndDate: '2000-11-24'

IntervalEndTime: '14.54.31'

CommandLevel: 0

OpenCount: [0, 2001, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

CloseCount: [0, 2000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

InqCou [0, 0, 0, 0, 0,

SetCou [0, 0, 0, 0, 0,

PutCou [1000]

Put1Count: [0, 1000]

PutBytes: [0, 2000000]

GetCount: [0, 4001]

GetBytes: [0, 2000000]

BrowseCount: [0, 0]

CommitCount: 3000

BackCount: 0

MonitoringType: **QueueStatistics** QueueManager: 'V22F' IntervalStartDate: '2000-11-24'

IntervalStartTime: '14.54.13'

IntervalEndDate: '2000-11-24'

IntervalEndTime: '14.54.31'

CommandLevel: 0

ObjectCount: 2

QueueStatistics: 0 QueueName: 'CP0000' QueueType: Local QMaxDepth: 0

PutCount: [0, 0]

Put1Count: [0, 1000]

PutBytes: [0, 1000000]

GetCount: [0, 2000]

GetBytes: [0, 1000000]

BrowseCount: [0, 0]

QueueStatistics: 1 QueueName: 'SERVER' QueueType: Local QMaxDepth: 0

PutCount: [0, 1000]

Put1Count: [0, 0]

PutBytes: [0, 1000000]

GetCount: [0, 2001]

GetBytes: [0, 1000000]

BrowseCount: [0, 0]

```
MonitoringType: ChannelStatistics
  DATA A:
  MonitoringType: ChannelStatistics
  QueueManager: 'CSQ7'
  IntervalStartDate: '2008-03-11'
  IntervalStartTime: '17.18.29'
  IntervalEndDate: '2008-03-11'
  IntervalEndTime: '17.21.38'
  CommandLevel: 600
  ObjectCount: 2
ChannelStatistics: 0
  ChannelName: 'CSQ7_TO_CSQ6'
  ChannelType: Sender
  ConnName: '11.0.0.55(1416)'
  MsgCount: 71
  TotalBytes: 30459
ChannelStatistics: 1
  ChannelName: 'CSQ6_TO_CSQ7'
  ChannelType: Receiver
  ConnName: '11.0.0.55'
  MsgCount: 2
TotalBytes: 404
```

12.9.3 How to Manually Dump an SMF Data Set

1. Use SDSF command `/D SMF` to display the SMF data sets.
2. Pick one that is in state `READY TO BE DUMPED`, for example: `MAN1`
3. Open JCL `naslib.JCL(NSQSMFDP)` for edit.
4. Edit the 3 lines where `MANnn` and `MQRECSnn` are located, changing `nn` to the chosen SMF data set number 1, 2, etc.
5. Submit the job.

The output in the log should look like

```
this:  FA010I      SMF      DUMP
PARAMETERS FA010I END(2400)
--      DEFAULT      FA010I
START(0000) -- DEFAULT

FA010I  DATE(1900000,2099366)  --
DEFAULT FA010I  OUTDD(DUMPOUT,TYPE
116)      --      SYSIN      FA010I
INDD(DUMPIN,OPTIONS(ALL))  --  SYSIN
FA020I DUMPOUT --      HBC.MP1B.MQRECS3
FA020I DUMPIN  --  SYS1.MAN3
```

FA018I SMF DATA SET DUMPIN HAS BEEN CLEARED.

SUMMARY ACTIVITY REPORT

RECORD	RECORDS	PERCENT	AVG. RECORD	END DATE-TIME
TYPE	READ	OF TOTAL	LENGTH	MIN. RECORD MA
				LENGTH
2	0			
3	0			
4	3	.03%	255.00	247
5	1	.01%	145.00	145
99	10,133	96.62%	633.63	182
116	1	.01%	6,500.00	6,500
TOTAL	10,488	100%	621.42	91
NUMBER OF RECORDS IN ERROR		0		

12.9.4 How to Manually Process an Existing SMF Dump File

1. On SDSF panel, use command `/D SMF` to display the SMF data sets.
2. Select `naslib.JCL.RUNZAS`, used to manually start program NSQZAS processing of an SMF dump. (`naslib` is a system high level qualifier, such as M6WMQ)
3. Edit the `nm` value on the SMFIN line then SUBMIT the job.
4. Check the output files `naslib.SYSPRINT` and `naslib.SMFDUMP`.

The former is a trace of the NSQZAS program; the latter contains a hex-ascii dump of the SMF records, assuming that the option `-ds` (dumpfile to stdout) was not used when starting the job in step 2. With `-ds`, the hex-ascii dump appears together with the trace in the SYSPRINT file.

In the trace file, all PCF message traces are preceded by a line containing the words **“dump and print”**.

Other key words to look for when searching for a trace or dump of a PCF message:

- MQCMD_ACCOUNTING_MQI
- MQCMD_ACCOUNTING_Q
- MQCMD_STATISTICS_MQI
- MQCMD_STATISTICS_Q
- MQCMD_STATISTICS_CHANNEL

To check the SMFDUMP file:

1. Enter the `s` option next to the SMFDUMP data set name.
2. If the display shows “Used cylinders: 0”, then an empty file was created because the `-ds` option was selected in the RUNZAS job.
3. If the Used Cylinder count is non-0, then use the `'e'` option to open the file.
4. The SMF records are printed in pure hex-ascii and then followed by a formatted print, displaying the field names, description (sometimes), memory address, hex value, and as a string if applicable. The dump will have lines similar to these:

```
000033 --SMFHEADER---H-E-X---P-R-I-N-T----
000034 Address      = 1DC63868
000035 00000000 : 5E740061 725E0107 317FE2E8 E2F1C3E2 ;.../.;..."SYS1CS>
000036 00000010 : D8F70001 F6F0F000 000018E4 00800001 Q7..600....U....>
000037 --SMFHEADER---F-O-R-M-A-T-T-E-D----
000038 SMFRECLFG : (addr) '1DC63868' (hex) '5E' : (disp) ';'
000039 SMFRECRTY : (addr) '1DC63869' (hex) '74' : (disp) '.'
..
000081 --QWHS-HEADER---H-E-X---P-R-I-N-T----
000082 Address = 1DC65148
000083 00000000 : 002401F7 00190410 1BFF8C98 C3E2D8F7 <...7.....qCSQ7>
000084 00000010 : C17E7796 E9F7E200 0001111B 0001111B <A=.oZ7S.....>
000085 00000020 : 00000002 <....>
000086 --QWHS-HEADER---F-O-R-M-A-T-T-E-D----
000087 qwhsnsda (nbr self-defining sections): 4 (hex), 4 (dec)
000088 qwhsssid (subsystem name = QMgr) : (addr) '1DC65154' (hex) 'C3E2 000089
qwhcaid (user ID for the OS/390 job) : (addr) '1DC65170' (hex) 'E2E8 000090 qwhccv
(thread cross ref) : (addr) '1DC65178' (hex) 'F2F4 000091 qwhccn (connection name):
(addr) '1DC65184' (hex) 'C3E2
```

```

000107 --W-T-I-D---H-E-X---P-R-I-N-T----
000108 Address = 1DC638A0
000109 00000000 : F70000D0 E6E3C9C4 00000002 D5E2D8E2 <7..}WTID....NSQS>
000110 00000010 : D4C64040 E2E3C1D9 E3F24040 40404040 <MF START2 >
000111 00000020 : 40404040 00000000 00000000 40404040 < ..... >
000112 00000030 : 40404040 40404040 40404040 40404040 < >
000113 00000040 : 40404040 40404040 C17E7795 478D0001 < A=.n....>
...
000122 --W-T-I-D---F-O-R-M-A-T-T-E-D----
000123 wtidshex = fffff700
000124 wtidlen = 208
000125 wtideyec = WTID
000126 wtidatyp = 0002
000127 wtidccn: (addr) '1DC638AC' (hex) 'D5E2D8E2D4C64040' : (disp) 'NSQSMF
000128 wtidopid : (addr) '1DC638B4' (hex) 'E2E3C1D9E3F24040' : (disp) 'START2
000129 wtidnid: (addr) '1DC638BC' (hex) '40404040404040400000000000000000' :
000130 wtidcori : (addr) '1DC638CC' (hex) '404040404040404040404040404040' : (disp) '
...
000148 --W-T-A-S---H-E-X---P-R-I-N-T----
000149 Address = 1DC63970
000150 00000000 : F70102D8 E6E3C1E2 C17E7795 47D53700 <7..QWTASA=.n.N..>
000151 00000010 : 54DCAB28 54DB5038 1BFF8CF8 00000000 <.....&....8....>
000152 00000020 : 000003F4 00000000 30800000 00000008 <...4.....>
000153 00000030 : 00000000 00000000 00000000 00000000 <.....>
...
000196 --W-T-A-S---F-O-R-M-A-T-T-E-D----
000197 wtasshex = fffff701
000198 wtaslen= 0728
000199 wtaseyec = WTAS
000200 wtasstrt : (addr) '1DC63978' (hex) 'C17E779547D53700' : (disp) 'A=.n.N..
000201 zasSTCKToStr: IN: 1lStck <-4504031094284929280>, OUT: Date <2007-11-13>,
...
000289 --W-Q-----H-E-X---P-R-I-N-T----
000290 Address = 1DC63C48
000291 00000000 : F70202A0 E6D8E2E3 00000004 54D4F578 <7...WQST.....M5.>
000292 00000010 : C17E7795 47D53700 54DCAB28 54DB5038 <A=.n.N.....&.>
000293 00000020 : E2E8E2E3 C5D44BC3 D6D4D4C1 D5C44BD9 <SYSTEM.COMMAND.R>
000294 00000030 : C5D7D3E8 4BD4D6C4 C5D34040 40404040 <REPLY.MODEL >
000295 00000040 : 40404040 40404040 40404040 40404040 < >
...
000333 --W-Q-----F-O-R-M-A-T-T-E-D----
000334 wqid = fffff702
000335 wqll = 0672
000336 wqeye = WQST
000337 wqver = 00000004
000338 correl : (addr) '1DC63C58' (hex) 'C17E779547D5370054DCAB2854DB5038' :
000339 objname = SYSTEM.COMMAND.REPLY.MODEL
000340 basename = CSQ.C17E77964D480100
000341 opentime : (addr) '1DC63CC8' (hex) 'C17E77964E878500' : (disp) 'A=.o+ge.

```

12.9.5 How to Print Formatted SMF Records

1. On SDSF panel, use command **/D SMF** to display the SMF data sets.
2. Select *naslib.JCL(NSQSMFDP)* to dump a specific SMF data set. Edit the data set related lines and submit the job.
3. Select *naslib.JCL(RUNCDUMP)* to print the contents of type 116 records in a dump-like format, in both EBCDIC and hex. (See format in previous section). Individual files are created for each of the following record subtypes:
 - **SMFHDR** – SMF record headers
 - **QWHS** – Common WebSphere MQ header
 - **WTID** – Class 3 accounting – Thread identification data
 - **WTAS** – Class 3 accounting – General thread accounting data
 - **WQ** – Class 3 accounting – Queue specific accounting data
4. Edit the nn value and/or data set name on the SMFIN line and SUBMIT the job.
5. Check the SYSPRINT log files for job CDUMP. The format is the same as shown in [Appendix F](#), except that the print for all like-records (e.g. all WTID) are in one file.

12.10 SMF Exit JCL Listing

This JCL is in USER.PROCLIB(NSQSMF).

```
//NSQSMF    PROC MAN='SYS1.MAN1',CNTL=MQSMF
//*-----
//* CALLED BY: NSQXU29 assembler exit program
//* LOCATION:  member of a procedure library, e.g. USER.PROCLIB
//* INPUT:     MAN = Name of data set that just filled
//*           Default: SYS1.MAN1
//*           CNTL = Name of file containing SMF dump options
//*           Default: MQSMF
//* STEPS:
//*
//* 1. NSQSMFDP - Dump all SMF records to temporary file
//* 2. 1-2.n NSQZAS1,NSQZAS2,...
//*    - Analyze and report on SMF 116 from temporary file
//*      Need one NSQZAS job step per qmgr. Copy NSQZAS job
//*      lines, rename jobs NSQZAS1, NSQZAS2,....
//*      Change the -m and -mc values to a qmgr.
//* 3. NSQGDG - Copy temporary file to GDG for later reuse
//*-----
//* CUSTOMIZATION:
//*   Change ++hlq++ to M6WMQ install data set
//*   Change ++mqser++ to IBM WMQ data set
//*   Add NSQZAS job sections, one per qmgr being monitored
//*   Set PARM options for each NSQZAS job
//*-----
//* Dump a filled SMF data set:
//*
//* The NASTEL.MQSMF.CNTL(MQSMF) data set contains:
//*   INDD(DUMPIN,OPTIONS(DUMP|ALL))
//*   OUTDD(DUMPOUT,TYPE(116))
//* You can change the OPTIONS to ALL, to clear all SMF records.
//* Only the last SMF exit in the chain should do this.
//*
//* &&NSQSMF = temp file created automatically for the job duration
//*-----
//NSQSMFDP EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN   DD DSN=&MAN,DISP=SHR
//DUMPOUT  DD DSN=&&NSQSMF,
//          DISP=(,PASS),
//          SPACE=(TRK,(15,15),RLSE),UNIT=SYSDA,
//          DCB=(RECFM=VBS,BLKSIZE=32760)
//SYSIN    DD DSN=NASTEL.MQSMF.CNTL(&CNTL),DISP=SHR
//*-----
//* Execute if prev step Return Code <= 4
//* 0M = give all resources (memory) needed
//*-----
//* -m List of qmgrs for which to collect acct/stats msgs
//* -mc Qmgr to use by the collector for writing the acct/
//*     stats messages to forward to M6WMQ workgroup server
//*
//* Note: -m and -mc values should be the same. To process multiple
//*       qmgrs, one NSQZAS job per qmgr is required. That is,
//*       replicate lines NSQZAS through SMFIN and edit the copied
//*       lines as required. All the NSQZAS jobs will process
//*       from the same temp SMF dump file.
```

```

/**
/** -qa Collect Queue Accounting on|off
/** -ma Collect MQI Accounting on|off
/** -ms Collect MQI Statistics on|off
/** -qs Collect Queue Statistics on|off
/** -cs Collect Channel Statistics on|off
/** -st n_secs Statistics time: aggregate over n_secs. Default: 1800.
/** -t Trace functions (stdout is file SYSPRINT)
/** -d Dump SMF records in ascii/hex
/** -ds Dump output to stdout. Default: file SMFDUMP.
/**
/**-----
/**NSQZAS EXEC PGM=NSQZAS,COND=(4,LT),REGION=0M,
/** PARM='-m CSQ6 -mc CSQ6 -qa on -ma on
/** -ms on -qs on -cs on -st 600 -t -d -ds'
/**STEPLIB DD DISP=SHR,DSN=++hlq++.LOAD
/** DD DISP=SHR,DSN=++mqser++.SCSQAUTH
/** DD DISP=SHR,DSN=++mqser++.SCSQANLE
/** DD DISP=SHR,DSN=++mqser++.SCSQLOAD
/**SYSPRINT DD SYSOUT=*,DCB=(LRECL=132,RECFM=F)
/**SMFDUMP DD SYSOUT=*,DCB=(LRECL=233,RECFM=F,BLKSIZE=233)
/**SMFIN DD DSN=&&NSQSMF,DISP=(OLD,PASS)
/**
/**-----
/** Copy SMF dump to file within GDG to preserve data for later reuse
/** GDG = Generation Data Group
/** DATA(0) = is replaced with latest generation version, e.g. G0002.V00
/**-----
/**NSQGDG EXEC PGM=IEBGENER
/**SYSPRINT DD SYSOUT=*
/**SYSUT1 DD DSN=&&NSQSMF,DISP=(OLD,DELETE)
/**SYSUT2 DD DSN=NASTEL.MQSMF.DATA(+1),
/** DISP=(NEW,CATLG),
/** SPACE=(TRK,(15,15),RLSE),UNIT=SYSDA,
/** DCB=(RECFM=VBS,BLKSIZE=32760)
/**SYSIN DD DUMMY

```

Chapter 12A: nsqprid Utility

12A.1 Description

Utility program nsqprid creates a specific user/password for a workgroup server and writes the credentials to the mqgroup.ini file.

12A.2 Command

Use the following command to run the utility:

```
C:\nastel\apwmq\bin>nsqprid
```

12A.3 Sample Output

```
C:\nastel\apwmq\bin>nsqprid
```

```
(C)Copyright 1995 - 2015, Nastel Technologies Inc.
```

```
Reproduction, adaptation, or distribution without prior  
written permission is prohibited, except as allowed under  
the copyright laws.
```

```
Date: Thu Oct 08 14:40:48 2015
```

```
NSQNSQ Processing ID Editor V6.5.0005, Build Jul 3 2015
```

```
Enter Workgroup Name [MQM]: MQMPRD
```

```
Enter Processing ID User Name: JohnM
```

```
Enter Processing ID Password: *****
```

```
Reenter Processing ID Password: *****
```

```
Wrote Processing ID credentials to:
```

```
C:\nastel\apwmq\config\groups\mqgroup.ini
```

```
(0028140)
```

```
Jane Doe (manager)
```

```
2015-10-08 14:48
```

```
The credentials lines written to the end of file mqgroup.ini:
```

```
MQMPRD::PROCIDUser=JohnM
```

```
MQMPRD::PROCIDPwd=jg1jx7TePOnNVE0Rggy8C2==
```

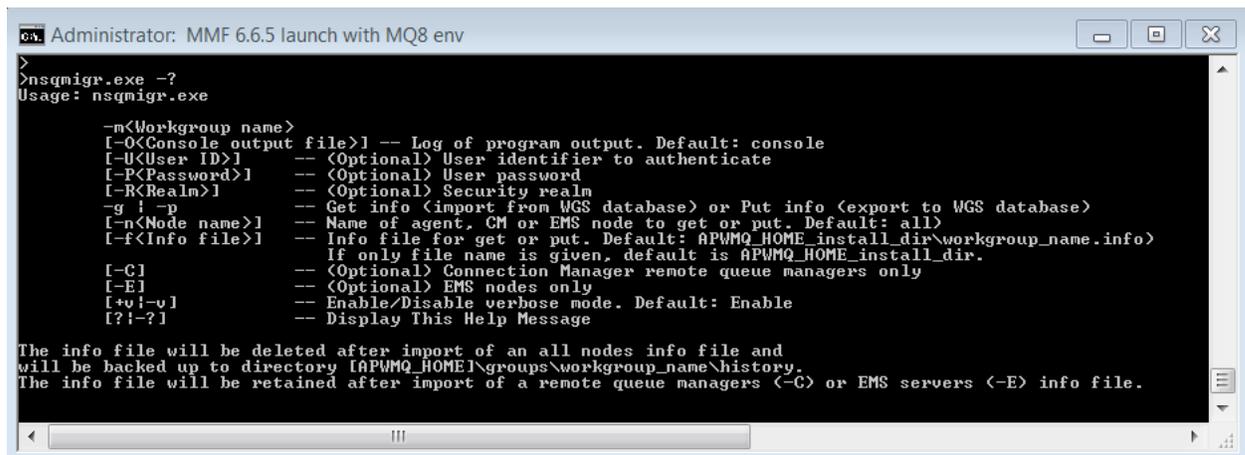
Chapter 12B: nsqmigr Utility

The nsqmigr utility is used to export/import workgroup server and node information in bulk. nsqmigr connects to the workgroup server and either inquires (exports) or updates (imports) database data to or from a .info file. Optional parameters can be set to export/import remote queue manager or remote EMS server information. See the sections immediately below for instructions on how to restore or export data using nsqmigr.

12B.1 Parameters

Enter the following command to view nsqmigr's command line options.

```
nsqmigr -?
```



```

Administrator: MMF 6.6.5 launch with MQ8 env
>nsqmigr.exe -?
Usage: nsqmigr.exe

-m<Workgroup name>
[-O<Console output file>] -- Log of program output. Default: console
[-U<User ID>] -- (Optional) User identifier to authenticate
[-P<Password>] -- (Optional) User password
[-R<Realm>] -- (Optional) Security realm
-g | -p -- Get info (import from WGS database) or Put info (export to WGS database)
[-n<Node name>] -- Name of agent, CM or EMS node to get or put. Default: all
[-f<Info file>] -- Info file for get or put. Default: APWMQ_HOME_install_dir\workgroup_name.info
                IF only file name is given, default is APWMQ_HOME_install_dir.
[-C] -- (Optional) Connection Manager remote queue managers only
[-E] -- (Optional) EMS nodes only
[+v|-v] -- Enable/Disable verbose mode. Default: Enable
[?!-?!] -- Display This Help Message

The info file will be deleted after import of an all nodes info file and
will be backed up to directory [APWMQ_HOME]\groups\workgroup_name\history.
The info file will be retained after import of a remote queue managers (-C) or EMS servers (-E) info file.

```

Figure 12B.1 – nsqmigr Utility

	NOTE: Parameters are case sensitive.
---	---

12B.2 Get / Put File

nsqmigr uses a *.info file when importing/exporting data. Please note that if the -f parameter is not used, nsqmigr will then use the default file name, *workgroup_name.info*, when exporting and importing.

12B.3 Recreating a WGS Database

To recreate a WGS database, you will need to:

- 1) **Create a backup:** Use the nsqmigr utility to create the *.info backup file. Run a command similar to the following to create this file:

```
nsqmigr -g -mworkgroup_name -ffilename.info
```

The default export directory is [APWMQ_HOME]/config/groups
- 2) **Restore:** Place the updated info file in [APWMQ_HOME]/config/groups if it is not there already. When the WGS is restarted, the file will automatically load and the database will be created. Please note that the *.info file is automatically deleted after it is successfully imported. It is first backed up in [APWMQ_HOME]/config/groups/workgroup/history.

12B.3.1 Imported / Exported Fields

When using `-g` or `-p` without other parameters, nsqmigr will read/write the following fields if supplied:

<i>Workgroup Server Level</i>	<i>Node Level</i>	<i>Node Level</i>	<i>Node Level</i>
DiscoveryTime	NodeName	AuthInfoMask	QMgrLogSec
BroadcastIPNetwork	UseDNS	LsrMask	QMgrLogFileSz
BroadcastIPPortNo	HostName	SvcMask	QMgrLogBuffSz
TraceMode	IPAddress	TopicMask	AgentTrace
Description	IPPort	SubMask	Authorization
MaxNodes	Platform	ClntMask	NodeOSInfo
MaxClients	Discovery	Description	NodeCPUCount
MaxQmgrs	CmdLimit	NodeType	NodeMIPCount
AuthRecMode	Timeout	MaxWtTime	
MaxBrowse	HBInterval	TcpQLim	
PortNo	Update	RetryCount	
EvtDepth	Manage	Uid2Upper	
AgentTrace	QmgrMask	BufferSize	
Authorization	LocQueMask	TcpNoDelay	
AcctDepth	AIQueMask	ReuseAddr	
StatDepth	RmtQueMask	ObjNmlsDisc	
Refresh	MdlQueMask	NodePing	
ClntCmdLimit	ChlMask	DnsName	
Uid2Upper	ProcMask	QMgrLogPath	
AdtinfDepth	NmlMask	QMgrLogMethod	
Alias	ClusQmgrMask	QMgrLogPri	

12.B.3.2 Example Get/Put WGS nodes.info File

```

NASTELWGS::DiscoveryTime=1440
NASTELWGS::BroadcastIPNetwork=0.0.0.0
NASTELWGS::BroadcastIPPortNo=5010
NASTELWGS::TraceMode=0
NASTELWGS::MaxNodes=100
NASTELWGS::MaxClients=200
NASTELWGS::MaxQmgrs=50
NASTELWGS::AuthRecMod=0
NASTELWGS::MaxBrowse=10000
NASTELWGS::PortNo=4020
NASTELWGS::EvtDepth=1000
NASTELWGS::AgentTrace=0
NASTELWGS::Authorization=0
NASTELWGS::AcctDepth=5000
NASTELWGS::StatDepth=500000
NASTELWGS::Refresh=10
NASTELWGS::CIntCmdLimit=15000
NASTELWGS::Uid2Upper=1
NASTELWGS::AdtInfDepth=0

NodeName=PC2
UseDNS=0
HostName=PC2
IPAddress=127.0.0.1
IPPort=5010
Platform=WINDOWS
Discovery=720
CmdLimit=5000
Timeout=60
HBInterval=1
Update=30
Manage=1
NodeType=1
MaxWtTime=1000
TcpQLim=10
RetryCount=1
Uid2Upper=1
BufferSize=64
TcpNoDelay=0
ReuseAddr=1
ObjNmIsDisc=0
NodePing=0
QMgrLogMethod=0
QMgrLogPri=0

QMgrLogSec=0
QMgrLogFileSz=0
QMgrLogBuffSz=0
AgentTrace=0
Authorization=0
NodeOSInfo=Microsoft Windows Vista
Business Edition, 64-bit, Version 6.2.9200
NodeCPUCount=8
NodeMIPCount=0

NodeName=REMOTE_QMGRS
UseDNS=0
HostName=PC2
IPAddress=127.0.0.1
IPPort=5020
Platform=WINDOWS
Discovery=720
CmdLimit=5000
Timeout=60
HBInterval=1
Update=30
Manage=1
NodeType=2
MaxWtTime=1000
TcpQLim=10
RetryCount=1
Uid2Upper=1
BufferSize=64
TcpNoDelay=0
ReuseAddr=1
ObjNmIsDisc=0
NodePing=0
QMgrLogMethod=0
QMgrLogPri=0
QMgrLogSec=0
QMgrLogFileSz=0
QMgrLogBuffSz=0
AgentTrace=0
Authorization=0
NodeOSInfo=Microsoft Windows Vista
Business Edition, 64-bit, Version 6.2.9200
NodeCPUCount=8
NodeMIPCount=0

NodeName=ZOS
UseDNS=0
IPAddress=172.29.122.132
IPPort=5010
Platform=MVS
Discovery=720
CmdLimit=5000
Timeout=90
HBInterval=1
Update=30
Manage=0
NodeType=1
MaxWtTime=10000
TcpQLim=10
RetryCount=1
Uid2Upper=1
BufferSize=64
TcpNoDelay=0
ReuseAddr=0
ObjNmIsDisc=0
NodePing=0
QMgrLogMethod=0
QMgrLogPri=0
QMgrLogSec=0
QMgrLogFileSz=0
QMgrLogBuffSz=0
AgentTrace=0
Authorization=0
NodeCPUCount=1
NodeMIPCount=1960

RemoteQmgrName=CITI_QM
ParentNode=ZOS

```

Figure 12.B.3.2-A. Example Get / Put WGS File

12B.4 Remote Queue Managers

Use `-p` or `-g` with the `-C` parameter to import/export remote queue managers, which are managed by a connection manager (CM, nsqcm). nsqmgr will only include the remote queue managers.

Note: The CM nodes are included in the info file created when all nodes are exported, by using `-g` without `-n,-C,-E`.

nsqmgr will read/write the following fields if supplied:

```

RemoteQmgrName
ParentNode
ConnectionName
ChannelName
SSLKeyRepos
SSLCrypHrdw
SSLCipherSpec
ConvertMsg
SecExit
SecExitData
Username
PW

```

12B.4.1 Bulk Importing Remote Queue Managers (-p parameter)

Perform the following to bulk import remote queue managers:

- 1) Either
 - a. manually create the *filename.info* file, such as *CM_RemQmgrs.info*, to define each remote queue manager being imported. Add the fields listed immediately above for each remote queue manager; or
 - b. run command `nsqmigr -mworkgroup -g -C -ffilename.info` to export the remote queue managers from one or more workgroup servers, and then merge the files into one.
- 2) Place the file in `[APWMQ_HOME]/config/groups`. Please note that the **.info* file is automatically backed up in `[APWMQ_HOME]/config/groups/workgroup/history` after importing and is not deleted.
- 3) Run a command similar to the following to import the file, perhaps into another workgroup. The `-C` parameter must be uppercase.


```
nsqmigr -mworkgroup -p -C -ffilename.info
```
- 4) The remote queue managers are now imported in to the WGS database. After they are imported, the import file is backed up in the `history` directory.

12B.4.2 Bulk Exporting Remote Queue Managers (-g parameter)

Perform the following to bulk export remote queue managers:

- 1) Use a command similar to the following to create a file containing a list of all remote queue managers. The `-C` parameter must be uppercase.

```
nsqmigr -g -C -ffilename.info
```

- 2) The *filename.info* file is now created. For each remote queue manager restored, the following two messages will appear on the console:

```
Creating Remote Queue Manager qmgr_name...
Remote Queue Manager created successfully!
```

The info file, such as *MQM.info*, will list all remote queue managers and their fields, as in Figure 12B.4.3-A below.

The default export directory is `[APWMQ_HOME]/config/groups` and the default output info file is `workgroup_name.info`. Both can be overridden by using the `-f` parameter.

12B.4.3 Example Get/Put Remote Queue Managers Commands and Info File

The following is an example of `-get` and `-put` operations and a sample output remote queue managers file, *remote_qmgrs.info*.

- **Using the `-p` command:** You need to create this file manually or have one from a prior export before importing it. The following command would put, or import, the data into the WGS database:

```
C:\nastel\apwmq\bin>nsqmigr -mWGS38 -p -C
-fC:\nastel\apwmq\config\groups\remote_qmgrs.info
```

- **Using the -g command:** This command would get, or export, the remote queue managers from the WGS database to file `remote_qmgrs.info`:
`C:\nastel\apwmq\bin>nsqmigr -mWGS38 -g -C
-fC:\nastel\apwmq\config\groups\remote_qmgrs.info`

```
remote_qmgrs.info - Notepad
File Edit Format View Help
ParentNode=REMOTE_QMGRS
ConnectionName=1xx.0.0.xxx(1500)
ChannelName=IBM.SVR.CONN
SSLKeyRepos=
SSLCrypHrdw=
SSLCipherSpec=
ConvertMsg=
SecExit=
SecExitData=
ConvertMsg=0
Username=user
PW=gGTZtBxqvHDIByWu0xsTkA==

RemoteQmgrName=ORIS
ParentNode=REMOTE_QMGRS
ConnectionName=11x.0.0.xxx(1502)
ChannelName=ORIS.SVR.CONN
SSLKeyRepos=
SSLCrypHrdw=
SSLCipherSpec=
ConvertMsg=
SecExit=
SecExitData=
ConvertMsg=0
Username=user
PW=gGTZtBxqvHDIByWu0xsTkA==
```

Figure 12B.4.3-A. Example Remote Queue Managers Info File

12B.5 Remote EMS Servers

Use `-p` or `-g` with the `-E` parameter to import/export Remote EMS Servers (`nsqcmems.jar`). `nsqmigr` will only include nodes that are Remote EMS Nodes. `nsqmigr` will read/write the following fields if supplied:

```
EMS::ManagerName
ParentNode
Server
SrvUrl
Username
Password
SSLTrustCertificate
SSLIdentity
SSLIssuer
SSLPwd
SSLKey
SSLCiphers
SSLVendor
SSLHostname
SSLVerifyHostName
SSLVerifyHost
SSLTrace
SSLDebugTrace
```

12B.5.1 Bulk Importing Remote EMS Servers (-p parameter)

Perform the following to bulk import remote EMS servers:

- 1) Create the *.info file, such as remote_ems.info, to define each remote EMS server being imported. Add the fields listed above in section 12B.5 for each remote EMS server.
- 2) Place the file in [APWMQ_HOME]/config/groups. .
- 3) Run a command similar to the following to import the file. Please note that the -E parameter must be uppercase.

```
nsqmigr -p -E -f<filename>.info
```
- 4) The remote EMS servers are now imported in to the WGS database. The info file is automatically backed up in [APWMQ_HOME]/config/groups/workgroup/history and is not deleted.

12B.5.2 Bulk Exporting Remote EMS Servers (-g parameter)

Perform the following to bulk export remote EMS servers:

- 1) Use a command similar to the following to create a *.info containing a list of all remote EMS servers. Please note that the -E parameter must be uppercase.

```
nsqmigr -g -E -f<filename>.info
```
- 2) The *.info file is now created. It will list all remote EMS servers and the fields listed in section 12B.5 above. The default export directory is [APWMQ_HOME]/config/groups, but can be overwritten by using the -O parameter.

12B.5.3 Example Get/Put Remote EMS Server Commands and Info File

Below is an example of a Remote EMS Server file.

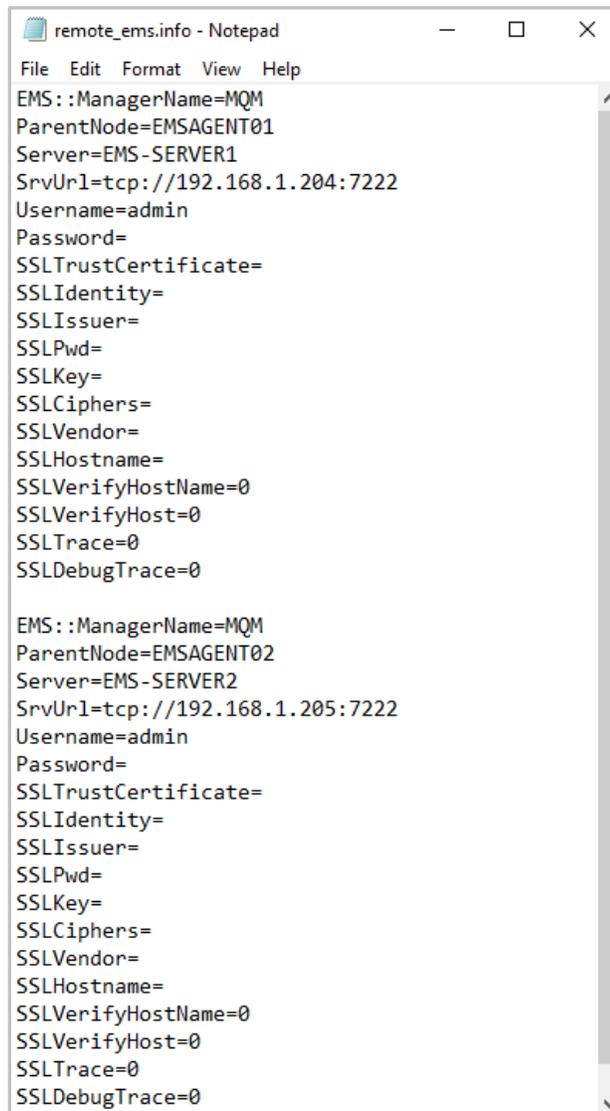
- **Using the -p command:** You need to create this file manually or have one from a prior export before importing it. The following command would put, or import, the data into the WGS database:

```
C:\nastel\apwmq\bin>nsqmigr -p -E  
-fC:\nastel\apwmq\config\groups\remote_ems.info
```

- **Using the -g command:** this file would be exported by nsqmigr. The following command would be run to generate it:

This command would get, or export, the remote EMS servers from the WGS database to the file remote_ems.info:

```
C:\nastel\apwmq\bin>nsqmigr -g -E  
-fC:\nastel\apwmq\config\groups\remote_ems.info
```



```
remote_ems.info - Notepad
File Edit Format View Help
EMS: :ManagerName=MQM
ParentNode=EMSAGENT01
Server=EMS-SERVER1
SrvUrl=tcp://192.168.1.204:7222
Username=admin
Password=
SSLTrustCertificate=
SSLIdentity=
SSLIssuer=
SSLPwd=
SSLKey=
SSLCiphers=
SSLVendor=
SSLHostname=
SSLVerifyHostName=0
SSLVerifyHost=0
SSLTrace=0
SSLDebugTrace=0

EMS: :ManagerName=MQM
ParentNode=EMSAGENT02
Server=EMS-SERVER2
SrvUrl=tcp://192.168.1.205:7222
Username=admin
Password=
SSLTrustCertificate=
SSLIdentity=
SSLIssuer=
SSLPwd=
SSLKey=
SSLCiphers=
SSLVendor=
SSLHostname=
SSLVerifyHostName=0
SSLVerifyHost=0
SSLTrace=0
SSLDebugTrace=0
```

Figure 12B.5.3-A. Example Remote EMS Servers Info File

This Page Intentionally Left Blank

Chapter 13: Troubleshooting

This chapter explains how to troubleshoot M6-WMQ effectively.

13.1 Identifying Common Problems

There are four types of common problems encountered when running M6-WMQ. They are listed in the *Common Problems Encountered Running M6-WMQ* table alongside the likely causes. Various potential problems and the solutions are addressed in this chapter.

Table 13-1. Common Problems Encountered Running M6-WMQ	
Type of Problem	Likely Cause
System component fails on start-up	Problem local to that component.
System component cannot interact	Communication problem, or problem local to one of the components.
No response to user actions	WebSphere MQ not receiving user actions, or mishandling them.
WebSphere MQ information is not collected	A workgroup server is not successfully connecting to or interacting with a WMQ Agent.

13.2 Troubleshooting a WMQ Agent

If you are having problems with a WMQ Agent, it may be malfunctioning, or it may lack sufficient privileges to carry out the tasks you are requesting.

Verifying WMQ Agent Operation

1. Use the `nsqcl` utility program. This utility prompts you for the location of the M6-WMQ Local Agent – enter host name and port number. Use conversational mode only. The utility prompts you for the Application Name – enter a random name if the host is running WebSphere MQ Level 2 systems. On WebSphere MQ Level 1 systems the Application Name should correspond to any available queue defined on the queue manager. The output of `nsqcl` command is in PCF format.
2. Once the connection has been established the utility prompts you for PCF commands. Before entering any PCF commands, do the following (example is for Windows environments):

```
C:\>nsqcl USER_NAME("john")
(C)Copyright 2007, Nistel Technologies Inc.
Reproduction, adaptation, or distribution without prior
written permission is prohibited, except as allowed under
the copyright laws.
```

```
Sizeof MSHDR = 136 Manager =
Sysname("Windows_XP")
NodeName("INSHAPE")
Release(" ")
```

```
Enter Application name: xxx
Enter host name (localhost): tag61
Enter service name: 5000
Mode of Connection ((S)ecure/(N)Secure) (N):
Connection has been established ok!
Enter Command Name:
```

3. Issue EXCMD_INQUIRE_Q_MGR_NAMES command to test the discovery of queue managers on the host system, both active and inactive. For the object name enter an *. A list of all the queue managers is returned. For example:

```

Enter Command Name: EXCMD_INQUIRE_Q_MGR_NAMES
Enter Object Name: *
Command (EXCMD_INQUIRE_Q_MGR_NAMES) has been issued
PCF RESPONSE FROM THE SERVER -->

PCF Header (MQCFH):
->Type(2), StrucLength(36), Version(1), Command(10007)
->MsgSeqNum(1), Control(1), ParameterCount(3)
->CompCode(0), Reason(0)

PCF String List (MQCFSL):
->Type(6), StrucLength(120), Parameter (2015), Count(2)
->StringLength(48), CharSetId(0)
->String List
{
    (1) - "NAS_QM"
    (2) - "NASOAM_QM"
};

PCF String (MQCFST):
->Type(4), StrucLength(24), Parameter(2015), CharSet(0)
->StringLength(1), String("*")

PCF Integer (MQCFIN):
->Type(3), StrucLength(16), Parameter (20119), Value(7)
THIS RECORD (LAST)

```

4. Issue EXCMD_MQ_OPEN command. For the object-name, enter the name of the remote queue manager whose operations you would like to test. This command opens a connection to the local queue manager on the remote machine. For example:

```

Enter Command Name: EXCMD_MQ_OPEN
Enter Object Name: NASOAM_QM
Command (EXCMD_MQ_OPEN) has been issued
PCF RESPONSE FROM THE SERVER -->

PCF Header (MQCFH):
->Type(2), StrucLength(36), Version(1), Command(10008)
->MsgSeqNum(1), Control(1), ParameterCount(2)
->CompCode(0), Reason(0)

PCF String (MQCFST):
->Type(4), StrucLength(32), Parameter(2015), CharSet(0)
->StringLength(9), String("NASOAM_QM")

PCF Integer (MQCFIN):
->Type(3), StrucLength(16), Parameter (20119), Value(7)
THIS RECORD (LAST)

```

5. Issue the EXCMD_OPEN_EVENT command that enables the WMQ Agent to attach to its event queue and send events to the nsqcl utility program. For example:

```
Enter Command Name: EXCMD_OPEN_EVENT Enter
Object Name: NASTEL.EVENT.QUEUE
```

Ensure all the above commands succeed. Commands were successful if the CompCode and the Reason code of all responses are zero. If CompCode response is warning (1) or failure (2), write down the CompCode and Reason values and consult IBM's PCF Reference for details on the command and the reason of the failure.

6. Issue any of the MQCMD_XXX commands described in the IBM's PCF Reference manual to verify operation of any IBM-supported PCF command. For example:

```
Enter Command Name: MQCMD_INQUIRE_Q_NAMES
Enter Object Name: MQ*
Command (MQCMD_INQUIRE_Q_NAMES) has been issued
PCF RESPONSE FROM THE SERVER -->
```

PCF Header (MQCFH):

```
->Type(2), StrucLength(36), Version(1), Command(18)
->MsgSeqNum(1), Control(1), ParameterCount(5)
->CompCode(0), Reason(0)
```

PCF String List (MQCFSL):

```
->Type(6), StrucLength(216), Parameter (3011), Count(4)
->StringLength(48), CharSetId(0)
->String List
{
    (1) - "NASTEL.MMF.AUDIT.QUEUE"
    (2) - "NASTEL.MMF.ADMIN.COMMAND.QUEUE"
    (3) - "NASTEL.REPLY.xxx1999031119025854"
    (4) - "NASTEL.EVENT.QUEUE"
};
```

PCF Integer (MQCFIN):

```
->Type(3), StrucLength(16), Parameter (20126), Value(10)
```

PCF String (MQCFST):

```
->Type(4), StrucLength(24), Parameter(2016), CharSet(0)
->StringLength(3), String("MQ*")
```

PCF Integer (MQCFIN):

```
->Type(3), StrucLength(16), Parameter (20), Value(1001)
```

PCF Integer (MQCFIN):

```
->Type(3), StrucLength(16), Parameter (20119), Value(7)
```

THIS RECORD (LAST)

	NOTE:	Some commands will require attributes that NSQCL does not supply, and the command will fail.
---	--------------	--

When a WMQ Agent is functioning properly, all PCF commands should succeed with reason code equal to zero. Note that this rule does not apply to PCF commands that *start/stop* channels or other components, since their failure may be attributed to the WebSphere MQ configuration and authorization.

13.2.1 Checking WMQ Agent Parameters

The WMQ Agent should have full privileges to access WebSphere MQ configuration files, queue managers and queues. If you suspect a WMQ Agent has not been given the correct parameters, you should check the configuration files associated with it. For more information about configuring WMQ Agents, see the section on configuring WebSphere MQ nodes in Chapter 4.

When you have located the configuration files, perform the following checks:

- Ensure that any M6-WMQ environment variables (example: APWMQ_MQM) are defined and are valid.
- Ensure all necessary queues are defined on each queue manager. (See [Appendix D](#).)
- Ensure node registration was successful. Check the group registration file `mygroup.ini`. Ensure the format is correct and that entries point to the correct Workgroup server locations.
- On UNIX and Windows systems, ensure the WMQ Agent is running within the MQM user group or as a Windows service under the SYSTEM account.

13.3 Troubleshooting a Workgroup Server

If you are having problems with a workgroup server, it may be malfunctioning, or it may lack sufficient privileges to carry out the tasks you are requesting.

Verifying Workgroup Server Operation:

Turn on trace mode by entering the command:

```
nsqmgr -console -mGroupName -t -tc.
```

This trace records all PCF commands, responses and communication to and from the workgroup server. If the trace contains any error codes, look up the codes listed in IBM's *Programmable System Management (PSM)* manual to determine the problem.

Use `-ts` to trace SQL database operations.

13.3.1 Checking Workgroup Server Parameters

If you suspect a workgroup server has not been given correct parameters, perform following checks:

- Ensure all necessary environment variables are defined. The M6-WMQ variable should contain the path for the M6-WMQ installation.
- Ensure that one group is defined for the workgroup server located in directory `[APWMQ_HOME]\group\<group_name>`. Groups are defined using the `nsqsqlmk` utility.
- Use M6-WMQ Explorer to check the properties.
- Review authorization file `permits.ini`, in directory `[APWMQ_HOME]\config\groups`, if the workgroup server is running with the “+u” option. Review the syntax and the actual commands used, which can affect the ability of a user to execute commands and to see the responses.
- Ensure there is a valid license.

13.4 Getting a Spool File from the AS/400

To convert an APWMQ program's log/trace from a spool file to a format for transmitting to Nastel, follow this procedure:

1. On the command line type the Work with Spool Files command, WRKSPLF to display the *Work with Printer Output* screen and find the spool file of interest.
2. Enter option **9** (Work with printing status) to display the job information and record the values for **Job, User, and Number**. Example:

Working with Printing Status		
Job	NSQMSG	File : QPRINT
User	QMOM	Number : 1
Number	261850	User data. :
Job system name	S106EF2B	Creation date. . . . : 10182011
		Creation time. . . . : 214615

3. Create a physical file in a library for copying the spool file contents.

Command line example:

```
CRTPF FILE(NASTEMP/SPLFILES) RCDLEN(120) MBR(*NONE) TEXT('Spool files for
xfer to Nastel')
```

Physical file SPLFILES is created in library NASTEMP and allows for members with up to 120 characters per record.

4. Using the copy spool file command (CPYSPLF), copy the spool file to a member of the physical file using the QPRINT job information from step 2.

Command line example:

```
CPYSPLF FILE(QPRINT) TOFILE(NASTEMP/SPLFILES) JOB(261850/QMOM/NSQMSG)
```

System response:

```
Member MSGSRVTRC added to file SPLFILES in NASTEMP.
```

Or you can use **F4=Prompt** to display the following screen, enter the QPRINT job information, and press **Enter**.

Copy Spool File (CPYSPLF)		
Type choices, press Enter.		
Spool file	<u>QPRINT</u>	Name
To data base file.	<u>SPLFILES</u>	Name
Library	<u>NASTEMP</u>	Name, *LIBL, *CURLIB
Job name	<u>NSQMSG</u>	Name, *
User	<u>QMOM</u>	Name
Number	<u>261850</u>	000000-999999
Spool file number	<u>1</u>	1-999999, *ONLY, *LAST, *ANY
Job system name	<u>*ONLY</u>	Name, *ONLY, *LAST, *ANY
Spool file created:		
Creation date	<u>*ONLY</u>	Date, *ONLY, *LAST
Creation time	<u></u>	Time, *ONLY, *LAST
To member	<u>MSGSRVTRC</u>	Name, *FIRST
Replace or add records	<u>*REPLACE</u>	*REPLACE, *ADD

5. Verify the copied spool file by viewing it with command WRKMBRPDM.

Command line example:

```
WRKMBRPDM nastemp/splfiles
```

Enter option **5** (Display) next to the member you copied.

6. FTP in ASCII the spool file member of the physical file to your PC. Then email it to support@nastel.com or attach it to your problem report.

Appendix A: References

This appendix contains a list of reference material and documents relevant to M6-WMQ.

A.1 Nastel Documentation

Table A-1. Nastel Documentation	
Document Number (or higher)	Title
M6WMQ-INS 650.001	<i>Nastel AutoPilot M6 for WebSphere MQ Installation Guide</i>
M6-SM 650.001	<i>Nastel AutoPilot On-Demand for WebSphere MQ Security Manager User's Guide</i>
M6/WMQ 600.002	<i>Nastel AutoPilot M6 Plug-in for WebSphere MQ</i>
APM6/INS 600.008	<i>Nastel AutoPilot M6 Installation Guide</i>
APM6/USR 600.011	<i>Nastel AutoPilot M6 User's Guide</i>

A.2 IBM Documentation

<http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=%2Fcom.ibm.mq.csqzaj.doc%2Fsc10120.htm>

<http://www-01.ibm.com/software/integration/wmq/library/index.html>

<http://www-306.ibm.com/software/websphere/>

<http://www.unixodbc.org/doc/db2.html>

This Page Intentionally Left Blank

Appendix B: Conventions

B.1 Typographical Conventions

Table B-1. Typographical Conventions	
Convention	Description
Blue/Underlined	Used to identify links to referenced material or websites. Example: support@nastel.com
Bold Print	Used to identify topical headings, glossary entries, and toggles or buttons used in procedural steps. Example: Click EXIT .
<i>Italic Print</i>	Used to place emphasis on a title, menu, screen name, or other category.
Monospaced Bold	Used to identify keystrokes/data entries, file names, directory names, etc.
<i>Monospaced Italic</i>	Used to identify variables in an address location. Example: [<i>AUTOPILOT_HOME</i>] \documents . Where the portion of the address in the brackets[] is a variable.
Monospaced Text	Used to identify addresses, commands, script, etc.
Normal Text	Typically used for general text throughout the document.
Table Text	Table text is generally a smaller size to conserve space. 10, 9, and 8 point type is used in tables throughout the AutoPilot M6 product family documents.

B.2 Naming Conventions

Naming conventions have been adjusted to accommodate IBM's re-naming of MQSeries products to WebSphere MQ.

Nastel has adapted AutoPilot M6 products to reflect IBM's product naming changes. In the redesign of AutoPilot M6, we have also defined many elements within the AutoPilot M6 product line.

Table B-2. AutoPilot M6 Related Naming Conventions	
Old Name	New Name
AutoPilot/MQSI	AutoPilot M6/WMQI
MQSeries Plug-in for AutoPilot	WebSphere MQ plug-in for AutoPilot M6
MQControl	AutoPilot M6-WMQ
MQSeries (IBM)	WebSphere MQ (IBM)

B.2.1 Component Naming Conventions

Table B-3. M6 for WebSphere MQ Naming Conventions		
New Name	Old Name	Convention
WMQ Agent	Local Manager	An intelligent agent that runs on a WebSphere MQ node
Workgroup Server	Group Manager	Collects information about a group of registered WMQ nodes and stores it in its <i>database</i>
CEP Server	Managed node	Provides M6-WMQ monitoring experts, automation policies, and business views

Appendix C: Executing User-Defined Scripts

M6-WMQ provides a method for executing user-supplied scripts or programs when a given event occurs. A simple script-naming convention associates a script with a specific event. When the event occurs, all event-dependent parameters are passed to the script on the command line that invoked the script.

Scripts can be executed on the WebSphere MQ node where the event occurs, or at the level of the Workgroup server(s) that are managing the node. They are invoked by the WMQ Agent on the node where the event actually occurs, or by any Workgroup server to which the event is sent. A script for an event may be placed on the WMQ Agent node, the Workgroup server level, or both.

C.1 Script Naming Conventions

Scripts must be named after the events to which they are associated. Each script must have the name:

<CATID>_<EVENTID>

where:

<CATID> is a PCF event category ID, given as one of the integers in Table C-1.

Event Category	Integer
MQCMD_Q_MGR_EVENT	44
MQCMD_PERFM_EVENT	45
MQCMD_CHANNEL_EVENT	46

<EVENTID> is a PCF event ID number, given as one of the integers in the *Integer Values for PCF Events* tables. For example, if association of a script with a channel-stopped event (MQRC_CHANNEL_STOPPED) is required, name the script 46_2283. If the script was a Windows XP batch script, its full name would be 46_2283.bat.

Table C-2. Integer Values for PCF Events/Reason Codes (from header file cmqc.h)

PCF Event	Integer Value	Integer Value	Integer Value
MQRC_ALIAS_BASE_Q_TYPE_ERROR	2001L	MQRC_ALREADY_CONNECTED	2002L
MQRC_BACKED_OUT	2003L	MQRC_BUFFER_ERROR	2004L
MQRC_BUFFER_LENGTH_ERROR	2005L	MQRC_CHAR_ATTR_LENGTH_ERROR	2006L
MQRC_CHAR_ATTRS_ERROR	2007L	MQRC_CHAR_ATTRS_TOO_SHORT	2008L
MQRC_CONNECTION_BROKEN	2009L	MQRC_DATA_LENGTH_ERROR	2010L
MQRC_DYNAMIC_Q_NAME_ERROR	2011L	MQRC_ENVIRONMENT_ERROR	2012L
MQRC_EXPIRY_ERROR	2013L	MQRC_FEEDBACK_ERROR	2014L
MQRC_GET_INHIBITED	2016L	MQRC_HANDLE_NOT_AVAILABLE	2017L
MQRC_HCONN_ERROR	2018L	MQRC_HOBJ_ERROR	2019L
MQRC_INHIBIT_VALUE_ERROR	2020L	MQRC_INT_ATTR_COUNT_ERROR	2021L
MQRC_INT_ATTR_COUNT_TOO_SMALL	2022L	MQRC_INT_ATTRS_ARRAY_ERROR	2023L
MQRC_SYNCPOINT_LIMIT_REACHED	2024L	MQRC_MAX_CONNS_LIMIT_REACHED	2025L
MQRC_MD_ERROR	2026L	MQRC_MISSING_REPLY_TO_Q	2027L
MQRC_MSG_TYPE_ERROR	2029L	MQRC_MSG_TOO_BIG_FOR_Q	2030L
MQRC_MSG_TOO_BIG_FOR_Queue_MGR	2031L	MQRC_NO_MSG_AVAILABLE	2033L
MQRC_NO_MSG_UNDER_CURSOR	2034L	MQRC_NOT_AUTHORIZED	2035L
MQRC_NOT_OPEN_FOR_BROWSE	2036L	MQRC_NOT_OPEN_FOR_INPUT	2037L
MQRC_NOT_OPEN_FOR_INQUIRE	2038L	MQRC_NOT_OPEN_FOR_OUTPUT	2039L
MQRC_NOT_OPEN_FOR_SET	2040L	MQRC_OBJECT_CHANGED	2041L
MQRC_OBJECT_IN_USE	2042L	MQRC_OBJECT_TYPE_ERROR	2043L
MQRC_OD_ERROR	2044L	MQRC_OPTION_NOT_VALID_FORTYPE	2045L
MQRC_OPTIONS_ERROR	2046L	MQRC_PERSISTENCE_ERROR	2047L
MQRC_PERSISTENT_NOT_ALLOWED	2048L	MQRC_PRIORITY_EXCEEDS_MAXIMUM	2049L
MQRC_PRIORITY_ERROR	2050L	MQRC_PUT_INHIBITED	2051L
MQRC_Q_DELETED	2052L	MQRC_Q_FULL	2053L
MQRC_Q_NOT_EMPTY	2055L	MQRC_Q_SPACE_NOT_AVAILABLE	2056L
MQRC_Q_TYPE_ERROR	2057L	MQRC_Q_MGR_NAME_ERROR	2058L
MQRC_Q_MGR_NOT_AVAILABLE	2059L	MQRC_REPORT_OPTIONS_ERROR	2061L
MQRC_SECOND_MARK_NOT_ALLOWED	2062L	MQRC_SECURITY_ERROR	2063L
MQRC_SELECTOR_COUNT_ERROR	2065L	MQRC_SELECTOR_LIMIT_EXCEEDED	2066L
MQRC_SELECTOR_ERROR	2067L	MQRC_SELECTOR_NOT_FOR_TYPE	2068L
MQRC_SIGNAL_OUTSTANDING	2069L	MQRC_SIGNAL_REQUEST_ACCEPTED	2070L
MQRC_STORAGE_NOT_AVAILABLE	2071L	MQRC_SYNCPOINT_NOT_AVAILABLE	2072L
MQRC_TRIGGER_CONTROL_ERROR	2075L	MQRC_TRIGGER_DEPTH_ERROR	2076L
MQRC_TRIGGER_MSG_PRIORITY_ERR	2077L	MQRC_TRIGGER_TYPE_ERROR	2078L
MQRC_TRUNCATED_MSG_ACCEPTED	2079L	MQRC_TRUNCATED_MSG_FAILED	2080L
MQRC_UNKNOWN_ALIAS_BASE_Q	2082L	MQRC_UNKNOWN_OBJECT_NAME	2085L
MQRC_UNKNOWN_OBJECT_Q_MGR	2086L	MQRC_UNKNOWN_REMOTE_Q_MGR	2087L
MQRC_WAIT_INTERVAL_ERROR	2090L	MQRC_XMIT_Q_TYPE_ERROR	2091L
MQRC_XMIT_Q_USAGE_ERROR	2092L	MQRC_NOT_OPEN_FOR_PASS_ALL	2093L
MQRC_NOT_OPEN_FOR_PASS_IDENT	2094L	MQRC_NOT_OPEN_FOR_SET_ALL	2095L

Table C-2. Integer Values for PCF Events/Reason Codes (from header file cmqc.h)

PCF Event	Integer Value	Integer Value	Integer Value
MQRC_NOT_OPEN_FOR_SET_IDENT	2096L	MQRC_CONTEXT_HANDLE_ERROR	2097L
MQRC_CONTEXT_NOT_AVAILABLE	2098L	MQRC_SIGNAL1_ERROR	2099L
MQRC_OBJECT_ALREADY_EXISTS	2100L	MQRC_OBJECT_DAMAGED	2101L
MQRC_RESOURCE_PROBLEM	2102L	MQRC_ANOTHER_Q_MGR_CONNECTED	2103L
MQRC_UNKNOWN_REPORT_OPTION	2104L	MQRC_STORAGE_CLASS_ERROR	2105L
MQRC_COD_NOT_VALID_FOR_XCF_Q	2106L	MQRC_XWAIT_CANCELED	2107L
MQRC_XWAIT_ERROR	2108L	MQRC_SUPPRESSED_BY_EXIT	2109L
MQRC_FORMAT_ERROR	2110L	MQRC_SOURCE_CCSID_ERROR	2111L
MQRC_SOURCE_INTEGER_ENC_ERROR	2112L	MQRC_SOURCE_DECIMAL_ENC_ERROR	2113L
MQRC_SOURCE_FLOAT_ENC_ERROR	2114L	MQRC_TARGET_CCSID_ERROR	2115L
MQRC_TARGET_INTEGER_ENC_ERROR	2116L	MQRC_TARGET_DECIMAL_ENC_ERROR	2117L
MQRC_TARGET_FLOAT_ENC_ERROR	2118L	MQRC_NOT_CONVERTED	2119L
MQRC_CONVERTED_MSG_TOO_BIG	2120L	MQRC_TRUNCATED	2120L
MQRC_NO_EXTERNAL_PARTICIPANTS	2121L	MQRC_PARTICIPANT_NOT_AVAILABLE	2122L
MQRC_OUTCOME_MIXED	2123L	MQRC_OUTCOME_PENDING	2124L
MQRC_BRIDGE_STARTED	2125L	MQRC_BRIDGE_STOPPED	2126L
MQRC_ADAPTER_STORAGE_SHORTAGE	2127L	MQRC_UOW_IN_PROGRESS	2128L
MQRC_ADAPTER_CONN_LOAD_ERROR	2129L	MQRC_ADAPTER_SERV_LOAD_ERROR	2130L
MQRC_ADAPTER_CONV_LOAD_ERROR	2133L	MQRC_BO_ERROR	2134L
MQRC_DH_ERROR	2135L	MQRC_MULTIPLE_REASONS	2136L
MQRC_OPEN_FAILED	2137L	MQRC_ADAPTER_DISC_LOAD_ERROR	2138L
MQRC_CNO_ERROR	2139L	MQRC_CICS_WAIT_FAILED	2140L
MQRC_DLH_ERROR	2141L	MQRC_HEADER_ERROR	2142L
MQRC_SOURCE_LENGTH_ERROR	2143L	MQRC_TARGET_LENGTH_ERROR	2144L
MQRC_SOURCE_BUFFER_ERROR	2145L	MQRC_TARGET_BUFFER_ERROR	2146L
MQRC_IH_ERROR	2148L	MQRC_PCF_ERROR	2149L
MQRC_DBCS_ERROR	2150L	MQRC_OBJECT_NAME_ERROR	2152L
MQRC_OBJECT_Q_MGR_NAME_ERROR	2153L	MQRC_RECS_PRESENT_ERROR	2154L
MQRC_OBJECT_RECORDS_ERROR	2155L	MQRC_RESPONSE_RECORDS_ERROR	2156L
MQRC_ASID_MISMATCH	2157L	MQRC_PMO_RECORD_FLAGS_ERROR	2158L
MQRC_PUT_MSG_RECORDS_ERROR	2159L	MQRC_CONN_ID_IN_USE	2160L
MQRC_Q_MGR QUIESCING	2161L	MQRC_Q_MGR_STOPPING	2162L
MQRC_DUPLICATE_RECOV_COORD	2163L	MQRC_PMO_ERROR	2173L
MQRC_API_EXIT_NOT_FOUND	2182L	MQRC_API_EXIT_LOAD_ERROR	2183L
MQRC_REMOTE_Q_NAME_ERROR	2184L	MQRC_INCONSISTENT_PERSISTENCE	2185L
MQRC_GMO_ERROR	2186L	MQRC_TMC_ERROR	2191L
MQRC_PAGESET_FULL	2192L	MQRC_PAGESET_ERROR	2193L
MQRC_NAME_NOT_VALID_FOR_TYPE	2194L	MQRC_UNEXPECTED_ERROR	2195L
MQRC_UNKNOWN_XMIT_Q	2196L	MQRC_UNKNOWN_DEF_XMIT_Q	2197L
MQRC_DEF_XMIT_Q_TYPE_ERROR	2198L	MQRC_DEF_XMIT_Q_USAGE_ERROR	2199L
MQRC_NAME_IN_USE	2201L	MQRC_CONNECTION QUIESCING	2202L
MQRC_CONNECTION_STOPPING	2203L	MQRC_ADAPTER_NOT_AVAILABLE	2204L
MQRC_MSG_ID_ERROR	2206L	MQRC_CORREL_ID_ERROR	2207L
MQRC_FILE_SYSTEM_ERROR	2208L	MQRC_NO_MSG_LOCKED	2209L
MQRC_SOAP_DOTNET_ERROR	2210L	MQRC_SOAP_AXIS_ERROR	2211L

Table C-2. Integer Values for PCF Events/Reason Codes (from header file cmqc.h)

PCF Event	Integer Value	Integer Value	Integer Value
MQRC_SOAP_URL_ERROR	2212L	MQRC_FILE_NOT_AUDITED	2216L
MQRC_CONNECTION_NOT_AUTHORIZED	2217L	MQRC_MSG_TOO_BIG_FOR_CHANNEL	2218L
MQRC_CALL_IN_PROGRESS	2219L	MQRC_RMH_ERROR	2220L
MQRC_Q_MGR_ACTIVE	2222L	MQRC_Q_MGR_NOT_ACTIVE	2223L
MQRC_Q_DEPTH_HIGH	2224L	MQRC_Q_DEPTH_LOW	2225L
MQRC_Q_SERVICE_INTERVAL_HIGH	2226L		
MQRC_Q_SERVICE_INTERVAL_OK	2227L	MQRC_CHANNEL_AUTO_DEF_OK	2233L
MQRC_CHANNEL_AUTO_DEF_ERROR	2234L	MQRC_CFH_ERROR	2235L
MQRC_CFIL_ERROR	2236L	MQRC_CFIN_ERROR	2237L
MQRC_CFSL_ERROR	2238L	MQRC_CFST_ERROR	2239L
MQRC_INCOMPLETE_GROUP	2238L	MQRC_INCOMPLETE_MSG	2242L
MQRC_INCONSISTENT_CCSIDS	2243L	MQRC_INCONSISTENT_ENCODINGS	2244L
MQRC_INCONSISTENT_UOW	2245L	MQRC_INVALID_MSG_UNDER_CURSOR	2246L
MQRC_MATCH_OPTIONS_ERROR	2247L	MQRC_MDE_ERROR	2248L
MQRC_MSG_FLAGS_ERROR	2249L	MQRC_MSG_SEQ_NUMBER_ERROR	2250L
MQRC_OFFSET_ERROR	2251L	MQRC_ORIGINAL_LENGTH_ERROR	2252L
MQRC_SEGMENT_LENGTH_ZERO	2253L	MQRC_UOW_NOT_AVAILABLE	2255L
MQRC_WRONG_GMO_VERSION	2256L	MQRC_WRONG_MD_VERSION	2257L
MQRC_GROUP_ID_ERROR	2258L	MQRC_INCONSISTENT_BROWSE	2259L
MQRC_XQH_ERROR	2260L	MQRC_SRC_ENV_ERROR	2261L
MQRC_SRC_NAME_ERROR	2262L	MQRC_DEST_ENV_ERROR	2263L
MQRC_DEST_NAME_ERROR	2264L	MQRC_TM_ERROR	2265L
MQRC_HCONFIG_ERROR	2280L	MQRC_FUNCTION_ERROR	2281L
MQRC_CHANNEL_STARTED	2282L	MQRC_CHANNEL_STOPPED	2283L
MQRC_CHANNEL_CONV_ERROR	2284L	MQRC_SERVICE_NOT_AVAILABLE	2285L
MQRC_INITIALIZATION_FAILED	2286L	MQRC_TERMINATION_FAILED	2287L
MQRC_UNKNOWN_Q_NAME	2288L	MQRC_SERVICE_ERROR	2289L
MQRC_Q_ALREADY_EXISTS	2290L	MQRC_USER_ID_NOT_AVAILABLE	2291L
MQRC_UNKNOWN_ENTITY	2292L	MQRC_UNKNOWN_AUTH_ENTITY	2293L
MQRC_UNKNOWN_REF_OBJECT	2294L	MQRC_CHANNEL_ACTIVATED	2295L
MQRC_CHANNEL_NOT_ACTIVATED	2296L	MQRC_UOW_CANCELED	2297L
MQRC_FUNCTION_NOT_SUPPORTED	2298L	MQRC_SELECTOR_TYPE_ERROR	2299L
MQRC_COMMAND_TYPE_ERROR	2300	MQRC_MULTIPLE_INSTANCE_ERROR	2301
MQRC_SYSTEM_ITEM_NOT_ALTERABLE	2302	MQRC_BAG_CONVERSION_ERROR	2303
MQRC_SELECTOR_OUT_OF_RANGE	2304	MQRC_SELECTOR_NOT_UNIQUE	2305
MQRC_INDEX_NOT_PRESENT	2306	MQRC_STRING_ERROR	2307
MQRC_ENCODING_NOT_SUPPORTED	2308	MQRC_SELECTOR_NOT_PRESENT	2309
MQRC_OUT_SELECTOR_ERROR	2310	MQRC_STRING_TRUNCATED	2311
MQRC_SELECTOR_WRONG_TYPE	2312	MQRC_INCONSISTENT_ITEM_TYPE	2313
MQRC_INDEX_ERROR	2314	MQRC_SYSTEM_BAG_NOT_ALTERABLE	2315
MQRC_ITEM_COUNT_ERROR	2316	MQRC_FORMAT_NOT_SUPPORTED	2317
MQRC_SELECTOR_NOT_SUPPORTED	2318	MQRC_ITEM_VALUE_ERROR	2319
MQRC_HBAG_ERROR	2320	MQRC_PARAMETER_MISSING	2321
MQRC_CMD_SERVER_NOT_AVAILABLE	2322	MQRC_STRING_LENGTH_ERROR	2323
MQRC_INQUIRY_COMMAND_ERROR	2324	MQRC_NESTED_BAG_NOT_SUPPORTED	2325

Table C-2. Integer Values for PCF Events/Reason Codes (from header file cmqc.h)

PCF Event	Integer Value	Integer Value	Integer Value
MQRC_BAG_WRONG_TYPE	2326	MQRC_ITEM_TYPE_ERROR	2327
MQRC_SYSTEM_BAG_NOT_DELETABLE	2328	MQRC_SYSTEM_ITEM_NOT_DELETABLE	2329
MQRC_CODED_CHAR_SET_ID_ERROR	2330	MQRC_MSG_TOKEN_ERROR	2331
MQRC_MISSING_WIH	2332	MQRC_WIH_ERROR	2333
MQRC_RFH_ERROR	2334	MQRC_RFH_STRING_ERROR	2335
MQRC_RFH_COMMAND_ERROR	2336	MQRC_RFH_PARM_ERROR	2337
MQRC_RFH_DUPLICATE_PARM	2338	MQRC_RFH_PARM_MISSING	2339
MQRC_CHAR_CONVERSION_ERROR	2340	MQRC_UCS2_CONVERSION_ERROR	2341
MQRC_DB2_NOT_AVAILABLE	2342	MQRC_OBJECT_NOT_UNIQUE	2343
MQRC_CONN_TAG_NOT_RELEASED	2344	MQRC_CF_NOT_AVAILABLE	2345
MQRC_CF_STRUC_IN_USE	2346	MQRC_CF_STRUC_LIST_HDR_IN_USE	2347
MQRC_CF_STRUC_AUTH_FAILED	2348	MQRC_CF_STRUC_ERROR	2349
MQRC_CONN_TAG_NOT_USABLE	2350	MQRC_GLOBAL_UOW_CONFLICT	2351
MQRC_LOCAL_UOW_CONFLICT	2352	MQRC_HANDLE_IN_USE_FOR_UOW	2353
MQRC_UOW_ENLISTMENT_ERROR	2354	MQRC_UOW_MIX_NOT_SUPPORTED	2355
MQRC_WXP_ERROR	2356	MQRC_CURRENT_RECORD_ERROR	2357
MQRC_NEXT_OFFSET_ERROR	2358	MQRC_NO_RECORD_AVAILABLE	2359
MQRC_OBJECT_LEVEL_INCOMPATIBLE	2360	MQRC_NEXT_RECORD_ERROR	2361
MQRC_BACKOUT_THRESHOLD_REACHED	2362	MQRC_MSG_NOT_MATCHED	2363
MQRC_JMS_FORMAT_ERROR	2364	MQRC_SEGMENTS_NOT_SUPPORTED	2365
MQRC_WRONG_CF_LEVEL	2366	MQRC_CONFIG_CREATE_OBJECT	2367
MQRC_CONFIG_CHANGE_OBJECT	2368	MQRC_CONFIG_DELETE_OBJECT	2369
MQRC_CONFIG_REFRESH_OBJECT	2370	MQRC_CHANNEL_SSL_ERROR	2371
MQRC_CF_STRUC_FAILED	2373	MQRC_API_EXIT_ERROR	2374
MQRC_API_EXIT_INIT_ERROR	2375	MQRC_API_EXIT_TERM_ERROR	2376
MQRC_EXIT_REASON_ERROR	2377	MQRC_RESERVED_VALUE_ERROR	2378
MQRC_NO_DATA_AVAILABLE	2379	MQRC_SCO_ERROR	2380
MQRC_KEY_REPOSITORY_ERROR	2381	MQRC_CRYPTO_HARDWARE_ERROR	2382
MQRC_AUTH_INFO_REC_COUNT_ERROR	2383	MQRC_AUTH_INFO_REC_ERROR	2384
MQRC_AIR_ERROR	2385	MQRC_AUTH_INFO_TYPE_ERROR	2386
MQRC_AUTH_INFO_CONN_NAME_ERROR	2387	MQRC_LDAP_USER_NAME_ERROR	2388
MQRC_LDAP_USER_NAME_LENGTH_ERR	2389	MQRC_LDAP_PASSWORD_ERROR	2390
MQRC_SSL_ALREADY_INITIALIZED	2391	MQRC_SSL_CONFIG_ERROR	2392
MQRC_SSL_INITIALIZATION_ERROR	2393	MQRC_Q_INDEX_TYPE_ERROR	2394
MQRC_CFBS_ERROR	2395	MQRC_SSL_NOT_ALLOWED	2396
MQRC_JSSE_ERROR	2397	MQRC_SSL_PEER_NAME_MISMATCH	2398
MQRC_SSL_PEER_NAME_ERROR	2399	MQRC_UNSUPPORTED_CIPHER_SUITE	2400
MQRC_SSL_CERTIFICATE_REVOKED	2401	MQRC_SSL_CERT_STORE_ERROR	2402
MQRC_CLIENT_EXIT_LOAD_ERROR	2406	MQRC_CLIENT_EXIT_ERROR	2407
MQRC_SSL_KEY_RESET_ERROR	2409	MQRC_UNKNOWN_COMPONENT_NAME	2410
MQRC_LOGGER_STATUS	2411	MQRC_COMMAND_MQSC	2412
MQRC_COMMAND_PCF	2413	MQRC_CFIF_ERROR	2414
MQRC_CFSF_ERROR	2415	MQRC_CFGR_ERROR	2416
MQRC_MSG_NOT_ALLOWED_IN_GROUP	2417	MQRC_FILTER_OPERATOR_ERROR	2418
MQRC_NESTED_SELECTOR_ERROR	2419	MQRC_EPH_ERROR	2420

Table C-2. Integer Values for PCF Events/Reason Codes (from header file cmqc.h)

PCF Event	Integer Value	Integer Value	Integer Value
MQRC_RFH_FORMAT_ERROR	2421	MQRC_CFBF_ERROR	2422
MQRC_CLIENT_CHANNEL_CONFLICT	2423	MQRC_SD_ERROR	2424
MQRC_TOPIC_STRING_ERROR	2425	MQRC_STS_ERROR	2426
MQRC_NO_SUBSCRIPTION	2428	MQRC_SUBSCRIPTION_IN_USE	2429
MQRC_STAT_TYPE_ERROR	2430	MQRC_SUB_USER_DATA_ERROR	2431
MQRC_SUB_ALREADY_EXISTS	2432	MQRC_IDENTITY_MISMATCH	2434
MQRC_ALTER_SUB_ERROR	2435	MQRC_DURABILITY_NOT_ALLOWED	2436
MQRC_NO_RETAINED_MSG	2437	MQRC_SRO_ERROR	2438
MQRC_SUB_NAME_ERROR	2440	MQRC_OBJECT_STRING_ERROR	2441
MQRC_PROPERTY_NAME_ERROR	2442	MQRC_SEGMENTATION_NOT_ALLOWED	2443
MQRC_CBD_ERROR	2444	MQRC_CTLO_ERROR	2445
MQRC_NO_CALLBACKS_ACTIVE	2446	MQRC_CALLBACK_NOT_REGISTERED	2448
MQRC_OPTIONS_CHANGED	2457	MQRC_READ_AHEAD_MSGS	2458
MQRC_SELECTOR_SYNTAX_ERROR	2459	MQRC_HMSG_ERROR	2460
MQRC_CMHO_ERROR	2461	MQRC_DMHO_ERROR	2462
MQRC_SMPO_ERROR	2463	MQRC_IMPO_ERROR	2464
MQRC_PROPERTY_NAME_TOO_BIG	2465	MQRC_PROP_VALUE_NOT_CONVERTED	2466
MQRC_PROP_TYPE_NOT_SUPPORTED	2467	MQRC_PROPERTY_VALUE_TOO_BIG	2469
MQRC_PROP_CONV_NOT_SUPPORTED	2470	MQRC_PROPERTY_NOT_AVAILABLE	2471
MQRC_PROP_NUMBER_FORMAT_ERROR	2472	MQRC_PROPERTY_TYPE_ERROR	2473
MQRC_PROPERTIES_TOO_BIG	2478	MQRC_PUT_NOT_RETAINED	2479
MQRC_ALIAS_TARGTYPE_CHANGED	2480	MQRC_DMPO_ERROR	2481
MQRC_PD_ERROR	2482	MQRC_CALLBACK_TYPE_ERROR	2483
MQRC_CBD_OPTIONS_ERROR	2484	MQRC_MAX_MSG_LENGTH_ERROR	2485
MQRC_CALLBACK_ROUTINE_ERROR	2486	MQRC_CALLBACK_LINK_ERROR	2487
MQRC_OPERATION_ERROR	2488	MQRC_BMHO_ERROR	2489
MQRC_UNSUPPORTED_PROPERTY	2490	MQRC_PROP_NAME_NOT_CONVERTED	2492
MQRC_GET_ENABLED	2494	MQRC_MODULE_NOT_FOUND	2495
MQRC_MODULE_INVALID	2496	MQRC_MODULE_ENTRY_NOT_FOUND	2497
MQRC_MIXED_CONTENT_NOT_ALLOWED	2498	MQRC_MSG_HANDLE_IN_USE	2499
MQRC_HCONN_ASYNC_ACTIVE	2500	MQRC_MHBO_ERROR	2501
MQRC_PUBLICATION_FAILURE	2502	MQRC_SUB_INHIBITED	2503
MQRC_SELECTOR_ALWAYS_FALSE	2504	MQRC_XEPO_ERROR	2507
MQRC_DURABILITY_NOT_ALTERABLE	2509	MQRC_TOPIC_NOT_ALTERABLE	2510
MQRC_SUBLEVEL_NOT_ALTERABLE	2512	MQRC_PROPERTY_NAME_LENGTH_ERR	2513
MQRC_DUPLICATE_GROUP_SUB	2514	MQRC_GROUPING_NOT_ALTERABLE	2515
MQRC_SELECTOR_INVALID_FOR_TYPE	2516	MQRC_HOBJ QUIESCED	2517
MQRC_HOBJ QUIESCED_NO_MSGS	2518	MQRC_SELECTION_STRING_ERROR	2519
MQRC_RES_OBJECT_STRING_ERROR	2520	MQRC_CONNECTION_SUSPENDED	2521
MQRC_INVALID_DESTINATION	2522	MQRC_INVALID_SUBSCRIPTION	2523
MQRC_SELECTOR_NOT_ALTERABLE	2524	MQRC_RETAINED_MSG_Q_ERROR	2525
MQRC_RETAINED_NOT_DELIVERED	2526	MQRC_RFH_RESTRICTED_FORMAT_ERR	2527
MQRC_CONNECTION_STOPPED	2528	MQRC_ASYNC_UOW_CONFLICT	2529
MQRC_ASYNC_XA_CONFLICT	2530	MQRC_PUBSUB_INHIBITED	2531
MQRC_MSG_HANDLE_COPY_FAILURE	2532	MQRC_DEST_CLASS_NOT_ALTERABLE	2533

Table C-2. Integer Values for PCF Events/Reason Codes (from header file cmqc.h)

PCF Event	Integer Value	Integer Value	Integer Value
MQRC_OPERATION_NOT_ALLOWED	2534	MQRC_ACTION_ERROR	2535
MQRC_CHANNEL_NOT_AVAILABLE	2537	MQRC_HOST_NOT_AVAILABLE	2538
MQRC_CHANNEL_CONFIG_ERROR	2539	MQRC_UNKNOWN_CHANNEL_NAME	2540
MQRC_LOOPING_PUBLICATION	2541	MQRC_ALREADY_JOINED	2542
MQRC_STANDBY_Q_MGR	2543	MQRC_RECONNECTING	2544
MQRC_RECONNECTED	2545	MQRC_RECONNECT_QMID_MISMATCH	2546
MQRC_RECONNECT_INCOMPATIBLE	2547	MQRC_RECONNECT_FAILED	2548
MQRC_CALL_INTERRUPTED	2549	MQRC_NO_SUBS_MATCHED	2550
MQRC_SELECTION_NOT_AVAILABLE	2551	MQRC_CHANNEL_SSL_WARNING	2552
MQRC_OSCP_URL_ERROR	2553	MQRC_CONTENT_ERROR	2554
MQRC_RECONNECT_Q_MGR_REQD	2555	MQRC_RECONNECT_TIMED_OUT	2556
MQRC_PUBLISH_EXIT_ERROR	2557	MQRCCF_CFH_TYPE_ERROR	3001
MQRCCF_CFH_LENGTH_ERROR	3002	MQRCCF_CFH_VERSION_ERROR	3003
MQRCCF_CFH_MSG_SEQ_NUMBER_ERR	3004	MQRCCF_CFH_CONTROL_ERROR	3005
MQRCCF_CFH_PARM_COUNT_ERROR	3006	MQRCCF_CFH_COMMAND_ERROR	3007
MQRCCF_COMMAND_FAILED	3008	MQRCCF_CFIN_LENGTH_ERROR	3009
MQRCCF_CFST_LENGTH_ERROR	3010	MQRCCF_CFST_STRING_LENGTH_ERROR	3011
MQRCCF_FORCE_VALUE_ERROR	3012	MQRCCF_STRUCTURE_TYPE_ERROR	3013
MQRCCF_CFIN_PARM_ID_ERROR	3014	MQRCCF_CFST_PARM_ID_ERROR	3015
MQRCCF_MSG_LENGTH_ERROR	3016	MQRCCF_CFIN_DUPLICATE_PARM	3017
MQRCCF_CFST_DUPLICATE_PARM	3018	MQRCCF_PARM_COUNT_TOO_SMALL	3019
MQRCCF_PARM_COUNT_TOO_BIG	3020	MQRCCF_Q_ALREADY_IN_CELL	3021
MQRCCF_Q_TYPE_ERROR	3022	MQRCCF_MD_FORMAT_ERROR	3023
MQRCCF_CFSL_LENGTH_ERROR	3024	MQRCCF_REPLACE_VALUE_ERROR	3025
MQRCCF_CFIL_DUPLICATE_VALUE	3026	MQRCCF_CFIL_COUNT_ERROR	3027
MQRCCF_CFIL_LENGTH_ERROR	3028	MQRCCF_QUIESCE_VALUE_ERROR	3029
MQRCCF_MSG_SEQ_NUMBER_ERROR	3030	MQRCCF_PING_DATA_COUNT_ERROR	3031
MQRCCF_PING_DATA_COMPARE_ERROR	3032	MQRCCF_CFSL_PARM_ID_ERROR	3033
MQRCCF_CHANNEL_TYPE_ERROR	3034	MQRCCF_PARM_SEQUENCE_ERROR	3035
MQRCCF_XMIT_PROTOCOL_TYPE_ERR	3036	MQRCCF_BATCH_SIZE_ERROR	3037
MQRCCF_DISC_INT_ERROR	3038	MQRCCF_SHORT_RETRY_ERROR	3039
MQRCCF_SHORT_TIMER_ERROR	3040	MQRCCF_LONG_RETRY_ERROR	3041
MQRCCF_LONG_TIMER_ERROR	3042	MQRCCF_SEQ_NUMBER_WRAP_ERROR	3043
MQRCCF_MAX_MSG_LENGTH_ERROR	3044	MQRCCF_PUT_AUTH_ERROR	3045
MQRCCF_PURGE_VALUE_ERROR	3046	MQRCCF_CFIL_PARM_ID_ERROR	3047
MQRCCF_MSG_TRUNCATED	3048	MQRCCF_CCSID_ERROR	3049
MQRCCF_ENCODING_ERROR	3050	MQRCCF_QUEUES_VALUE_ERROR	3051
MQRCCF_DATA_CONV_VALUE_ERROR	3052	MQRCCF_INDOUBT_VALUE_ERROR	3053
MQRCCF_ESCAPE_TYPE_ERROR	3054	MQRCCF_REPOS_VALUE_ERROR	3055
MQRCCF_CHANNEL_TABLE_ERROR	3062	MQRCCF_MCA_TYPE_ERROR	3063
MQRCCF_CHL_INST_TYPE_ERROR	3064	MQRCCF_CHL_STATUS_NOT_FOUND	3065
MQRCCF_CFSL_DUPLICATE_PARM	3066	MQRCCF_CFSL_TOTAL_LENGTH_ERROR	3067
MQRCCF_CFSL_COUNT_ERROR	3068	MQRCCF_CFSL_STRING_LENGTH_ERR	3069
MQRCCF_BROKER_DELETED	3070	MQRCCF_STREAM_ERROR	3071
MQRCCF_TOPIC_ERROR	3072	MQRCCF_NOT_REGISTERED	3073

Table C-2. Integer Values for PCF Events/Reason Codes (from header file cmqc.h)

PCF Event	Integer Value	Integer Value	Integer Value
MQRCCF_Q_MGR_NAME_ERROR	3074	MQRCCF_INCORRECT_STREAM	3075
MQRCCF_Q_NAME_ERROR	3076	MQRCCF_NO_RETAINED_MSG	3077
MQRCCF_DUPLICATE_IDENTITY	3078	MQRCCF_INCORRECT_Q	3079
MQRCCF_CORREL_ID_ERROR	3080	MQRCCF_NOT_AUTHORIZED	3081
MQRCCF_UNKNOWN_STREAM	3082	MQRCCF_REG_OPTIONS_ERROR	3083
MQRCCF_PUB_OPTIONS_ERROR	3084	MQRCCF_UNKNOWN_BROKER	3085
MQRCCF_Q_MGR_CCSID_ERROR	3086	MQRCCF_DEL_OPTIONS_ERROR	3087
MQRCCF_CLUSTER_NAME_CONFLICT	3088	MQRCCF_REPOS_NAME_CONFLICT	3089
MQRCCF_CLUSTER_Q_USAGE_ERROR	3090	MQRCCF_ACTION_VALUE_ERROR	3091
MQRCCF_COMMS_LIBRARY_ERROR	3092	MQRCCF_NETBIOS_NAME_ERROR	3093
MQRCCF_BROKER_COMMAND_FAILED	3094	MQRCCF_CFST_CONFLICTING_PARM	3095
MQRCCF_PATH_NOT_VALID	3096	MQRCCF_PARM_SYNTAX_ERROR	3097
MQRCCF_PWD_LENGTH_ERROR	3098	MQRCCF_FILTER_ERROR	3150
MQRCCF_WRONG_USER	3151	MQRCCF_DUPLICATE_SUBSCRIPTION	3152
MQRCCF_SUB_NAME_ERROR	3153	MQRCCF_SUB_IDENTITY_ERROR	3154
MQRCCF_SUBSCRIPTION_IN_USE	3155	MQRCCF_SUBSCRIPTION_LOCKED	3156
MQRCCF_ALREADY_JOINED	3157	MQRCCF_OBJECT_IN_USE	3160
MQRCCF_UNKNOWN_FILE_NAME	3161	MQRCCF_FILE_NOT_AVAILABLE	3162
MQRCCF_DISC_RETRY_ERROR	3163	MQRCCF_ALLOC_RETRY_ERROR	3164
MQRCCF_ALLOC_SLOW_TIMER_ERROR	3165	MQRCCF_ALLOC_FAST_TIMER_ERROR	3166
MQRCCF_PORT_NUMBER_ERROR	3167	MQRCCF_CHL_SYSTEM_NOT_ACTIVE	3168
MQRCCF_ENTITY_NAME_MISSING	3169	MQRCCF_PROFILE_NAME_ERROR	3170
MQRCCF_AUTH_VALUE_ERROR	3171	MQRCCF_AUTH_VALUE_MISSING	3172
MQRCCF_OBJECT_TYPE_MISSING	3173	MQRCCF_CONNECTION_ID_ERROR	3174
MQRCCF_LOG_TYPE_ERROR	3175	MQRCCF_PROGRAM_NOT_AVAILABLE	3176
MQRCCF_PROGRAM_AUTH_FAILED	3177	MQRC_REOPEN_EXCL_INPUT_ERROR	6100
MQRC_REOPEN_INQUIRE_ERROR	6101	MQRC_REOPEN_SAVED_CONTEXT_ERR	6102
MQRC_REOPEN_TEMPORARY_Q_ERROR	6103	MQRC_ATTRIBUTE_LOCKED	6104
MQRC_CURSOR_NOT_VALID	6105	MQRC_ENCODING_ERROR	6106
MQRC_STRUC_ID_ERROR	6107	MQRC_NULL_POINTER	6108
MQRC_NO_CONNECTION_REFERENCE	6109	MQRC_NO_BUFFER	6110
MQRC_BINARY_DATA_LENGTH_ERROR	6111	MQRC_BUFFER_NOT_AUTOMATIC	6112
MQRC_INSUFFICIENT_BUFFER	6113	MQRC_INSUFFICIENT_DATA	6114
MQRC_DATA_TRUNCATED	6115	MQRC_ZERO_LENGTH	6116
MQRC_NEGATIVE_LENGTH	6117	MQRC_NEGATIVE_OFFSET	6118
MQRC_INCONSISTENT_FORMAT	6119	MQRC_INCONSISTENT_OBJECT_STATE	6120
MQRC_CONTEXT_OBJECT_NOT_VALID	6121	MQRC_CONTEXT_OPEN_ERROR	6122
MQRC_STRUC_LENGTH_ERROR	6123	MQRC_NOT_CONNECTED	6124
MQRC_NOT_OPEN	6125	MQRC_DISTRIBUTION_LIST_EMPTY	6126
MQRC_INCONSISTENT_OPEN_OPTIONS	6127	MQRC_WRONG_VERSION	6128
MQRC_REFERENCE_ERROR	6129		

Table C-3. Integer Values for Extended PCF Events/Reason Codes

EPCF Event	Integer Value	EPCF Event	Integer Value
EXRC_EVENT_BASE	20000	EXRC_OBJECT_NEW	20001
EXRC_OBJECT_CHANGE	20002	EXRC_OBJECT_DELETED	20003
EXRC_Q_MGR_DELETED	20004	EXRC_EXISTING_REGISTRATION	20005
EXRC_NEW_REGISTRATION	20006	EXRC_USER_ACTION	20007
EXRC_SYSTEM_ACTION	20008	EXRC_LIMIT_OVERFLOW	20009
EXRC_DISCOVERY_ERROR	20010	EXRC_DISCOVERY_STARTED	20011
EXRC_CMD_SERVER_DOWN	20012	EXRC_MSG_Q_SERVER_DOWN	20013
EXRC_PROCESS_NOT_EXEC	20014	EXRC_STATE_CHANGE	20015
EXRC_DEAD_LETTER_MESSAGE	20016	EXRC_DISCOVERY_STOPPED	20017
EXRC_PUBLISHER_DOWN	20018	EXRC_CHANNEL_RESET	20019
EXRC_CHANNEL_RESOLVE	20020	EXRC_CHANNEL_START	20021
EXRC_CHANNEL_STOP	20022	EXRC_LISTENER_START	20023
EXRC_LISTENER_STOP	20024	EXRC_SERVICE_START	20025
EXRC_SERVICE_STOP	20026	EXRC_DISCOVERY_IN_PROGRESS	20027
EXRC_LIMIT_REACHED	20028	EXRC_MCA_STARTED	8014
EXRC_MCA_STOPPED	8015	EXRC_MQNODE_COMMAND_FAILED	20100
EXRC_MQNODE_CONNECTED	20101	EXRC_MQNODE_DISCONNECTED	20102
EXRC_CLIENT_CONNECTED	20103	EXRC_CLIENT_DISCONNECTED	20104
EXRC_CLIENT_COMMAND_FAILED	20105	EXRC_MQDB_ERROR	20106
EXRC_CMD_IN_PROGRESS	20107	EXRC_MQNODE_UNREACHABLE	20301
EXRC_NO_HEART_BEAT	20302	EXRC_DELAYED_HEART_BEAT	20303
EXRC_OBJECT_NOT_MANAGED	20304	EXRC_Q_MGR_PROXY	20305
EXRC_MANAGER_DOWN	20306	EXRC_COMMAND_TIMEOUT	20307
EXRC_Q_MGR_NON_PCF	20308	EXRC_MANAGER_NOT_AVAILABLE	20309
EXRC_CUSTOM_EVENT	20310	EXRC_LA_HEARTBEAT	20311
EXRC_DEF_Q_MGR_INCONSISTENT	20501	EXRC_LOG_LOCATION_ERROR	20502
EXRC_LOG_SIZE_ERROR	20503	EXRC_Q_MGR_CREATING	20504
EXRC_NO_Q_MGRS_DEFINED	20505	EXRC_NOT_LICENSED	20506
EXRC_CLIENT_NOT_TRUSTED	20507	EXRC_DB_BACKUP_OK	20600
EXRC_DB_BACKUP_FAILED	20601	EXRC_DB_RESTORE_OK	20602
EXRC_DB_RESTORE_FAILED	20603	EXRC_SQLDB_ERROR	20700
EXRC_SQLDB_RECOVERED	20701	EXRC_MEMALLOC_ERROR	9000
EXRC_MCA_PROCESS_CREATE_FAILED	9001	EXRC_AUTH_NOT_REQUIRED	1000
EXRC_ACCOUNT_PASSWORD_EXPIRED	1001	EXRC_ACCOUNT_DISABLED	1002
EXRC_ACCOUNT_LOCKED	1003	EXRC_ACCOUNT_UNKNOWN	1004
EXRC_ACCOUNT_NOT_LICENSED	1005	EXRC_ACCOUNT_ILLEGAL_USAGE	1006
EXRC_ACCOUNT_ACCESS_DENIED	1007	EXRC_INVALID_SIGNATURE	1008
EXRC_WRONG_SIGNATURE	1009	EXRC_INVALID_SESSION_TOKEN	1010
EXRC_EXPIRED_SESSION_TOKEN	1011	EXRC_INVALID_PASSWORD	1012
EXRC_KERBEROS_AUTH_REQUIRED	1050	EXRC_KERBEROS_AUTH_FAILED	1051
EXRC_CMD_NOT_ALLOWED	1098	EXRC_AUTH_CLEAR_PASSWORD	1099
EXRC_INVALID_ARGUMENT	2001	EXRC_UNEXPECTED_ERROR	2002
EXRC_UNSUPPORTED_COMMAND	2003	EXRC_AUTHORIZATION_UPDATED	10508

Table C-3. Integer Values for Extended PCF Events/Reason Codes

EPCF Event	Integer Value	EPCF Event	Integer Value
EXRC_WRONG_SIGNATURE	1009	EXRC_INVALID_SESSION_TOKEN	1010
EXRC_EXPIRED_SESSION_TOKEN	1011	EXRC_INVALID_PASSWORD	1012
EXRC_KERBEROS_AUTH_REQUIRED	1050	EXRC_KERBEROS_AUTH_FAILED	1051
EXRC_CMD_NOT_ALLOWED	1098	EXRC_AUTH_CLEAR_PASSWORD	1099
EXRC_INVALID_ARGUMENT	2001	EXRC_UNEXPECTED_ERROR	2002
EXRC_UNSUPPORTED_COMMAND	2003	EXRC_AUTHORIZATION_UPDATED	10508

C.2 Locations and Scripts

All WMQ Agent scripts must be located in `[APWMQ_HOME]\scripts`, and all Workgroup server scripts must be located in `[APWMQ_HOME]\groups\[GroupName]\scripts`, where `[APWMQ_HOME]` is your M6-WMQ installation directory and `[GroupName]` is the name of your Workgroup server.

By default, the Workgroup server and WMQ Agent script directories are created when you install M6-WMQ. The script files in the directory have the extension `".sam"`. To activate the scripts, rename them to `".bat"` on Windows, or remove the extension in UNIX.

The following are sample scripts:

45_2224.sam: Adjust queue low threshold when queue low event is received (calls `nsqqw`).

45_2225.sam: Adjust queue high threshold when queue high event is received (calls `nsqqw`).

46_2226.sam: Restart a channel after channel stop event is received.

The following executables are associated with the above script samples:

nsqqw: Program to adjust reporting threshold levels for queue high/low events.

nstrig: Program to format trigger messages for a queue.

pcfparm: Program to parse PCF parameters in the script command line

pcfval: Program to parse PCF value in the script command line

On Windows, the executables are already in the `[APWMQ_HOME]\bin` directory.

On UNIX, the executables must be copied from one of the "scripts" directories to `$APWMQ/bin`

See the `README.txt` file in the scripts directory for additional details.

C.3 How Scripts are Invoked

Scripts are invoked by the WMQ Agent at the node where the event occurred, or at the workgroup server(s) that is managing the node. The script (or executable) must parse the command line to obtain the event parameters. The parameters associated with the event are then passed to the script in the following format:

```
script_name pcfid1=p1 pcfid2=p2 ... pcfidn=pn
```

where:

- `script_name` is the script name (following the script naming convention)
- `pcfidi` is an integer identifying the name of an event parameter
- `pi` is the actual event parameter

Scripts that are invoked by the workgroup server have three additional arguments added to their command line:

`EXCA_MANAGER_NAME` (20006)

Name of the Workgroup server executing the script.

For example, 20006=MQM

`EXCA_MQNODE_NAME` (20007)

Name of the WMQ Agent on which the event occurred.

For example, 20007=LS1

`MQCA_Q_MGR_NAME` (2015)

Name of the queue manager for which the event occurred.

For example, 2015=MY.QMGR

For example, the WMQ Agent script associated with a channel-stopped event might be invoked as follows:

```
46_2283 3501=MY.CHANNEL
```

The integer 3501 represents the event parameter `MQCACH_CHANNEL_NAME`; its value is `MY.CHANNEL` (note that, in reality, there would be several PCF event parameters associated with this event). The same script executed by a Workgroup server would be invoked as:

```
46_2283 20007=LS1 20006=MQM 2015=MY.QMGR 3501=MY.CHANNEL
```

C.4 Script Examples

The examples below demonstrate:

- A UNIX shell script, 46_2283, used to process a channel-stopped event
- The C programs, pcfparm.c and pcfval.c, used to retrieve the PCF ID (pcfparm) and its value (pcfval)
- The deletion of a queue manager from the database.

The nsqmqsc program (executed near the end of the script) *must* run in detached mode (or else the Workgroup server may hang waiting for nsqmqsc to finish). This requires the nsqmqsc command line to start with 'start' on Windows systems or end with the ampersand character '&' on UNIX systems.

Example of UNIX Script to Restart a Channel:

```
#!/bin/ksh
# 46_2283 Script to restart a channel after receiving channel stop event

# Comment out or set next line to null if you don't want script echo lines
OUT=$AUTOPILOT/WMQ/script.out

echo ----- Channel Stop script starting ----- >>$OUT
echo `date` >>$OUT

STOPPED_OK=7
STOPPED_ERROR=8
STOPPED_RETRY=9
STOPPED_DISABLED=10

while (($# > 0))
do
    PCFPARM=$(pcfparm
    $1)
    PCFVAL=$(pcfval $1)
    case $PCFPARM in
        20006)
            echo "-- Workgroup server=$PCFVAL" >>$OUT
            GS=$PCFVAL
            ;;
        20007)
            echo "-- Node=$PCFVAL" >>$OUT
            NN=$PCFVAL
            ;;
        2015)
            echo "-- Queue Manager=$PCFVAL" >>$OUT
            QM=$PCFVAL
            ;;
        1013)
            echo "-- AMQ Msg Error Identifier=$PCFVAL" >>$OUT
            EI=$PCFVAL
            ;;
        1020)
            echo "-- Reason Qualifier: $PCFVAL" >>$OUT
            RC=$PCFVAL
            ;;
        3501)
            echo "-- Channel Name: $PCFVAL" >>$OUT
            CH=$PCFVAL
            ;;
        3505)
            echo "-- Transmit Queue: $PCFVAL" >>$OUT
```

```

        TQ=$PCFVAL
    ;;
3506)
    echo "-- Connection Name: $PCFVAL" >>$OUT
    CO=$PCFVAL
    ;;
3507)
    echo "-- Message Channel Agent: $PCFVAL" >>$OUT
    MA=$PCFVAL
esac
shift
done

echo "Reason Code RC = $RC" >>$OUT

# Do not restart channel if closed with either a zero return code or
# a warning return code
if [ $RC -eq $STOPPED_OK ]
then
echo "Bypass channel restart due to rc STOPPED_OK" >>$OUT
exit 0
fi

# Restart sndr or rcvr channel which had some error
if [ $RC -eq $STOPPED_ERROR ]
then
echo "Restarting channel due to rc STOPPED_ERROR" >>$OUT
fi

# Do not restart channel since it's in retry state
if [ $RC -eq $STOPPED_RETRY ]
then
echo "Bypass channel restart due to rc STOPPED_RETRY" >>$OUT
exit 0
fi

# Do not restart channel if in stopped state (it was manually stopped)
if [ $RC -eq $STOPPED_DISABLED ]
then
echo "Bypass channel restart due to rc STOPPED_DISABLED" >>$OUT exit 0
fi

echo "Executing nsqmqsc start channel command: " >>$OUT
echo "$APWMQ_HOME/bin/nsqmqsc -m$GS -n$NN -q$QM" >>$OUT

echo "start channel ($CH)" > $CH.tst
$APWMQ_HOME/bin/nsqmqsc -m$GS -n$NN -q$QM < $CH.tst > $CH.out &
exit 0

```

pcfparm.c /* C-program to get PCF parameter; returns a numeric string*/

```
#include <stdio.h>
#include <strings.h>
#ifndef TRUE
#define TRUE 1
#endif

void main (int argc, char* argv[]) {
    int ok;
    int len, i;
    char buf[256];
    if(argc < 2) {
        return;
    }
    strcpy(buf, argv[1]);
    len = strlen(buf);
    for(i=0; i < len; i++) {
        if(buf[i] == '=') {
            ok = TRUE;
            buf[i] = 0;
            break;
        }
    }
    if(ok)
    printf(buf);
    return;
}
```

pcfval.c /* C-program to get PCF value following the '=' sign; returns a character string*/

```
#include <stdio.h>
#include <strings.h>
#ifndef TRUE
#define TRUE 1
#endif
void main (int argc, char* argv[]) {
    int ok;int len, i;
    char buf[256];
    if(argc < 2) {
        return;
    }
    strcpy(buf, argv[1]);
    len = strlen(buf);
    for(i=0; i < len; i++) {
        if(buf[i] == '=') {
            ok = TRUE;
            break;
        }
    }
    if(ok)
    printf(&buf[i+1]);
    return;
    return;
}
```

Example 1: UNIX Script Output:

In this example, a sender channel was manually stopped. A sender channel stop event was generated, indicating the stop reason. The output shows that channel is not restarted because the user stopped the channel intentionally.

```
----- Channel Stop script starting -----  
Mon Dec 13 13:04:40 CUT 1999  
-- Queue Manager=JPC  
-- Channel Name: TO_AIXQM  
-- Workgroup server=MQM  
-- Node=AIXIS  
-- Queue Manager=JPC  
-- Channel Name: TO_AIXQM  
-- Reason Qualifier: 10  
-- AMQ Msg Error Identifier=0  
Reason Code RC = 10  
Bypass channel restart due to rc STOPPED_DISABLED
```

Example 2: UNIX Script Output:

In this example, the sender channel TO_AIXQM was started on node AIXIS by a runmqchl command and then the process was killed. A receiver channel stop event was generated and the receiver channel was restarted.

```
----- Channel Stop script starting -----  
Mon Dec 13 13:34:51 CUT 1999  
-- Queue Manager=AIX_QM  
-- Channel Name: TO_AIXQM  
-- Workgroup server=MQM  
-- Node=AIXIS  
-- Queue Manager=AIX_QM  
-- Channel Name: TO_AIXQM  
-- Connection Name: 11.0.0.73  
-- Reason Qualifier: 8  
-- AMQ Msg Error Identifier=9208  
Reason Code RC = 8  
Restarting channel due to STOPPED_ERROR  
Executing nsqmqsc start channel command:  
/opt/nastel/apwmq/bin/nsqmqsc -mMQM -nAIXIS -qAIX_QM
```

Queue Manager Deletion from Database:

Use the following script to delete the object from the database and the actual queue manager from the system.

```
/opt/mqm/bin/endmqm -i $QMGRNAME
/opt/mqm/bin/endmqm -m $QMGRNAME & sleep 5
/opt/mqm/bin/dltmqm $QMGRNAME sleep 10
# Delete the queue manager via Nastel's nsqmqsc utility
echo "CD .." > Nastel.TEMP.txt
echo "UNMANAGE GAMEIPT" >> Nastel.TEMP.txt
echo "CD GAMEIPT" >> Nastel.TEMP.txt
echo "DELETE $QMGRNAME" >> Nastel.TEMP.txt
echo "CD .." >> Nastel.TEMP.txt
echo "MANAGE GAMEIPT" >> Nastel.TEMP.txt
echo "EXIT" >> Nastel.TEMP.txt
/opt/nastel/apwmq/bin/nsqmqsc -mMQM -nGAMEIPT < Nastel.TEMP.txt
```

Appendix D: Differences Between Platforms

M6-WMQ provides the same basic features on most supported platforms. On some platforms, however, there are minor differences. These differences are summarized in this appendix.

D.1 Supported Platforms

M6-WMQ provides basic WebSphere MQ monitoring and configuration management for the platforms listed in Table D-1:

Platform	Notes
UNIX	All versions supported by M6-WMQ, including HP-UX, Sun Solaris, AIX, and Linux.
System component cannot interact	Communication problem, or problem local to one of the components.
No response to user actions	WebSphere MQ not receiving user actions, or mishandling them.
WebSphere MQ information is not collected	A workgroup server is not successfully connecting to or interacting with a WMQ Agent.

D.1.2 Specific Features by Platform

Most M6-WMQ features are the same on all platforms. Platforms supported by M6-WMQ are limited to those platforms supported by WebSphere MQ 7.4. These features are listed in Table D-2.

Platform	Monitoring	Configuration Management	Management via Agent
HP-UX	YES	YES	YES
Sun Solaris	YES	YES	YES
AIX, Linux	YES	YES	YES
Windows	YES	YES	YES
OS/400	YES	YES	YES
z/OS	YES	YES	YES

This Page Intentionally Left Blank

Appendix E: M6-WMQ Component Summary

M6-WMQ provides the same basic features on most supported platforms. On some platforms, however, there are minor differences. These differences are summarized in this appendix.

Appendix E lists the following:

- M6-WMQ components
- The names of their executables
- The platforms they run on
- The command-line arguments they accept (required and optional)

E.1 Workgroup Server

The workgroup server is an application named `nsqmgr (.exe)`.

E.1.1 Platforms

The workgroup server is available on the following platforms:

- Windows 2000 or later
- UNIX

E.1.2 Command Line Arguments

Table E-1 lists all command-line arguments applicable to the workgroup server.

Table E-1. Workgroup Server Command Line Arguments

Arguments	Description
Required Arguments:	
-m<GroupName>	Load workgroup server database.
Optional Arguments:	
-?	Display workgroup server usage/help message information.
-a +a(1,2,3)	Disable/enable user authentication. Default: authentication off
+a1	Authenticate user via local system security.
+a2	Authenticate user via AutoPilot M6 domain security.
+au	Authenticate user via AutoPilot M6 (also enables +u3 and +a2). If authentication for a user passes, Autopilot will return the LDAP user groups to the Workgroup Server which then stores them in the Permits V3 Database to be used by Workgroup Server at runtime for authorization. For all of this to work correctly, the LDAP groups must be added manually to APODWSM (Security Manager) with the rights each group should have.
-b (UNIX only)	Runs the workgroup server in the background as a daemon.
-C	Disables user ID conversion to uppercase. By default, the user ID is converted to uppercase and passed to M6 WMQ agent for further checks and mapping.
-e	Reuse TCP/IP address when binding. If port is in use or still lingering, workgroup can forcibly reclaim it. Default: off
-f	Perform full queue manager discovery upon connection with an agent.
-F	Generate alter events during full queue manager discovery.
-g	Disables the Nagle algorithm for send coalescing on TCP/IP connection. This option is used to improve TCP performance.
-i<buff_size>	Sets send/receive TCP buff size, in KB; for example, 2 = 2 KB.
-IgnoreEnv	Ignore all environment variables and only use variables supplied
-k	Set TCP KEEPALIVE option for all connections. This option is needed when workgroup server communicates with agents and/or clients thru a firewall. Default: off
-l<Cmd. Limit>]	Client pending command limit. Sets backlog limit of response messages being sent to client. If there are more WMQ objects than the default of 15000, the default needs to be overridden.
-n<ServerAlias>	Name used in messages originated by workgroup server to other M6-WMQ processes.
-po	Runs WGS with Permits V3 in the mode of no pre-load and reading each record from the database. (See +u3 below.)
-r	Set WMQ local queue and channel properties and status refresh interval (in seconds) for updating cached copy of local queue configuration from queue manager. Default: 10 seconds.
-R	Set WMQ Object cache refresh interval (Sec.)
-ru	Set AP authorization (+au)/Permits cache refresh interval. Interval is in minutes (default is 360 for 3 hours)
-S	Run in single-threaded mode.
-safe	The safe mode startup option starts the Workgroup Server without it managing any MQ Nodes. This is a temporary setting that does not modify the settings of the nodes usually managed in the Workgroup Server. Useful for traces when you want to see the results of just one MQ Node used by the Workgroup Server.
-t	Turns on tracing of PCF messages.
-ta	Turns on user authorization trace. Default: off
-tc	Turns on communication trace. Default: off
-tn	Turns on connection trace. Default: off
-ts	Turns on tracing of SQL database operations.
-tu	Turns on authorization trace. Default: off

Table E-1. Workgroup Server Command Line Arguments

Arguments	Description
+u/-u	Enables/disables user authorization checks for letting users submit commands to WMQ command server. -u authorization information is read from permits.ini file, +u2 has been deprecated and +u3 retrieves user authorization information from security SQL database. Default: off
+u3	Enables user authentication using PermitsV3 SQL DB. PermitsV3 data is preloaded at startup and when APODWSM is refreshed. All Permits V3 supported databases support regex when run in this mode. To run WGS with PermitsV3 in the previous mode of no pre-load and reading each record from the database, you must use the -po flag.
-wc	Disable writing to Accounting table
-wd	Disable writing to Auditing table
-we	Disable writing to Event table
-ws	Disable writing to Statistics table
-x<MigGroupName>	Migrate workgroup and node definition from <MigGroupName>.
-auditinfo	Enables Auditing statements to nsqmgr's output
SQL Database-Related	
-D<DataSourceName>	Load workgroup server SQL database using data source
-P<DbPasswd>	Data source name password
-U<LoginID>	Database name login ID
-h1<SchemaName>	Database schema name. Default :NASTEL_M6WMQ
-pD<DataSourceName>	Load Permits SQL database using data source
-pU<LoginID>	Permits data source name login id
-pP<DbPasswd>	Permits data source name password
-h2<SchemaName>	Permits database schema name. Default: NASTEL_PERMITS
Windows only arguments:	
-console	Start workgroup server as a console program.
-install	Install workgroup server service.
-remove	Remove the workgroup server service.

	NOTE:	On Windows, the workgroup server requires the M6-WMQ environment variable to be set to the full path of the WebSphere MQ installation directory (by default, c:\nastel\apwmq).
---	--------------	--

E.2 WMQ Agent

The WMQ Agent is an application named nsqmq (.exe).

E.2.1 Platforms

The WMQ Agent is available for all platforms currently supported by M6-WMQ, as listed below:

- Windows 2000/XP/Vista
- OS/400
- UNIX (Linux, Solaris, AIX, HP_UX)
- z/OS.

E.2.2 Command-Line Arguments

Table E-2 lists all command-line arguments applicable to the WMQ Agent.

Table E-2. WMQ Agent Command-Line Arguments	
Arguments	Description
Optional Arguments:	
-Q<mask>	Manage only queue manager whose names start with <mask> string.
-D<name.com>	DNS name to create fully qualified host name.
-m<WorkgroupName>	Specifies the workgroup server to which the WMQ Agent must register.
-m* (asterisk)	Registers the WMQ Agent under all available workgroup servers. You can also use the asterisk to represent partial names.
-t +t	Turns on/off PCF protocol trace. Default: trace off
-tu +tu	Turns on/off authorization trace. Default: trace off
-x +x	Turns on/off tracing for TCP/IP packets. Default: off
-e	Reuse TCP/IP address when binding. If port is in use or still lingering, workgroup can forcibly reclaim it. Default: off
+E	Enable encryption when communicating with the Workgroup Server.
-f	Force full discovery for initial discovery. Default: off
-r<retries>	Number of attempts WMQ agent makes to bind to an available TCP Port. It starts with specified port followed by successive port numbers until. A port is found that is not in use. Default: 1
-n<AliasNodeName>	Assigns a new name to be used when registering with workgroup servers. This name will override the name of the local host.
+u -u	Enables/disables authorization checks for letting users submit commands to the WebSphere MQ command server. +u gives all users effective authority over the WMQ Agent. Default is: Authorization checked off.
+q -q	Enables/disables queue manager verification. This option is needed to test if queue manager is properly configured for M6 WMQ agent. Default: disabled.
-c<subtasks> (z/OS only)	Specifies the maximum number of z/OS MTF subtasks.
-P +P (z/OS only)	Run PCF Command Server as MTF subtask (-P) or in independent address space (+P). Default: MTF subtask.
-R (z/OS only)	Queue manager connect retry time (secs) when PCF Command Server running as MTF subtask. Default: 30 sec.
+z -z	Enables/disables use of DNS (Domain Name Server) for host name resolution. Default: enabled. Disabling of DNS should be used on systems where an Object Detached event states in its Description field that a host name cannot be resolved by the WMQ Agent.
+i -i	Enable/disable reporting node IP address to workgroup server. Default: on
-p<time>	Time interval to send registration to workgroup server periodically (in minutes). Sent only at startup time.
-b<Buf.Size>	Set TCP send/receive buffer size in KB. Default: 64KB
+g -g	Enables/disables TCP Nagle algorithm for send coalescing on TCP/IP connection. Improves TCP performance. Default: disabled.
-s<port/service_name>	Specifies the port or service name to which the WMQ Agent will bind. Default: 5000
-v	Verify queue manager and communication only.
-C	Disables user id conversion to uppercase. By default, the user id is converted to uppercase and passed to WMQ queue manager as an alternate user id.

Table E-2. WMQ Agent Command-Line Arguments	
Arguments	Description
[-l<Que. Limit>]	TCP buffering queue limit. This option sets soft limit to buffer responses from WMQ to be sent to the workgroup. The smaller the buffer size, the more responsive the GUI will be. Default: 10
-w	Maximum wait time (msec) for socket read/write event. Default: 1000
-tN	Enable debug trace of name list discovery. Default: off
-N	Enable special name list discovery. Use on z/OS when thousands of queues, channels, processes, namelists, or auth info objects are defined and IBM 64KB PCF message size limit exists. Default: Disabled
-f	Force full discovery for initial discovery. Default: off
-?	Display usage/help messages.
Windows only arguments: (Used only when running this program from the command prompt.)	
-install	Installs the WMQ Agent as a service (if not done during installation.)
-remove	Removes the WMQ Agent as a service.
-console	Runs the WMQ Agent as a console application.

E.3 Deleted

Table E-3. Deleted

E.4 MMF Server

The MMF (Message Management Facilities) Server is an application named `nsqmsg (.exe)`. A second application, `endmcs (.exe)`, is used to stop the MMF Server.

E.4.1 Platforms

The MMF Server is available for all platforms currently supported by M6-WMQ:

- Windows 2000 or later
- OS/400
- UNIX
- z/OS

E.4.2 Command-Line Arguments

Table E-4 lists all command-line arguments applicable to the MMF Server.

Table E-4. Start MMF Server Command-Line Arguments	
Arguments	Description
Optional Arguments:	
-m<Qmgr_Name>	Name of the WebSphere MQ queue manager where MMF is required.
-r<N>	Set reconnect retry number.
-g<Grace>	Set grace period in seconds between reconnects.
-a	Enable command auditing.
-b	Enable binary representation of message body.
-c	Enable conversion on MQGET.
-t	Enable EPCF protocol trace.
-?	Display usage information.
Windows only arguments: (Used only when running this program from the command prompt)	
-install	Installs MMF Server as a service (if this has not already been done during installation.)
-remove	Removes MMF Server as a service.
-console	Runs MMF Server as a console application.

	NOTE: “-m*” will attach MMF Server to all QMGRs on Windows platforms.
--	--

Table E-5. Stop MMF Server Command-Line Arguments	
Arguments	Description
Required Arguments:	
QueueManager	Queue Manager whose MMF Server is to be stopped.
Optional Arguments:	
CommandQueue	Command Queue to send stop command to. Default: NASTEL.MMF.ADMIN.COMMAND.QUEUE

E.5 MQ Event Publisher

The MQ Event Publisher is an application named nsqpub (.exe). A second application, pubend (.exe) is used to stop the MQ Event Publisher.

E.5.1 Platforms

The MQ Event Publisher is available on all platforms currently supported by M6-WMQ, as listed below:

- Windows or later
- z/OS
- OS/400
- UNIX

E.5.2 Command-Line Arguments

Table E-6 lists all command-line arguments applicable when starting the MQ Event Publisher.

Table E-6. Starting MQ Event Publisher Command-Line Arguments

Arguments	Description	Default
<code>-f<perm_subscriber_file></code>	Path to permanent subscription file. PubSub distributes according to subscriptions. M6-WMQ agent subscription request is stored in temp subscription file. This file is stored in the same directory and has name in the form QMGR.ini. Other applications may subscribe to events only thru permanent subscription file.	\subscrib (UNIX) dd:subscrib (z/OS)
<code>-i<temporary_subscriber_file_HLQ></code>	Defines high level qualifier (HLQ) of temporary subscription files. z/OS: If specified, structure of the name is HLQ.QMGRName.INI. If not specified, structure of the name is UserID.QMGRName.INI. Other platforms: If specified, structure of the name is HLQ.MGRName.ini. If not specified, structure of the name is QMGRName.ini.	An empty string
<code>-e<SystemEventQ></code>	EventQ. Defines the name of the event queues. These queues are inbound for MQ Event Publisher. All queue managers must have same event queue name.	NASTEL.PUBSUB. EVENT.QUEUE
<code>-d<DeadLetterQ></code>	Defines the default name of the dead letter queues (DLQs) for all queue managers.	SYSTEM.DEAD. LETTER.QUEUE
<code>-w<WaitInterval></code>	Defines the wait interval, in milliseconds, on MQGET calls.	30000
<code>-l<DisableListenDLQ></code>	If specified, will not listen on DLQ.	Listens on DLQ.
<code>-q<DLQTimeRate></code>	Defines time interval (in milliseconds) that must elapse before MQ Event Publisher will report on a subsequent identical DLQ message.	300000
<code>-n<DLQCountRate></code>	Defines number of messages that must elapse before MQ Event Publisher will report on a subsequent identical DLQ message.	1000
<code>-k<DLQDeleteRate></code>	Defines time interval (in milliseconds). During this time interval the information about a reported DLQ message will be kept for future frequency testing.	3000000
<code>-v<DLQEventsMax></code>	Defines maximum number of unique DLQ events.	100
<code>-o<ShortReconnMax></code>	Defines number of short MQ reconnections.	5
<code>-h<ShortReconnInt></code>	Defines time interval (milliseconds) for short MQ reconnect retries.	5000
<code>-g<LongReconnInt></code>	Defines time interval (milliseconds) for long MQ reconnect retries.	60000
<code>-c<SubscribersMax></code>	Defines the maximum number of subscribers.	100
<code>-b<QMGRsMax></code>	Defines maximum number of queue managers. (Maximum number of queue managers is 50.) z/OS only: One queue manager requires three threads per connection which includes one thread for Publisher parent process and two DLQ threads. This translates to 1M ~ 4M of storage utilization by PubSub process. Since default maximum number of queue managers is 20, due to z/OS constraints (REGION=), PubSub process should be run with <code>-b</code> = actual number of queue managers.	20
<code>-r<UDPConnRetry></code>	Defines the number of retries on a UDP connection.	5
<code>-u</code>	If specified, will reuse the UDP address. Publisher uses UDP port 5500 (default).	Off
<code>-s<ServiceName></code>	Defines the UDP port to send a SHUTDOWN_MGR command event to the MQ Event Publisher. The Publisher listens on this port number. It must be the same value as <code>-sServiceName</code> , specified when the Publisher was started.	5500

Table E-6. Starting MQ Event Publisher Command-Line Arguments

Arguments	Description	Default
-p	Enable duplicate event suppression. Default: Disabled. NOTE: Use -p, -a, and -x if you typically get storms of duplicate events in a short period of time and you don't need to be made aware of every instance. The -x time interval expiration or the -a maximum counter reached will cause a reset of both the timer and the counter, a new one-time appearance of the event, and another suppression interval.	Disabled
-a	The maximum number of duplicate events that will be suppressed. Default: 100. NOTE: Use -p, -a, and -x if you typically get storms of duplicate events in a short period of time and you don't need to be made aware of every instance. The -x time interval expiration or the -a maximum counter reached will cause a reset of both the timer and the counter, a new one-time appearance of the event, and another suppression interval.	100
-x	The maximum time interval (in seconds) during which duplicate events are suppressed. Default: 60. NOTE: Use -p, -a, and -x if you typically get storms of duplicate events in a short period of time and you don't need to be made aware of every instance. The -x time interval expiration or the -a maximum counter reached will cause a reset of both the timer and the counter, a new one-time appearance of the event, and another suppression interval.	60
-t	If specified, turns tracing on.	Off
-P/+P	Enable multi- or single-threaded process	+P; Single-threaded
-S	Do not use shared memory to shutdown process.	Off
-W (z/OS only)	Enable WTO logging.	Off
-?	Display usage information.	

Table E-7 lists all command-line arguments applicable when stopping the MQ Event Publisher

Table E-7. Stopping MQ Event Publisher Command-Line Arguments

Arguments	Description	Default
-r<UDPConnRetry>	Defines the number of retries on a UDP connection.	5
-u	If specified, will reuse the UDP address.	Off
-l<LstnrSrvName>	Defines the UDP listening port used by the pubend program for receiving PCF messages.	6000
-s<ServiceName>	Defines the UDP port to send a SHUTDOWN_MGR command event to the MQ Event Publisher. The Publisher listens on this port number. It must be the same value as -sServiceName, specified when the Publisher was started.	5500
-h<HostName>	Defines MQ Event Publisher host name.	Localhost
-i<HostIPAddress>	Defines MQ Event Publisher host IP address.	127.0.0.1 (localhost)
-t	If specified, turns tracing on.	Off
-?	Display usage information.	

E.6 M6-WMQ z/OS PCF Command Server

The M6-WMQ z/OS PCF Command Server is an application named `nsqmvts (.exe)` used to distinguish from IBM's Command Server.

E.6.1 Platforms

- The PCF Command Server runs on z/OS for direct management of z/OS with WMQ Agent.
- Windows 2000 or later.

E.6.2 Command-Line Arguments

Table E-8 lists all command-line arguments applicable to the PFC Command Server.

Table E-8. M6-WMQ z/OS PCF Command Server Command-Line Arguments		
Arguments	Description	Default
Required Arguments:		
<code>-m<QmgrName></code>	Specified the name of the queue manager.	None
Optional Arguments:		
<code>-p<PCF_Command_Queue></code>	The name of the PCF command queue to which the WMQ Agent submits a PCF command for the M6-WMQ command server.	SYSTEM.ADMIN.COMMAND.QUEUE
<code>-r<Reply_Queue></code>	The queue that receives the reply from the WebSphere MQ command server for processing the M6-WMQ command server.	APWMQ.PCF.MQSC.SERVER
<code>-w<Timeout_ms></code>	The interval, in milliseconds, that the M6-WMQ command server waits for the WebSphere MQ command server to WebSphere MQ command server before considering the command request as having timed out.	6000
<code>-t</code>	Turns the trace option on.	off
<code>-c<MQSC_System_Command_Queue></code>	The queue for submitting the command to the WebSphere MQ command server.	SYSTEM.COMMAND.INPUT
<code>-R<Connect_Retry_Time></code>	The interval, in seconds, that the PCF command server will attempt to reconnect to the queue manager.	30
<code>-?</code>	Display usage information	

E.7 MQSC Utility

The MQSC utility is an application named `nsqmqsc (.exe)`.

E.7.1 Platforms

The MQSC utility is available on the following platforms:

- Windows 2000 or later
- UNIX

E.7.2 Command-Line Arguments

Table E-9 lists all command-line arguments applicable to the MQSC utility.

Table E-9. MQSC Utility Command-Line Arguments	
Arguments	Description
Required Arguments:	
<code>-m<GroupName></code>	Workgroup server name.
Optional Arguments:	
<code>-n<NodeName></code>	Node name.
<code>-q<QmgrName></code>	Name of queue manager.
<code>-d [-r] [FileName]</code>	Dump the database, in MQSC format, to the console or redirect to the output file. Cannot be used with <code>-f</code> . (See <code>-r</code> below.)
<code>-r</code>	This option adds REPLACE to every DEFINE statement, to force changes when the script is applied to a queue manager. Use with <code>-d</code> .
<code>-f<FileName></code>	Restore the database for the given node and queue manager from the given MQSC file. Cannot be used with <code>-d</code> .
<code>-v</code>	Enable verbose mode.
<code>-t</code>	Enable debug trace.
<code>-U</code>	Username for authenticating with workgroup server. Default: current user.
<code>-P</code>	Password for authenticating with workgroup server. Default: current user.
<code>-R</code>	Security realm for authenticating with workgroup server. Default: security realm for current user.
<code>-a</code>	Don't authenticate with workgroup server. Connect to workgroup server as GUEST user.
<code>-w<WaitInterval></code>	Number of seconds to wait for reply messages. Default: infinite.

E.8 nsqping (Ping) Utility

The Ping Utility allows for testing of WebSphere MQ response time and WebSphere MQ links. The Ping Utility can be used only with local queue managers, but it can test links to remote queue managers. To use the utility, the NSQPING symbol must be installed.

E.8.1 Platforms

The nsqping utility is available on the following platforms:

- Windows 2000 or later
- UNIX
- z/OS

E.8.2 Command-Line Arguments

Table E-10 lists all command-line arguments applicable to the MQSC utility.

Table E-10. Ping Utility Command-Line Arguments		
Arguments	Description	Default
Required Arguments:		
-m<QmgrName>	Specifies the name of the local queue manager.	N/A
-q<QueueName>	Queue name of any type.	N/A
Optional Arguments:		
-b<BatchSize>	Number of messages in a batch.	10
-s<Bytes>	Size, in bytes, of generated messages.	512
-d<Interval>	Interval, in seconds, that ping messages are generated. An interval of 0 (zero) indicates to ping only once.	0
-w<Wait>	Wait interval, in milliseconds, before timeout.	60000
-e<Expiry>	Message expires after Expiry time in 0.1 seconds.	Unlimited
-u<AltUser>	Issue ping on behalf of another user.	N/A
-ccoaa	Generate confirm-on-arrival report.	False
-ccod	Generate confirm-on-delivery report.	False
-cexp	Generate confirm-on-expiry report.	False
-cexc	Generate confirm-on-exception report.	False
-cdlq	Generate message that will go onto DLQ if undelivered.	False
-h<HostName>	Host name of the target AutoPilot M6 Process Wrapper	localhost
-p<Port>	Port number of the target AutoPilot M6 Process Wrapper	6000
-v<0 1 2>	Set verbose mode to 0 – summary; 1- reports; 2- all	0

E.9 nsqcode Utility

The nsqcode utility is used to identify and define WebSphere MQ and M6-WMQ system error codes. If the code you received is not listed, contact either IBM or Nastel support for further assistance.

Detailed lists of current error codes and messages can be found in Appendices F, G, and H in this guide.

E.9.1 Platforms

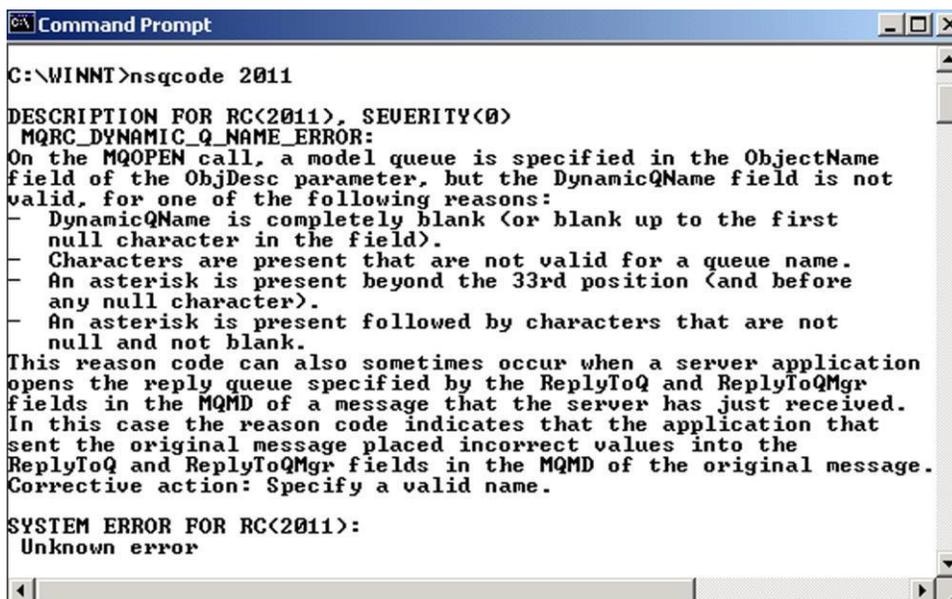
The MQSC utility is available on the following platforms:

- Windows 2000 or later
- OS/400
- UNIX
- z/OS

E.9.2 Additional Information: Using nsqcode

To identify error definitions perform the following steps.

1. At the command prompt go to your operating system directory (example: `C:\[winnt]`).
2. Type: `nsqcode <code>`, where code = error code received.
(example: `nsqcode 2011`)
3. The Error code and its definition as well as possible variations will be displayed as shown below.



```

C:\WINNT>nsqcode 2011

DESCRIPTION FOR RC(2011), SEVERITY(0)
MQRC_DYNAMIC_Q_NAME_ERROR:
On the MQOPEN call, a model queue is specified in the ObjectName
field of the ObjDesc parameter, but the DynamicQName field is not
valid, for one of the following reasons:
- DynamicQName is completely blank (or blank up to the first
  null character in the field).
- Characters are present that are not valid for a queue name.
- An asterisk is present beyond the 33rd position (and before
  any null character).
- An asterisk is present followed by characters that are not
  null and not blank.
This reason code can also sometimes occur when a server application
opens the reply queue specified by the ReplyToQ and ReplyToQMgr
fields in the MQMD of a message that the server has just received.
In this case the reason code indicates that the application that
sent the original message placed incorrect values into the
ReplyToQ and ReplyToQMgr fields in the MQMD of the original message.
Corrective action: Specify a valid name.

SYSTEM ERROR FOR RC(2011):
Unknown error
  
```

Figure E-1. nsqcode Definitions

E.10 Connection Manager

The Connection Manager is an application named `nsqcm (.exe)`.

E.10.1 Platforms

The Connection Manager is available on the following platform:

- Windows 2000 or later
- Linux

E10.2 Command-Line Arguments

Table E-11 lists all command-line arguments applicable to the Connection Manager.

	NOTE:	The +a parameter <i>cannot</i> be used when running in conjunction with AutoPilot M6. Do not change the default -a setting.
---	--------------	---

Table E-11. Connection Manager Command-Line Arguments		
Arguments	Description	Default
Optional Arguments:		
-Q<mask>	Manage only queue managers whose names start with <mask> string.	
-D<name.com>	DNS name to create fully qualified host name.	
-m<WorkgroupName>	Specifies the workgroup server to which the Connection Manager must register.	
-m* (asterisk)	Registers the Connection Manager under all available workgroup servers. You can also use the asterisk to represent partial names.	
-t +t	Turns on/off PCF protocol trace.	Trace off
-x +x	Turns on/off tracing for TCP/IP packets.	Off
-e	Reuse TCP/IP address when binding. If port is in use or still lingering, workgroup can forcibly reclaim it.	Off
-r<retries>	Number of attempts WMQ agent makes to bind to an available TCP Port. It starts with specified port followed by successive port numbers until a port is found that is not in use.	1
-n<AliasNodeName>	Assigns a new name to be used when registering with workgroup servers. This name will override name of the local host.	
+u -u	Enables/disables authorization checks for letting users submit commands to the WebSphere MQ command server. +u gives all users effective authority over the WMQ Agent.	Authorization checked off
+q -q	Enables/disables queue manager verification. This option is needed to test if queue manager is properly configured for M6 WMQ agent. Default: disabled.	Disabled
+z -z	Enables/disables use of DNS (Domain Name Server) for host name resolution. Disabling of DNS should be used on systems where an Object Detached event states in its Description field that a host name cannot be resolved by the WMQ Agent.	Enabled
+i -i	Enable/disable reporting of node IP address to workgroup server.	On
-p<time>	Time interval to send registration to workgroup server periodically (in minutes).	Sent only at startup time.

Table E-11. Connection Manager Command-Line Arguments

Arguments	Description	Default
-b<Buf.Size>	Set TCP send/receive buffer size in KB.	64KB
+g -g	Enable/disable the Nagle algorithm for send coalescing on TCP/IP connection. This option improves TCP performance.	Disabled
-s<port service_name>	Specifies the port number or service name to which the Connection Manager will bind.	5000
-v	Verify queue manager and communication only.	
-C	Disables user id conversion to uppercase. By default, the user id is converted to uppercase and passed to WMQ queue manager as an alternate user id.	
[-l<Que.Limit>]	TCP buffering queue limit. This option sets soft limit to buffer responses from WMQ to be sent to workgroup. The smaller buffer size, the more responsive the GUI will be.	10
-nosyevt	Bypasses collecting system event queue information	
-w	Maximum wait time (msec) for socket read/write event.	1000
-tN	Enable debug trace of name list discovery.	Off
-N	Enable special name list discovery. Use on z/OS when thousands of queues, channels, processes, namelists, or auth info objects are defined and IBM 64KB PCF message size limit exists.	Disabled
-?	Display usage.	
Windows only arguments: (Used only when running this program from the command prompt)		
-install	Installs the Connection Manager as a service (if this has not already been done during installation.)	
-remove	Removes the Connection Manager as a service.	
-console	Runs the Connection Manager as a console application.	

E.11 Log Adapter

The log adapter is an application named `nsqadp (.exe)`.

E.11.1 Platforms

The log adapter is available on the following platforms:

- Windows 2000 or later
- UNIX

E.11.2 Command-Line Arguments

Table E-12 lists all command-line arguments applicable to the log adapter.

Table E-12. Log Adapter Command Line Arguments	
Arguments	Description
Required Arguments:	
<code>-m<GroupName></code>	Specifies the name of the workgroup to connect to.
Optional Arguments:	
<code>-?</code>	Display usage/help message
<code>-U</code>	Username for authenticating with workgroup server. Default: current user
<code>-P</code>	Password for authenticating with workgroup server. Default: current user
<code>-R</code>	Security realm for authenticating with workgroup server. Default: security realm for current user.
<code>-a</code>	Don't authenticate with workgroup server. Connect to workgroup server as GUEST user.
<code>-i</code>	Enable ITO environment. Default: Enable Tivoli environment.
<code>-t<TimeOut></code>	Interval, in msec, between attempts to connect to workgroup server. Default: 10000 (10 seconds)
<code>-l [LogName]</code>	Write messages to log file. Defaults: For ITO: if no <code>-l</code> , write events using <code>opcmsg</code> For ITO and Tivoli: if <code>-l</code> with no <code>LogName</code> , default log is: Windows: ITO: <code>\\usr\OV\nsq\log\nsqadp_GM.log</code> Tivoli: <code>\\temp\group_name.log</code> UNIX: ITO: <code>/var/nsq/log/nsqadp_GM.log</code> Tivoli: <code>/tmp/group_name.log</code>
<code>-n<node host both></code>	Include MQ node name, host name, or both in event messages. Default: node
<code>-s (UNIX only)</code>	Write to UNIX syslog. Not compatible with <code>-i</code> and <code>-l</code> .
<code>-F<n> (UNIX only)</code>	syslog facility index, 0-7, for facility <code>LOG_LOCAL<n></code> . Default: 0
<code>-b (UNIX only)</code>	Runs the Log Adapter in the background as a daemon

Table E-12. Log Adapter Command Line Arguments

Arguments	Description
-r<all nn none>	Append reason code description text: all: Entire text of descriptions <nn>: first <nn> bytes of descriptions none: Don't append reason code descriptions Default: <nn> = length of message ID
-f<cc>	Use field delimiter <cc> in the messages, maximum 2 chars. Use -fsp to use space character as delimiter. Default: ':'
-h	Interval, in seconds, to send heartbeat message if no event message was sent in the last <Interval> seconds. Default: 0 (don't send heartbeat message)
-d	Enable debug trace
-?	Display usage information
Windows only arguments: (Used only when running this program from the command prompt)	
-install	Installs Log Adapter as a service.
-remove	Removes Log Adapter as a service.
-console	Runs Log Adapter as a console application.

E.12 Message Management Utility

The message management utility is an application named `nsqmmfcl (.exe)`.

E.12.1 Platforms

The message management utility is available on the following platforms:

- Windows 2000 or later
- UNIX
- OS/400
- z/OS

E.12.2 Command-Line Arguments

Table E-13 lists all command-line arguments applicable to the message management utility.

Table E-13. Message Management Utility Command Line Arguments	
Arguments	Description
Required Arguments:	
<code>-m<GroupName></code>	Specifies the name of the workgroup to connect to.
Optional Arguments:	
<code>-n<NodeName></code>	Access queues from named WMQ Node.
<code>-q<QmgrName></code>	Access queues from named queue manager.
<code>-U<User></code>	Username for authenticating with workgroup server. Default: current user
<code>-P<Password></code>	Password for authenticating with workgroup server. Default: current user
<code>-R<Realm></code>	Security realm for authenticating with workgroup server. Default: security realm for current user
<code>-a</code>	Don't authenticate with workgroup server. Connect to workgroup server as a guest user.
<code>+v -v</code>	Enable/Disable verbose mode. Default: Enable

E.12.2 Message Management Commands

The following topics are discussed in this section:

- Requirements
- Node and Queue Manager Selection
- Command Descriptions
- Possible Problems.

E.12.3.1 Requirements

For the utility to work, the following programs must be running:

- Command server and queue manager on the nodes of interest
- M6-WMQ workgroup server
- M6-WMQ agents and message servers on the nodes of interest.

E.12.3.2 Node and Queue Manager Selection

You can select the MMF node and queue manager of interest in one of two ways:

- From the command line
- By navigating from within the utility.

Selection from the command line

Enter the MMF command as follows:

```
nsqmmfcl -mGroupName -nNodeName -qQmgrName [options]
```

Selection by navigation within the utility

Navigating among nodes and queue managers is similar to the nsqmqsc utility. At a command prompt (Figure E-2), enter:

1. `nsqmmfcl -mMQM`
2. `dir` (to display the managed nodes)
3. `cd NodeName` (to select a node to work with)
4. `dir` (to display the queue managers in the selected node)
5. `cd QmgrName` (to select a queue manager to work with)
6. `dir` (to display the queues)
7. You are ready to enter any of the commands described in the next section.

```

Command Prompt - nsqmmfcl - mMQM
1 → >nsqmmfcl -mMQM
<C>Copyright 1995 - 2000, Nistel Technologies Inc.
Reproduction, adaptation, or distribution without prior
written permission is prohibited, except as allowed under
the copyright laws.

Sysname("Windows_NT")
NodeName("bebop")
Release("6.0.6004")
User ID("")

2 → \MQM:\> dir
NODE NAME                                STATUS
-----
BLANDEV                                  UNKNOWN
BEBOP                                    ACTIVE
CHARO7                                   UNKNOWN
HPITANIU                                 ACTIVE
NEON                                      UNKNOWN
Q03                                       ACTIVE
ROB                                       ACTIVE
SUNFIRE                                  ACTIVE
TITANIC                                  ACTIVE
WIN2003                                  ACTIVE

3 → \MQM:\> cd BEBOP
4 → \MQM\BEBOP:\> dir
QUEUE MANAGER NAME                       STATUS
-----
SPITEST2                                  UNKNOWN
TEST1                                     STOPPED
TEST2                                     UNKNOWN

5 → \MQM\BEBOP:\> cd TEST1
6 → \MQM\BEBOP\TEST1:\> dir
LOCAL QUEUE NAME                          DEPTH    MAX
-----
AJMS                                        1         5000
AJMS.REPLY                                 4         5000
NASTEL.APIM.CONFIG                         2         5000
NASTEL.APIM.MESSAGE                       0         5000
NASTEL.EVENT.QUEUE                        0         5000
NASTEL.MMF.ADMIN.COMMAND.QUEUE           0         5000
NASTEL.MMF.AUDIT.QUEUE                    0         5000
NASTEL.PUBSUB.EVENT.QUEUE                 0         5000
CLEC                                       0         5000
CLEC2                                      0         5000
SYSTEM.ADMIN.ACTIVITY.QUEUE               0         5000
SYSTEM.ADMIN.COMMAND.QUEUE                0         3000
SYSTEM.ADMIN.LOGGER.EVENT                  0         3000
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE            0         5000
SYSTEM.QUIN.DATA.QUEUE                     84        99999999
SYSTEM.CHANNEL.INITQ                       0         1000
SYSTEM.CHANNEL.SYNCQ                       3         20000
SYSTEM.CICS.INITIATION.QUEUE               0         1000
SYSTEM.CLUSTER.COMMAND.QUEUE               0         99999999
SYSTEM.CLUSTER.REPOSITORY.QUEUE           1         99999999
SYSTEM.CLUSTER.TRANSMLI.QUEUE              0         99999999
SYSTEM.DEAD.LETTER.QUEUE                   0         99999999
SYSTEM.DEFAULT.INITIATION.QUEUE            0         1000
SYSTEM.DEFAULT.LOCAL.QUEUE                 0         5000
SYSTEM.PENDING.DATA.QUEUE                  0         99999999
TEST1                                      0         5000
TEST2                                      0         8888
TEST3                                      0         5000
X                                           0         5000
XFER.CONFIRMATION.RECEIUE                  0         5000
test                                       0         5000
test1                                      1         5000
test2                                      1         5000
testReply                                  0         5000

7 → \MQM\BEBOP\TEST1:\> _

```

Figure E-2. MMF Command Line Interface

E.12.3.1 Command Descriptions

The following commands can be entered at the **nsqmmfcl** command prompt once you have established the node and queue manager scope. When a command completes, a **Command completed** message is displayed with a reason code (RC). If the command completes successfully, the reason code is zero; displayed as **RC (0)**. However, if the command is not successful, the appropriate reason code is displayed. To display a description of the reason code, use the **mqrcc** command described below.

	NOTE:	The node, queue manager, and queue names are case sensitive and if lower case must be surrounded by single quotes.
---	--------------	--

COPY – This command copies specific messages or a batch of messages from one queue to another. The messages stay in the source queue and a copy is put in another queue.

```
COPY FROM (SOURCE_QUEUE) TO (DESTINATION_QUEUE)
      CURSOR (1-999999999)
      COUNT (1-999999999)
      BATCH (1-999999999)
```

CURSOR – option to position the cursor to a specific message number and start the operation from there; default is 1.

COUNT – option to define the number of messages to copy; default is 1.

BATCH – option to define the number of messages copied under one sync point; default is 1.

DELETE – This command deletes a number of messages or a batch of messages from a queue.

```
DELETE FROM (SOURCE_QUEUE)
        CURSOR (1-999999999)
        COUNT (1-999999999)
        BATCH (1-999999999)
```

DISPLAY – This command displays messages on the console.

```
DISPLAY FROM (SOURCE_QUEUE)
        CURSOR (1-999999999)
        COUNT (1-999999999)
        DATAOFFSET (1-104857600)
        DATASIZE (1-104857600)
        DSPOPT (HDR | HEX | TXT | BODYHEX |
        BODYTXT) BREAK (1-1000)
```

CURSOR – specifies which message to start at (message 1, 2, etc.)

COUNT – specifies the number of messages to display

DATAOFFSET – option to specify the starting character offset within the message text to display; default is 1.

DATASIZE – option to specify the number of characters to display; default is all.

DSPOPT – option to select the display format:

- HDR – headers only
- HEX – hexadecimal
- TXT – text
- BODYHEX – only the body in hexadecimal
- BODYTXT – only the body in text

BREAK – specifies the number of messages to display at one time. For example, BREAK(2) displays two messages. Press **Enter** to display the next two messages and so on. When all messages are displayed, **Command completed RC (0)** is displayed.

HELP – Enter **Help** to see all the command names or **Help <command>** (for example, **Help Copy**) to see the basic usage for each command.

LOAD – This command copies messages from a text file to a queue. Each message in the file must be in XML format or one line of text terminated by CRNL (carriage return, line feed) or NL (line feed). No MQ headers must be in the file. The file contents are preserved; that is the messages remain in the file and a copy is put on the queue.

```
LOAD TO (DESTINATION_QUEUE) FILE (TEXT_FILE)
      IMPORT (ONE | MANY)
      DLMTR (XML | NL | CRNL)
```

IMPORT – option that defines how many messages to load from the file

- ONE – load the first message
- MANY – load all the messages

DLMTR – option to specify the message delimiter used in the file.

- XML – message is in XML format
- NL – single line of text terminated by line feed
- CRNL – single line of text terminated by carriage return, line feed

MOVE – This command moves a number of messages or a batch of messages from one queue to another and deletes the message from the source queue.

```
MOVE FROM (SOURCE_QUEUE) TO (DESTINATION_QUEUE)
      CURSOR (1-999999999)
      COUNT (1-999999999)
      BATCH (1-999999999)
```

MQRC – This command translates the specified MQ reason code and provides a short description.

MQRC <reason code>

```

Command Prompt - nsqmmfcl -mMQM -nBEBOP -qTEST1
\nQM\BEBOP\TEST1:\> copy from(test1) to(test2)
Date: Tue Jun 16 16:18:40 2009
NSQ3101I Command submitted
Date: Tue Jun 16 16:18:41 2009
NSQ3190I Command completed, RC(2033)
\nQM\BEBOP\TEST1:\> mqrc 2033
DESCRIPTION FOR RC(2033):
MQRC_NO_MSG_AVAILABLE:
An MQGET call was issued, but there is no message on the queue
satisfying the selection criteria specified in MQMD (the MsgId
and CorrelId fields), and in MQGMO (the Options and MatchOptions
fields). Either the MQGMO_WAIT option was not specified, or the time
interval specified by the WaitInterval field in MQGMO has expired.
This reason is also returned for an MQGET call for browse, when the
end of the queue has been reached.
This reason code can also be returned by the mqGetBag and mqExecute
calls. mqGetBag is similar to MQGET. For the mqExecute call, the
completion code can be either MQCC_WARNING or MQCC_FAILED:
- If the completion code is MQCC_WARNING, some response messages
were received during the specified wait interval, but not all.
The response bag contains system-generated nested bags for the
messages that were received.
- If the completion code is MQCC_FAILED, no response messages were
received during the specified wait interval.
Corrective action: If this is an expected condition, no corrective
action is required.
If this is an unexpected condition, check that:
- The message was put on the queue successfully.
- The unit of work (if any) used for the MQPUT or MQPUT1 call
was committed successfully.
- The options controlling the selection criteria are specified
correctly. All of the following can affect the eligibility of
a message for return on the MQGET call:
MQGMO_LOGICAL_ORDER
MQGMO_ALL_MSGS_AVAILABLE
MQGMO_ALL_SEGMENTS_AVAILABLE
MQGMO_COMPLETE_MSG
MQMO_MATCH_MSG_ID
MQMO_MATCH_CORREL_ID
MQMO_MATCH_GROUP_ID
MQMO_MATCH_MSG_SEQ_NUMBER
MQMO_MATCH_OFFSET
Value of MsgId field in MQMD
Value of CorrelId field in MQMD
Consider waiting longer for the message.

```

Figure E-3. MQRC Command

PUT – This command puts messages to a queue that are either typed in from the command line (text or hex data) or read from a file. The entire file is considered as the data for one message. It is assumed that the message descriptor (MQMD) is not present in the file.

```

PUT TO (DESTINATION_QUEUE) FILE (FILE_NAME)
BODYTYPE (HEX | TXT | FILE)
COUNT (1-999999999)
PMO (SYNC | NOSYNC | PMO INTEGER)

```

BODYTYPE – option to specify the message body type.

- HEX – option to type hex message at the prompt
- TXT – option to type text message at the prompt
- FILE – option to read message text from a file. You must use BODYTYPE (FILE) if you specify a file name.

COUNT – specifies the number of copies of the message to put to the queue

PMO – put message options

- SYNC – put all the messages under a syncpoint; that is, using MQCMIT
- NOSYNC – put the messages without a syncpoint
- PMO INTEGER – integer representing the sum of a subset of the decimal MQPMO option values in Table E-13A.

Option	Decimal Value	Hexadecimal Value
MQPMO_NONE	0	X'00000000'
MQPMO_SYNCPOINT	2	X'00000002'
MQPMO_NO_SYNCPOINT	4	X'00000004'
MQPMO_DEFAULT_CONTEXT	32	X'00000020'
MQPMO_NEW_MSG_ID	64	X'00000040'
MQPMO_NEW_CORREL_ID	128	X'00000080'
MQPMO_PASS_IDENTITY_CONTEXT	256	X'00000100'
MQPMO_PASS_ALL_CONTEXT	512	X'00000200'
MQPMO_SET_IDENTITY_CONTEXT	1024	X'00000400'
MQPMO_SET_ALL_CONTEXT	2048	X'00000800'
MQPMO_ALTERNATE_USER_AUTHORITY	4096	X'00001000'
MQPMO_FAIL_IF QUIESCING	8192	X'00002000'
MQPMO_NO_CONTEXT	16384	X'00004000'
MQPMO_LOGICAL_ORDER	32768	X'00008000'
MQPMO_RESOLVE_LOCAL_Q	262144	X'00040000'

REQUEUE – This command takes the specified file containing messages with MQ headers and requeues the previously unloaded messages. This is the reverse operation of the UNLOAD command.

```
REQUEUE TO (DESTINATION_QUEUE) FILE (FILE_NAME)
```

REROUTE – This command reroutes messages or a batch of messages from one queue to another. It is similar to the MOVE command, but has more options.

```
REROUTE FROM (SOURCE_QUEUE) TO (DESTINATION_QUEUE)
STRIPHDR (DLH | XQH | ALL)

DLHRC (1-999999999)

FAILMTD (STOP | BYPASS | EXCLUDE)

CURSOR (1-999999999)

COUNT (1-999999999)

BATCH (1-999999999)
```

STRIPHDR – option to strip message headers (DLH, XQH, or all).

DLHRC – option to only reroute messages with this DLH reason code.

FAILMTD – option defining how to proceed if the put of a message fails.

- STOP – stop the program
- BYPASS – ignore the failed put and continue writing additional messages
- EXCLUDE – ignore the failed put and stop writing additional messages

SAVE – This command copies messages from a queue to a file. The messages stay on the queue and a copy is written to the file.

```
SAVE FROM(SOURCE_QUEUE) FILE(FILE_NAME)
      CURSOR(1-999999999)
      COUNT(1-999999999)
      STRIPHDR(MD | ALL)
      DLMTR(NL | CRNL)
```

UNLOAD – This command moves messages from a queue to the specified file and deletes the message from the source queue.

```
UNLOAD FROM(SOURCE_QUEUE) FILE(FILE_NAME)
        FLOPT(NEW | APPEND)
        CURSOR(1-999999999)
        COUNT(1-999999999)
        STRIPHDR(MD | ALL)
        DLMTR(NL | CRNL)
```

FLOPT – option to specify how to write the messages to the file.

- NEW – write the messages to a new file. This is the default.
- APPEND – append the messages to the end of an existing file.

STRIPHDR – option to remove headers from the message.

- MD – remove only the message descriptor
- ALL – remove all message headers. This is the default.

DLMTR – option to specify a delimiter to separate the unloaded messages. The MQ message headers are preserved. The default is no delimiter.

- NL – line feed
- CRNL – carriage return, line feed

12.3.4 Possible Problems

The reason code 20013 is displayed as a reply to a message operation.

2013 is EXRC_MSG_Q_SERVER_DOWN:

The M6-WMQ Agent detected that M6-WMQ Message Management Server is not reading from NASTEL.MMF.ADMIN.COMMAND.QUEUE therefore could not carry out its operation(s).

Corrective Action: Start\restart M6-WMQ Message Management Server and make sure NASTEL.MMF.ADMIN.COMMAND.QUEUE is defined and both Gets and Puts are enabled.

Destination queue depth does not show the number of messages copied or moved.

If you use the **nsqmmfcl dir** command to see the destination queue depth after a message command, such as copy or move, the queue depth seen may not yet reflect the actual depth, because the workgroup server, from which the depth count is obtained, has not yet refreshed its local database.

Corrective Action:

- Repeat the dir command one or two times or
- Wait for the refresh interval (typically 30 seconds) and repeat the dir command or
- Use the M6-WMQ Explorer to force a refresh.

E.13 Queue Monitor Utility

The queue monitor utility is an application named `nsqmon (.exe)`.

E.13.1 Platforms

The queue monitor utility is available on the following platforms:

- Windows 2000 or later
- UNIX

E.13.2 Command-Line Arguments

Table E-14 lists all command-line arguments applicable to the queue monitor utility.

Table E-14. Queue Monitor Utility Command Line Arguments	
Arguments	Description
Required Arguments:	
<code>-m<GroupName></code>	Specifies the name of the workgroup to connect to.
<code>-n<NodeName></code>	Monitor queues from named WMQ Node.
<code>-q<QmgrName></code>	Monitor queues from named queue manager.
Optional Arguments:	
<code>-o<QueueName></code>	Monitor specific queue. Default: all queues
<code>-U<User></code>	Username for authenticating with workgroup server. Default: current user
<code>-P<Password></code>	Password for authenticating with workgroup server. Default: current user
<code>-R<Realm></code>	Security realm for authenticating with workgroup server. Default: security realm for current user.
<code>-a</code>	Don't authenticate with workgroup server. Connect to workgroup server as a GUEST user.

E.14 Message Get/Put Utility

The message get/put utility is an application named `nsqmsgf (.exe)`.

E.14.1 Platforms

The message get/put utility is available on the following platforms:

- Windows 2000 or later
- UNIX
- OS/400
- z/OS

E.14.2 Command-Line Arguments

Table E-15 lists all command-line arguments applicable to the message get/put utility.

Table E-15. Message Get/Put Utility Command Line Arguments	
Arguments	Description
Required Arguments:	
<code>-m<QmgrName></code>	Queue manager to connect to
<code>-q<QueueName></code>	Queue to get/put messages from/to
Optional Arguments:	
<code>-g<0 1></code>	Get messages from queue 0 = browse queue, leaving messages on queue 1 = read and remove messages from queue This is the default if <code>-p</code> is not specified.
<code>-p</code>	Put messages onto queue.
<code>-f<FileName></code>	Name of file to read messages from (for put) or to write messages to (for get). Default: <code><QueueName>.msg</code>
<code>-d<1,2></code>	Indicates character(s) used to separate messages 1 = Messages separated by newline (“\n”) 2 = Messages separated by double at sign (“@@”) Default: 1
<code>-c</code>	Enable conversion of application data to conform to CCSID and Encoding. Default: Disable conversion
Optional Arguments for getting messages:	
<code>-w<WaitInterval></code>	Interval, in seconds, to wait for message. Default: 1
<code>-s<1,2></code>	Strip WMQ headers from messages 1 = Strip Message Descriptor (MD) header 2 = Strip all headers Default: Don't strip headers
<code>-b<MessageCursor></code>	Cursor of message to start reading from. Range: 1 – 999999999 Default: 1
<code>-n<MessageCount></code>	Number of messages to get. Range: 1 – 999999999 Default: all messages
<code>-r</code>	Generate replies to messages. Default: don't generate replies

Table E-15. Message Get/Put Utility Command Line Arguments

Arguments	Description
Optional Arguments for putting messages:	
-i<1,2,3>	Indicates how to interpret contents of <FileName> 1 = Treat entire file as a single message 2 = Treat file as containing many messages (delimited as defined by -d option) 3 = Treat file as an XML file where each XML structure defines a separate message
-h	Generate default Message Descriptor (MD) for messages
-o	Generate message with zero-length application data
-U<UserName>	User ID to be used in queue manager authentication
-P<Password>	Password to be used in queue manager authentication

E.14.3 Options to Get Messages from Queue to File

```
-m<Queue Manager name>
-q<Queue name>
[-g<0,1> (0 - BROWSE, 1 - GET)
[-w<Wait interval in sec.>]
[-f<File name>]
[-s<1,2>] (Strip WMQ headers, 1 - MD, 2 - ALL)
[-d<1,2>] (Message Separator, 1 - NL, 2 - @@)
[-b<Message Cursor> (1 - 999999999)
[-n<Message Count> (1 - 999999999)
[-c] (Enable conversion)
[-r] (Generate reply)
```

Example 1:

1. Transfer messages from queue to a file
 - a. First put two sample messages on a queue:

```
>amqsput AAA.LQ QMA71
Sample AMQSPUT0 start
target queue is AAA.LQ
message 1
message 2
Sample AMQSPUT0 end
```
 - b. Now get the messages from the queue to a file in browse mode (the messages remain on the queue):

```
nsqmsgf -mQMA71 -qAAA.LQ -g0 -fAAA.LQ.msgs -d2
```

Command Options:

```
Queue Manager : QMA71
Queue Name : AAA.LQ
File Name : AAA.LQ.msgs
Wait Interval : 1 seconds
```

```

MD Conversion : Disabled
Message Separator : @@
Strip MQ Headers : None
Message Cursor : 1
Message Count : -1
Delete Messages : NO
Queue AAA.LQ:
    Depth : (2\100000)
    GET : ALLOWED
Getting message 1, size of 9 bytes, format 'MQSTR ' ...
Getting message 2, size of 9 bytes, format 'MQSTR ' ...
No more messages available.
Current queue depth is 2

```

- c. Show the message file:

```

>dir AAA*

Volume in drive C is HP
Volume Serial Number is 0C14-E063
Directory of C:\temp
05/17/2016 11:01 AM 750 AAA.LQ.msgs
1 File(s) 750 bytes
0 Dir(s) 32,397,086,720 bytes free

```

- d. Verify the messages are still on queue:

```

>echo dis qlocal(AAA.LQ) curdepth |runmqsc QMA71
5724-H72 (C) Copyright IBM Corp. 1994, 2011. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QMA71.
1 : dis qlocal(AAA.LQ) curdepth
AMQ8409: Display Queue details.
QUEUE(AAA.LQ) TYPE(QLOCAL)
CURDEPTH(2)
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.

```

E.14.4 Options to Put Messages from File to Queue

```

-m<Queue Manager name>
-q<Queue name>
[-p<PUT>]
[-f<File name>]
[-i<1, 2, 3>]

```

```

i1 - Import file as 1 message;
i2 - Import text file as many messages;
i3 - Import XML file as many messages.
[-h<Generate default Message Descriptor>]
[-d<1,2>] (Message Separator, 1 - NL, 2 - @@)
[-c] (Enable conversion)
[-o] (Generate message with zero-length application data)
[-U<UserName>] - User ID to be used in queue manager authentication
[-P<Password>] - Password to be used in queue manager authentication

```

Example:

1. Restore the messages from example 1 to another queue BBB.LQ:

First check queue BBB.LQ:

```

>amqsbcg BBB.LQ QMA71
AMQSBCG0 - starts here
*****
MQOPEN - 'BBB.LQ'
No more messages
MQCLOSE
MQDISC

```

2. Restore the messages:

```
> nsqmsgf -mQMA71 -qBBB.LQ -p -fAAA.LQ.msgs -d2
```

Putting messages to the queue:

Command Options:

```

Queue Manager : QMA71
Queue Name : BBB.LQ
File Name : AAA.LQ.msgs
MD Conversion : Disabled
Message Separator : @@
Generate MQ Headers : No

```

Queue BBB.LQ:

```

Depth : (0\100000)
PUT : ALLOWED

```

Write messages to queue with the actual application data length

Number of bytes to analyze is file size 750

Found MQ Message 0 at position 0

Found MQ Message 1 at position 375

Putting message 1, size of 9 bytes...

Putting last message 2, size of 9 bytes...

Queue depth is 2

3. Sample dump of the exported message file.

This is what the exported message file AAA.LQ.msg looks like. Note the '@@' separator after each message text and before the next 'MD'.

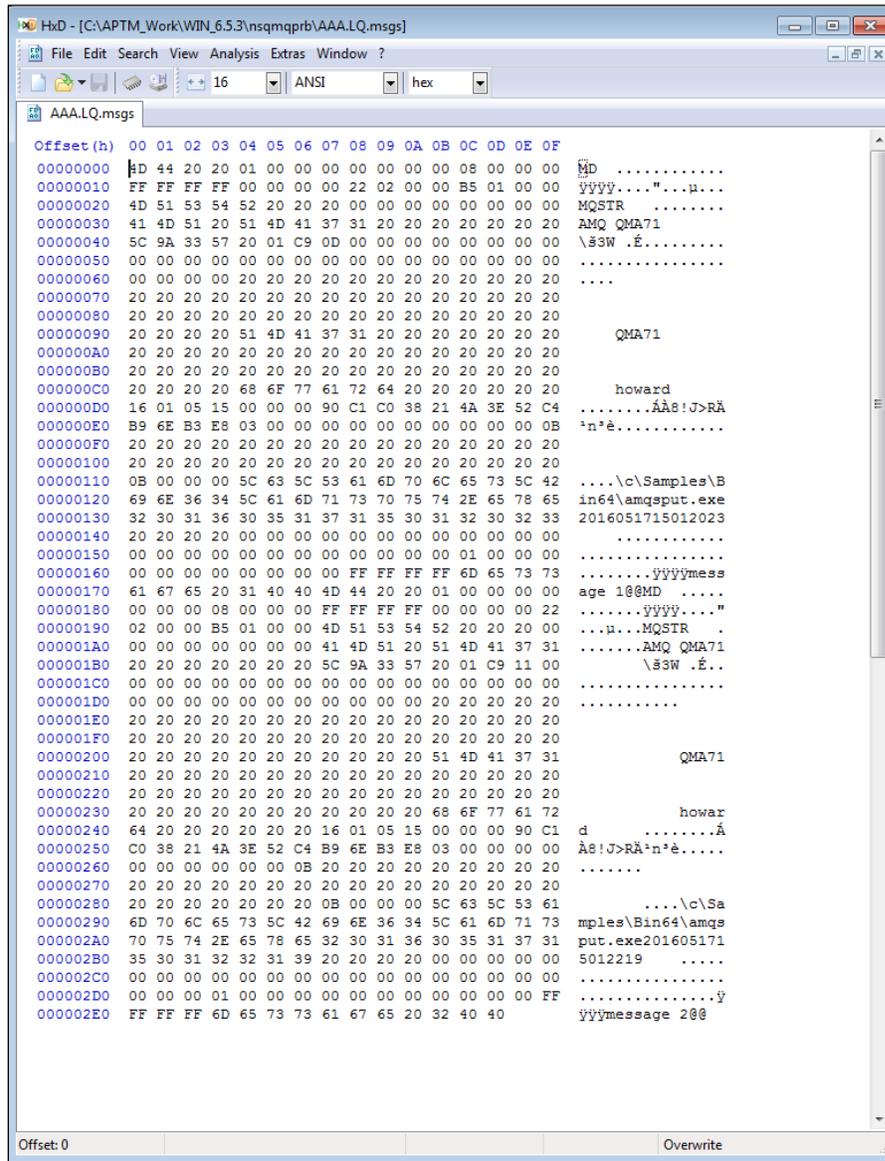


Figure E-4. Sample Dump of Exported Message File

E.15 SQL Database Creation Utility

The SQL database creation utility is an application named `nsqsqlmk (.exe)`.

E.15.1 Platforms

The SQL database creation utility is available on the following platforms:

- Windows 2000 or later
- UNIX

E.15.2 Command-Line Arguments

Table E-16 lists all command-line arguments applicable to the SQL database creation utility.

Table E-16. SQL Database Creation Utility Command-Line Arguments	
Arguments	Description
Optional Arguments (prompted for if not silent installation):	
<code>-D<DataSourceName></code>	Data Source Name identifying ODBC connection to M6-WMQ SQL database server.
<code>-U<DbUserName></code>	Database user name to use when establishing database connection.
<code>-P<DbPasswd></code>	Password to use when establishing database connection.
Optional Arguments:	
<code>-b<Message Browse Limit></code>	Maximum amount of MQ Messages that can be browsed on one request. This Workgroup Server variable will override the client settings. Default: 10,000.
<code>-m<WorkgroupName></code>	Name of workgroup to create. Default: MQM
<code>-r</code>	Include WMQ Authority records on Discovery. Default: 0.
<code>-s<PortNumber></code>	Port number that workgroup server for this workgroup will bind to. Default: 4010
<code>-l<EventLogSize></code>	Specifies the event log size to be allocated for the group. The size is measured in the number of events per event category. A log size of 100 means that there will be 100 event entries allocated for each event category. Default: 1,000
<code>-i</code>	Enable silent installation mode. Loads default M6-WMQ SQL database creation script with no user intervention required. Default: disabled
<code>-p</code>	Silently load M6-WMQ Permits SQL database creation script. Requires <code>-i</code> option (<code>-p</code> is ignored if <code>-i</code> is not specified). Default: disabled
<code>-d<Delimiter></code>	SQL statement delimiter (default: %)
<code>-c<CommentPrefix></code>	Comment line prefix (default: --)
<code>-A<AcctLogSize></code>	Specifies the accounting message log size to be allocated for the group. The size is measured in the number of accounting messages per accounting message category. A log size of 100 means that there will be 100 accounting entries allocated for each accounting message category. Default: 5,000
<code>-S<StatLogSize></code>	Specifies the statistics message log size to be allocated for the group. The size is measured in the number of statistics messages per statistics message category. A log size of 100 means that there will be 100 statistics entries allocated for each statistics message category. Default is: 5,000
<code>-L<M6-WMQAgentPort></code>	Port number that M6-WMQ Agents are listening on (default: 5010).



NOTE:

On Windows, SQL Database Creation Utility expects the M6-WMQ environment variable to be set to path name of M6-WMQ product installation directory (by default, `c:\nastel\apwmq`).

E.16 Remote Shutdown Utility

The remote shutdown utility is an application named `nsqstop (.exe)`.

E.16.1 Platforms

The remote shutdown utility is available on the following platforms:

- Windows 2000 or later
- UNIX
- OS/400
- z/OS

E.16.2 Command-Line Arguments

Table E-17 lists all command-line arguments applicable to the remote shutdown utility for the remote workgroup server.

Table E-17. Remote Workgroup Server Shutdown Command-Line Arguments	
Arguments	Description
Required Arguments (if not silent installation):	
<code>-m<GroupName></code>	Name of workgroup server to shutdown.
Optional Arguments:	
<code>-n<NodeName></code>	Node where WMQ Agent should be shut down. Default: only workgroup server is shut down.
<code>-g<GracePeriod></code>	Interval, in seconds, to wait for the workgroup server to shut down.
<code>-U<User></code>	Username for authenticating with workgroup server. Default: current user
<code>-P<Password></code>	Password for authenticating with workgroup server. Default: current user
<code>-R<Realm></code>	Security realm for authenticating with workgroup server. Default: security realm for current user
<code>-a</code>	Don't authenticate with workgroup server. Connect to workgroup server as a GUEST user.

Table E-18 lists all command-line arguments applicable to the remote shutdown utility for the remote WMQ Agent.

TableE-18. Remote WMQ Agent Shutdown Command-Line Arguments	
Arguments	Description
Required Arguments (if not silent installation):	
-n<NodeName>	Node where WMQ Agent should be shut down.
Optional Arguments:	
-i<IPAddress>	IP Address of host WMQ Agent is running on. Default: 127.0.0.1
-s<IPPort>	TCP/IP Port that the WMQ Agent is listening on. Default: 5000
CPYENVVAR (*YES)	Copy the environment variables, of which there is only APWMQ_HOME. Required on AS400/iSeries, where full command is: CALL PGM(nsqstop) PARM('nnode_name' ['-iip_adrs'] ['-sport']) CPYENVVAR(*YES)

E.17 Trigger Message Utility

The trigger message utility is an application named `nsqtrig(.exe)`.

E.17.1 Platforms

The trigger message utility is available on the following platforms:

- Windows 2000 or later
- UNIX

E.17.2 Command-Line Arguments

Table E-19 lists all command-line arguments applicable to the trigger message utility.

Table E-19. Trigger Message Utility Command-Line Arguments	
Arguments	Description
Required Arguments (if not silent installation):	
-m<GroupName>	Specifies the name of the workgroup to connect to.
-n<NodeName>	Name of WMQ Node to use.
-q<QmgrName>	Name of queue manager to use.
-l<LocalQueueName>	Name of local queue on <QmgrName> to put trigger message on.
Optional Arguments:	
-U<User>	Username for authenticating with workgroup server. Default: current user.
-P<Password>	Password for authenticating with workgroup server. Default: current user.
-R<Realm>	Security realm for authenticating with workgroup server. Default: security realm for current user
-a	Don't authenticate with workgroup server. Connect to workgroup server as a GUEST user.

E.18 WMQ Event Adapter

The WMQ event adapter is an application named `mgevent (.exe)`.

E.18.1 Platforms

The WMQ event adapter is available on the following platforms:

- Windows 2000 or later

E.18.2 Command-Line Arguments

Table E-20 lists all command-line arguments applicable to the WMQ event adapter.

Table E-20. WMQ Event Adapter Command-Line Arguments	
Arguments	Description
Optional Arguments:	
<code>-m<GroupName></code>	Specifies the name of the workgroup to connect to.
<code>-a</code>	Enables auditing of WMQ object change events. Default: disabled
<code>-r<Reconnect></code>	Interval, in sec, between attempts to connect to workgroup server. Default: 60
<code>-U<User></code>	Username for authenticating with workgroup server. Default: current user
<code>-P<Password></code>	Password for authenticating with workgroup server. Default: current user
<code>-R<Realm></code>	Security realm for authenticating with workgroup server. Default: security realm for current user
<code>-A</code>	Don't authenticate with workgroup server. Connect to workgroup server as a GUEST user.
<code>-install</code>	Installs Event Adapter as a service.
<code>-remove</code>	Removes Event Adapter as a service.
<code>-console</code>	Runs Event Adapter as a console application.

E.19 NSQZAS

E.19.1 Platforms

The NSQZAS is available on the following platform:

- z/OS

E.19.2 Command-Line Arguments

Table E-21 lists all command-line arguments applicable to ZAS.

Table E-21. ZAS Command-Line Arguments	
Arguments	Description
-m	qm1,qm2... List of queue managers for collecting accounting and statistics messages.
-mc	Queue Manager used by the collector for writing accounting and statistics messages to forward to workgroup server.
-qa	Enable or disable queue accounting
-ma	Enable or disable MQI accounting
-ms	Enable or disable MQI statistics
-qs	Enable or disable queue statistics
-cs	Enable or disable channel statistics
-st	Statistics interval in seconds. Default: 1800
[-t]	Trace functions
[-d]	Dump SMF records
[-ds]	Dump output to stdout. Default: File SMFDUMP

Appendix F: M6-WMQ Objects

Below are two tables listing M6-WMQ and WMQ objects that are created by the following M6-WMQ components:

- Workgroup servers
- WMQ Agents
- MQ Event Publisher
- MMF Server
- z/OS PCF Command Server

Table F-1 lists when and by which M6-WMQ component an object is created, the object's type, platforms that utilize the object and a comment about which M6-WMQ components access the object.

Table F-2 lists WebSphere MQ objects used by M6-WMQ components.

Table F-1. M6-WMQ Objects and Descriptions				
M6-WMQ Object	Platform	Created By/When	Attribute	Comments
NASTEL.ADMIN.AUDIT.QUEUE	All	Created during MMF installation (mmf.tst MQSC script).	Permanent, local, persistent	For future use by MMF.
NASTEL.MMF.ADMIN.COMMAND.QUEUE	All	Created during MMF installation (mmf.tst MQSC script).	Permanent, local, non-persistent	MMF Server uses this queue to receive Message Explorer commands. Messages are received and put there by the WMQ Agent.
NASTEL.EVENT.DEFAULT_date_time_hash** For non-z/OS platforms, the date_time string yyyyymmddhhmmsshh is attached as a suffix for a dynamic queue. For z/OS, a hash number is attached.	All	Created by WMQ Agent before subscribing to the Publisher. Occurs when Workgroup server sends EXCMD_OPEN_EVENT command to WMQ Agent.	Temporary dynamic, local, non-persistent	Publisher puts WebSphere MQ instrumentation events requested during subscription and DLQ events on the specified temporary.
NASTEL.PUBSUB.EVENT.QUEUE	All	Created during Publisher installation (pubsub.tst MQSC script).	Permanent, local, non-persistent	Publisher uses this queue to: Get QMGR instrumentation event messages Get M6-WMQ command message (open/close subscription, shutdown Publisher) Put DLQ event message.
NASTEL.REPLY. ClientName_Date_Time_hash where ClientName is name of connecting client application. Example: NSQMGR for WGS or NSQCL for the nsqcl test pgm	All	Created by WMQ Agent when connects and sends command EXCMD_MQ_OPENqmgr_name to connect to the queue manager.	Temporary dynamic, local, non-persistent	WMQ Agent thread uses this queue as a ReplyTo queue for responses from the WebSphere MQ PCF or MQSC command server.
NASTEL.REPLY. NASTEL.PCF.MQhash	z/OS	Created by MMF Server when it starts up.	Temporary dynamic, local, non-persistent	MMF Server uses this as ReplyTo queue to obtain responses for commands it submits to WebSphere MQ command server.
NASTEL.REPLY. NASTEL.MSG. Redate_Time_hash	All	Created by MMF Server when it starts up.	Temporary dynamic, local, non-persistent	MMF Server uses this as ReplyTo queue to obtain responses for commands it submits to WebSphere MQ command server.
NASTEL.EVENT.QUEUE	All	Created during M6-WMQ installation (by NASTEL.tst MQSC script).	Permanent, local, non-persistent	WMQ Agent gets WMQ instrumentation event messages from queue when Publisher is not configured.

Table F-2. WebSphere MQ Objects Used by M6-WMQ Components

M6-WMQ Object	Platform	Created By/When	Attribute	Comments
SYSTEM.ADMIN.COMMAND.QUEUE	Non-z/OS	Created during QMGR creation	Permanent, local, non-	WMQ Agent puts PCF commands on this queue for the WebSphere MQ PCF Server.
	z/OS	Created during M6-WMQ installation	Permanent, local, non-persistent	z/OS PCF Command Server gets commands from this queue, converts them to MQSC commands and submits them to MQSC command server.
SYSTEM.COMMAND.INPUT	z/OS	Created during QMGR creation	Permanent, local, non-persistent	z/OS PCF Command Server puts converted PCF commands onto this queue. WebSphere MQ Command Server gets commands from this queue.
SYSTEM.ADMIN.CHANNEL.EVENT SYSTEM.ADMIN.PERFMON.EVENT SYSTEM.ADMIN.QMGR.EVENT SYSTEM.ADMIN.CONFIG.EVENT SYSTEM.ADMIN.COMMAND.EVENT	All	Created during QMGR creation as local queues. Modified during M6-WMQ and Publisher installation (MQSC scripts NASTEL.tst, pubsub.tst)	Permanent, local, non-persistent	M6-WMQ installation script NASTEL.tst converts from local queue to alias queue with target queue set to NASTEL.EVENT.QUEUE Publisher installation script pubsub.tst redefines the alias queue with target queue set to NASTEL.PUBSUB.EVENT.QUEUE
SYSTEM.DEAD.LETTER.QUEUE	All	Created during QMGR creation.	Temporary, dynamic,	Publisher monitors this queue and creates an event when appropriate.
SYSTEM.ADMIN.ACCOUNTING.QUEUE	Non-z/OS	Created during QMGR creation as local queues. Modified during M6-WMQ and Publisher installation (MQSC scripts NASTEL.tst, pubsub.tst)	Permanent, local, non-persistent	M6-WMQ installation script NASTEL.tst converts from local queue to alias queue with target queue set to NASTEL.EVENT.QUEUE. Publisher installation script pubsub.tst redefines the alias queue with target queue set to NASTEL.PUBSUB.EVENT.QUEUE
SYSTEM.ADMIN.STATISTICS.QUEUE	Non-z/OS	Created during QMGR creation as local queues. Modified during M6-WMQ and Publisher installation (MQSC scripts NASTEL.tst, pubsub.tst)	Permanent, local, non-persistent	M6-WMQ installation script NASTEL.tst converts from local queue to alias queue with target queue set to NASTEL.EVENT.QUEUE. Publisher installation script pubsub.tst redefines the alias queue with target queue set to NASTEL.PUBSUB.EVENT.QUEUE

Appendix G: Deleted

This Page Intentionally Left Blank

Appendix H: WMQ Agent Run-Time Messages

This appendix lists run-time messages returned by the WMQ Agent.

Table H-1. Run-Time Messages Returned by WMQ Agents (Sheet 1 of 3)		
Msg Id	WMQ Agent Message/Meaning	Corrective Action
IMI0000	(NodeName, GroupName, port_number): WMQ Agent version RC(process_id.) Specifies version of the WMQ Agent.	None
IMI0001	(NodeName, GroupName, port_number): Usage: NSQMQ [- console] - mGroupName -sport_number -t. Process specified was started with improper command line options.	Use correct command line arguments. Check value of environment variable APWMQ_GS_OPTIONS
IMI0002	(NodeName, GroupName, port_number): Initialization completed. WMQ Agent initialized its internal resources.	None
IMI0003	(NodeName, GroupName, port_number): Successfully terminated RC(0).WMQ Agent terminated due to a user request or termination signal.	None
IMI0004	(NodeName, GroupName, port_number): Received signal indication RC (signal).WMQ Agent received an OS signal whose id is signal. All OS signals cause WMQ Agent to terminate. Signals usually originate from a user.	If the process terminates without explicit user request, contact your customer support representative.
IMI0005	(NodeName, GroupName, port_number): Executing signal indication RC (signal).WMQ Agent received an OS signal whose id is signal, which is executing clean-up procedures. All OS signals cause WMQ Agent to terminate. Signals usually originate from a user.	If the process terminates without explicit user request, contact your customer support representative.
IMI0007	(NodeName, GroupName, port_number): Group registration completed ok! RC(0).WMQ Agent registered with a Workgroup server specified on a command line.	None
IMI0008	(NodeName, GroupName, port_number): Client has disconnected! RC(system error). WMQ Agent detected that a connected client disconnected from the WMQ Agent with the system reason code specified in the RC() field.	Consult your OS guide for system error code to determine cause of disconnection. When a client drops connection normally, return code should indicate connection was reset by remote site.
IMI0009	(NodeName, GroupName, port_number): WMQ Agent node bound successfully! RC(Port).WMQ Agent bound to TCP/IP port number Port. This port is used by the WMQ Agent to accept new clients. The workgroup server that communicates with this WMQ Agent must have the same port in its node definition table.	None
IMI0010	(NodeName, GroupName, port_number): WMQ Agent node bound successfully! RC(Port). WMQ Agent bound to UDP port number Port. This port is used by the WMQ Agent to accept new clients. The workgroup server that communicates with this WMQ Agent must have the same port in its node definition table.	None
IMI0011	(NodeName, GroupName, port_number): Terminating due to signal RC(signal). WMQ Agent received an OS signal indicating that it should terminate. Signals usually originate from a user.	None
IMI0012	(NodeName, GroupName, port_number): Successfully terminated (RC(0). WMQ Agent completed its shutdown procedures successfully. The WMQ Agent has stopped.	None

Table H-1. Run-Time Messages Returned by WMQ Agents (Sheet 2 of 3)

Msg Id	WMQ Agent Message/Meaning	Corrective Action
IMI0014	(NodeName, GroupName, port_number): Registration sent to WORKGROUP(GroupName) at IPAddress(Port). WMQ Agent sends a registration request to the specified workgroup server at the indicated IP Address and service port.	None.
IMI0015	(NodeName, GroupName, port_number): Registration confirmation received from WORKGROUP(GroupName) at IPAddress(Port). WMQ Agent received the registration confirmation from the specified workgroup server at the indicated IP Address and service port. Workgroup server is not managing this WMQ Agent.	None
IMI0016	(NodeName, GroupName, QmgrName): Connected to queue manager RC(MQRC_). WMQ Agent established a connection to the specified queue manager.	None
IMI0017	(NodeName, GroupName, QmgrName): Reconnected to queue manager RC(MQRC_). WMQ Agent reestablished a previously lost connection to the specified queue manager.	None
IMI0018	(NodeName, GroupName, QmgrName): Terminating due to command RC(Command). WMQ Agent received the specified command indicating that it should terminate.	None
IMI0019	(NodeName, GroupName, QmgrName): Terminating due to command M6-WMQ Agent is terminating because it has received a command instructing it to do so.	None
IME0001	(NodeName, GroupName, port_number): Failed to create TC/IP server: RC(system_error).WMQ Agent was unable to establish a TCP/IP listening port due to one of these system errors:1000 - timeout occurred1001 - invalid socket identifier (call customer support)1002 - invalid service name specified. The -sservice parameter is not valid because service is not defined in the SERVICES file.1003 - invalid host name. The local host name is not defined or unknown. Ensure the local host name is defined in HOSTS, DNS, etc. 1004 - connection reset by TCP/IP stack (call customer support). Other errors - consult the OS system guide.	Follow the steps outlined.
IME0002	(NodeName, GroupName, port_number): Failed to accept new client RC(system_error)WMQ Agent was unable to accept a new client connection because either no more threads can be allocated to service the client, or system_error occurred (consult the OS system guide for system_error).	None
IME0004	(NodeName, GroupName, port_number): Out of memory RC(0).WMQ Agent cannot allocate memory and cannot continue.	Close applications or allocate enough resources for the WMQ Agent.
IME0005	(NodeName, GroupName, port_number): Unable to open queue manager RC(MQRC_). WMQ Agent cannot open a queue manager because of a WebSphere MQ error.	To get a description for MQRC_ code in the RC() field, invoke nsqcode MQRC_.
IME0007	(NodeName, GroupName, port_number): Invalid PCF command received RC(MQRC_). WMQ Agent received an EPCF command that does not conform to all EPCF standards.	To get a description for MQRC_, invoke nsqcode MQRC_. Ensure that all client applications use strict EPCF format.
IME0008	(NodeName, GroupName, port_number): Unable to process PCF command RC(MQRC_). WMQ Agent received an EPCF command that cannot be processed due to the reason specified in the RC() field.	To get an MQRC_ description, invoke nsqcode MQRC_. Ensure that all client applications use a strict EPCF format.
IME0009	(NodeName, GroupName, port_number): Unable to read client commands RC(system_error). WMQ Agent was unable to read EPCF commands from a client due to a system error.	Consult the OS system guide for an error description.
IME0010	(NodeName, GroupName, port_number): Unable to install signal handlers RC(system_error). WMQ Agent was unable to register required signal handlers with the OS.	Consult the OS system guide for an error description

Table H-1. Run-Time Messages Returned by WMQ Agents (Sheet 3 of 3)

Msg Id	WMQ Agent Message/Meaning	Corrective Action
IME0011	(NodeName, GroupName, port_number): Unable to read from event queue RC(MQRC_). WMQ Agent failed to read a message from the queue manager's event queue.	To get an MQRC_ description, invoke nsqcode MQRC_.
IME0012	(NodeName, GroupName, port_number): Unable to register under group RC(MQRC_). WMQ Agent was unable to contact the workgroup server due to one of these errors: 1000 - timeout occurred 1001 - invalid socket identifier (call customer support) 1002 - invalid service name specified. The -sservice parameter is not valid because service is not defined in SERVICES file. 1003 - invalid host name. The local host name is not defined or unknown. Ensure the local host name is defined in HOSTS, DNS, etc. 1004 - connection reset by TCP/IP stack (call customer support). MQRC_ (2000+ codes) - use nsqcodes to retrieve MQRC_ values. Other errors - consult the OS system guide for system description. WMQ Agent will continue to work normally; however, Workgroup server may not be aware of WMQ Agent's existence until the next discovery period.	Follow the steps outlined.
IME0013	(NodeName, GroupName, port number): Failed to initialize: unable to continue RC(Reason). WMQ Agent was unable to continue due to a previous error.	Refer to the previous error reported by the WMQ Agent.
IME0014	(NodeName, GroupName, port number): Failed to create UDP server RC(system_error). M6-WMQ failed to create socket to receive requests on.	Consult the OS system guide for an error description
IME0015	(NodeName, GroupName, port number): Failed to obtain digital signature. M6-WMQ failed to create digital signature for encryption, usually due to lack of memory.	Contact your customer support representative.
IME0016	(NodeName, GroupName, port number): Invalid UDP/PCF command received from IPAddress(Port)! WMQ Agent received an invalid or corrupted command message structure.	Verify messages sent by all clients.
IME0017	(NodeName, GroupName, QmgrName): Unable to read from reply queue RC(MQRC_). WMQ Agent failed to read a message from the queue manager's reply queue.	To get an MQRC_ description, invoke nsqcode MQRC_.
IME0018	(NodeName, GroupName, port number): Serious late IO PID (WMQ Agent Process ID). WMQ Agent is unable to respond to requests in a timely fashion.	Contact your customer support representative.
IME0019	(NodeName, GroupName, port number): Registration failed M6-WMQ Agent failed to register with the Workgroup Server.	Contact Nastel Support

This Page Intentionally Left Blank

Appendix I: Workgroup Server Run-Time Messages

This appendix lists run-time messages returned by the Workgroup servers.

Table I-1. Run-Time Messages Returned by Workgroup Server		
Message Id	Workgroup Server Message/Meaning	Corrective Action
IGE0001	Unable to open M6-WMQ Group (Group) RC (MQRC_). Workgroup server was unable to open group database, which must be located in install\groups directory.	Start workgroup server with a correct group name, or make a desired group using the nsqsqlmk utility.
IGE0002	Could not locate M6-WMQ install directory. Workgroup server cannot locate the M6-WMQ installation directory. On UNIX the default, installation directory is opt/nastel/M6-WMQ. On other OS, the default installation directory is set through the M6-WMQ environment variable.	Refer to the Installation Guide to configure the workgroup server.
IGE0003	Group group does not exist. Workgroup server cannot find the group record in the group database specified on a command line (-mgroup).	Try re-indexing the database using the nsqfix utility, or remake the database using the nsqsqlmk utility.
IGE0004	Group group is in use by another application! Workgroup server cannot start since some other application is using the group database. There already may be a workgroup server servicing this group.	Stop all applications that are using the group, and restart the workgroup server.
	(NodeName, GroupName, port_number): Received signal indication RC (signal). WMQ Agent received an OS signal whose id is signal. All OS signals cause WMQ Agent to terminate. Signals usually originate from a user.	If the process terminates without explicit user request, contact your customer support representative.
IGE0005	Group database is corrupted. Workgroup server detected that its database has been corrupted. It will attempt to repair the database and continue. Warning: During database repair, data loss is possible. Lost data is usually rediscovered automatically when the workgroup server starts.	None
IGE0006	Still unable to open M6-WMQ Group (group) RC (MQRC_XXX). Workgroup server was still unable to open the group database even after the repair process.	The database must be rebuilt using the nsqsqlmk utility. All data will be lost during this operation. Use nsqcode to get the MQRC_ code.
IGE0007	Node (group.node) not found in MQDB RC (DBRC_). Workgroup server cannot locate a referenced node in its database due to one of these DBRC_ codes: D_NF=1 Record not found D_PRIOR=2 No prior record for this request D_EOF=3 End of file D_BOF=4 Beginning of file D_DUPL=5 Primary key already exists D_OM=6 Out of memory D_INDXC=7 Index corrupted D_IOERR=8 I/O error D_LOCK=9 Database locked	Stop the workgroup server and re-index the database using the nsqfix utility.
IGE0008	Unable to establish [TCP/IP UDP] server:Host = host Service = service, Reason = system_error. Workgroup server was unable to start because it cannot establish a listening TCP/IP/UDP socket. Either a system error occurred, or the port/service is in use by some other application. Refer to the OS system guide for a description of system_error.	Ensure that TCP/IP stack is available, then remake databases using a different port number using the command: nsqsqlmk -mgroup [-sport] Another application is using the port. Close that application and retry.
IGE0009	Unable to load Group (group). Workgroup server was unable to load the group properties.	Follow the applicable corrective action in one of the following error messages.

Table I-1. Run-Time Messages Returned by Workgroup Server

Message Id	Workgroup Server Message/Meaning	Corrective Action
IGE0010	Record for Workgroup <group> not found. Workgroup server cannot locate the group record in the group database.	Try re-indexing the database using the nsqfix utility, or remake the nsqsqlmk utility.
IGE0011	-MANAGER (<group>) -MQNODE (<node>) -APPLNM (<appl_name>) -STATUS (WRITE_FAILED) RC (<system_error>). Workgroup server was unable to send commands to a WMQ Agent due to a communication error specified by the RC () field.	Refer to the OS system guide for details on the system_error code. System Action: Workgroup server will drop the connection. The connection will be retried in the next discovery period.
IGE0012	-MANAGER (<group>) -MQNODE (<node>) -APPLNM (<appl_name>) -STATUS (READ_FAILED) RC (<system_error>) Workgroup server was unable to read responses from a WMQ Agent due to a communication error specified by the RC () field.	Refer to the OS system guide for a description of system_error. System Action: Workgroup server will drop the connection. The connection will be retried in the next discovery period.
IGE0013	-MANAGER (<group>) -MQNODE (<node>) -APPLNM (<appl_name>) -STATUS (TIMEOUT) RC(<system_error>). Workgroup server was unable to read responses from a WMQ Agent due to a timeout.	Increase the group Cmd Timeout interval using M6-WMQ Explorer or some other utility. WMQ Agent may be hanging. Refer to the OS system guide for a description of system_error. System Action: Workgroup server will drop the connection. The connection will be retried in the next discovery period.
IGE0014	Initialization failed, unable to proceed. workgroup server was unable to continue due to a previous error.	See previous error(s) reported by workgroup server. System Action: Workgroup server will terminate.
IGE0015	Workgroup “<group>”, Unable to initiate ‘Node Discovery’. Workgroup server was unable to complete node discovery due to one of the following: No nodes have been defined or registered under the group. This is a normal condition. All WMQ Agents are either inaccessible or not running. This does not indicate a problem with a workgroup server, but may mean a problem with connectivity or availability of WMQ Agents.	None. Ensure that all WMQ Agents are running and connectivity is available. System Action: Workgroup server stops the discovery process until the next discovery period (see the Discovery Timer property of the Workgroup server).
IGE0016	Workgroup “<group>”, Unable to SyncPoint database, RC (<MQRC_ DBRC_>). Workgroup server was unable to commit the group database (normally done after every discovery cycle). This process failed due to one of the following DBRC_ or MQRC_ error codes: D_NF=1record not found D_PRIOR=2no prior record for this request D_EOF=3end of file D_BOF=4Beginning of file D_DUPL=5primary key already exists D_OM=6out of memory D_INDXC=7index corrupted D_IOERR=8I/O error D_LOCK=9database lockedMQRC_ codes. Use the utility nsqcode to get the MQRC_ code description.	Ensure that the workgroup server has read/write permission for all group database files. Verify that there is enough disk storage. After correcting the problem, restart the workgroup server. System Action: Workgroup server will terminate.

Table I-1. Run-Time Messages Returned by Workgroup Server		
Message Id	Workgroup Server Message/Meaning	Corrective Action
IGE0017	Discovery message could not be sent RC (<SystemCode>). Workgroup server could not send a discovery message to WMQ Agents due to an error.	Look up the SystemCode for your platform, or call technical support.
IGE0018	Starting with the following system limits: MAX_GROUP_CLIENTS =MaxClients MAX_QMGR_THREADS =MaxQMgrThreads MAX_MANAGED_NODES =MaxManagedNodes MAX_TOTAL_CONNS =MaxTotalConns MAX_Q_MGRS_PER_NODE =MaxQMgrsPerNode MAX_OBJ_INSTANCES =MaxObjectInstances. Workgroup server reports its operating limits, which are different in the group properties.	None
IGE0019	Workgroup server group is ready! Workgroup server group is ready for operation and management of WebSphere MQ.	None
IGE0020	Could not allocate memory! Workgroup server is unable to claim memory and cannot continue.	Close some programs to free up resources.
IGE0021	Broadcasting discovery to WMQ Agents on <IPAddress>(<Port>). Workgroup server is trying to discover all WebSphere MQ Nodes that run WMQ Agents on a specified port. WMQ Agents will respond only if they run within a given subnet, listen on a given port, and belong to the broadcasting group.	None
IGE0022	Workgroup server "<group>" is terminating due to signal(sig)! Workgroup server received a termination signal from a user or OS.	None System Action: Workgroup server will terminate.
IGE0023	Client (host IP IP service proto) requested Workgroup server "<group>" to shutdown! A client from a specified location (host IP IP_service proto) requested workgroup server to terminate.	None System Action: Workgroup server will terminate based upon a user request.
IGE0024	Workgroup server "<group>" is terminating in <time> (sec). Workgroup server will terminate in the given number of seconds. This message is generated due to a user shutdown request (EXCMD_MANAGER_SHUTDOWN).	None System Action: Workgroup server will terminate no earlier than time seconds after the issue of this message
IGE0025	Workgroup server "<group>" terminated successfully! Workgroup server instance was terminated by user request, or by an OS signal, and is no longer an active process. Refer to previous messages for more details.	None
IGE0026	workgroup server "<workgroup>" is terminating due to signal WM_CLOSE!! Specified workgroup received a Windows WM_CLOSE signal and will terminate.	None System Action: Workgroup server will terminate as soon as possible.
IGE0027	Could not load license from "<lic_file>", <system error>, RC (Reason) Workgroup server could not load a license file from the [AutoPilot-WMQ]\config\groups directory due to one of the following: SysRc - an OS specific reason code (example: file not found, or a security problem).Reason - one of these licensing reason codes: LR_NO_ERROR 0no error in the license format LR_CHECKSUM_ERROR -1license file is corrupted LR_LICENSE_FILE_ERROR -2general failure (refer to SysRc)	Inspect the reason code and perform the appropriate action: Install a license file. Give the Workgroup server access to the license file. Restore the license file if it was corrupted. System Action: Workgroup server will terminate.
IGE0028 (Node-based licensing only.)	Workgroup "<workgroup>" is not licensed on HOST("<hostname>") Specified workgroup is not licensed on the local machine hostname. The license file installed on the local machine does not allow group to run locally.	Verify that the workgroup server license file "<workgroup>.lic" exists in [APWMQ_HOME]/groups/config and that this is the proper license file for this workgroup server.

Table I-1. Run-Time Messages Returned by Workgroup Server		
Message Id	Workgroup Server Message/Meaning	Corrective Action
		Contact your sales representative and obtain the license file for your group name. System Action: Workgroup server will terminate.
IGE0028 (CPU-based licensing only.)	Workgroup "<workgroup>" is not licensed on HOST("<hostname>") The workgroup server is not licensed to run on the specified host.	Verify that the workgroup server has been installed on the proper host and that the license file AutoPilotWMQ.lic contains the required host name. System Action: Workgroup server will terminate.
IGE0029 (Node-based licensing only.)	Host "<hostname>", resolving to IP address... Workgroup server is trying to determine the IP address for the local machine specified in the <host> parameter in order to verify licensing information.	None. System Action: Workgroup server will resolve the host name either through the local HOSTS file, or DNS, whichever is configured. This operation may take some time if DNS is down or is not reachable. If host name is unresolved (and license unverified), local machine will be shut down.
IGE0030 (Node-based licensing only.)	Host "<hostname>", resolved to IP address <IPAddress>. Workgroup server completed IP address resolution. If address is blank, the resolution process failed and licensing verification may not be completed.	None System Action: None
IGE0031 (Node-based licensing only.)	Host "<hostname>(<IPAddress>)", incorrect license "<lic_hostname>(<lic_IPAddress>)" Local host is not licensed to run any of the Workgroup servers.	Obtain a license file for the host. System Action: Workgroup server will terminate.
IGE0032 (CPU-based licensing only.)	Software license has expired on <date>. Workgroup server reports that the license has expired.	Obtain a license file from your sales representative. System Action: Workgroup server will terminate.
IGE0032 (Node-based licensing only.)	Host "<hostname>(<IPAddress>)", Software license has expired on <date>. Workgroup server reports that the license has expired.	Obtain a license file from your sales representative. System Action: Workgroup server will terminate.
IGE0033	Failure in SET_BUFFER_SIZE(<Hostname> <IPAddress>, <size> (Bytes))! Workgroup Server failed to allocate a message buffer of the indicated size.	Contact your customer support representative.
IGE0034	Unrecoverable error while processing <method>(Node#<node id>)! Workgroup Server encountered an internal processing error from which it cannot recover	Contact your customer support representative. System Action: Workgroup server will terminate.
IGE0035	Workgroup database "<GroupName>", backup completed, RC(<system error>). Workgroup Server completed backup of database for specified workgroup.	If RC is non-zero, database backup failed. In this case, look up the SystemCode for your platform, or call technical support.
IGE0036	Workgroup database "<GroupName>" is corrupted. Attempting to use backup... The database for the specified workgroup is corrupted and cannot be used. Workgroup Server is attempting to restore the latest backup.	None

Table I-1. Run-Time Messages Returned by Workgroup Server		
Message Id	Workgroup Server Message/Meaning	Corrective Action
IGE0037	Workgroup database "<GroupName>" has been restored. Workgroup Server successfully restored the latest database backup for the specified workgroup.	None
IGE0038	Unable to open Workgroup backup database "<GroupName>", RC(<system>). Workgroup server could not access database backup for specified group.	Look up the SystemCode for your platform, or call technical support.
IGE0039	Workgroup database "<GroupName>", failed to restore, (<system error>). Workgroup Server was unable to restore the database backup.	None System Action: Workgroup server will attempt to repair the database
IGE0040	Workgroup database "<GroupName>", creating backup... Workgroup Server is creating a backup of the database for the specified workgroup.	None
IGE0041	Could not open %s log, RC(<system error>). Workgroup Server could not open the indicated log type/file.	Look up the SystemCode for your platform, or call technical support.
IGE0042	Workgroup backup database is corrupted. Attempting to repair... The database backup for the specified workgroup is corrupted and cannot be used. Workgroup Server is attempting to repair the corrupted database	None
IGE0043	Unable to obtain digital signature! Workgroup Server failed to create digital signature for encryption, usually due to lack of memory.	Contact your customer support representative.
IGE0044	Workgroup database "<GroupName>" is not loaded. Workgroup server could not load record for specified workgroup database.	Verify that workgroup database has been created successfully and contains a definition for the specified workgroup.
IGE0045	Workgroup database "<GroupName>" could not connect RC(<system/db-server error>). Workgroup server could not connect to database for specified workgroup.	If using Nastel database, look up the SystemCode for your platform, or call technical support. If using SQL database, look up the system code for your database server type.
IGE0046	"Workgroup '<GroupName>' is not licensed! No license was found for this workgroup. System Action: Workgroup server will terminate.	Verify that the license file AutoPilotWMQ_<GroupName>.lic exists in [APWMQ_HOME]\config\groups and that the file is readable, contains the proper format, and is for the specified workgroup.
IGE0047 (CPU-based licensing only.)	"Unable to load CPU-based license: <reason> Workgroup server failed to load the CPU-based licensing information. System Action: Workgroup server will terminate.	Verify that the license file AutoPilotWMQ_<GroupName>.lic exists in [APWMQ_HOME]\config\groups, where <GroupName> is the workgroup being processed by Workgroup server, and that the file is readable and contains the proper format.
IGE0048	Workgroup '<GroupName>' is not licensed to use Kerberos Authentication! System Action: Workgroup server will terminate.	Verify that the license file AutoPilotWMQ_<GroupName>.lic exists in [APWMQ_HOME]\config\groups and that the file is readable, contains the proper format, and has an entry for feature 'Kerberos'.

Table I-1. Run-Time Messages Returned by Workgroup Server

Message Id	Workgroup Server Message/Meaning	Corrective Action
IGE0049	Workgroup '<GroupName>' is not licensed to use SQL Database feature! Workgroup server failed to validate the use of SQL database support. System Action: Workgroup server will terminate.	Verify that the license file AutoPilotWMQ_<GroupName>.lic exists in [APWMQ_HOME]\config\groups and that the file is readable, contains the proper format, and has an entry for feature 'SQLDB'.
IGE0050	Creating link to Security Server <IPAddress-Port>. Workgroup Server is establishing connection to AutoPilot M6 Security Server.	None
IGE0051	Workgroup Server "<GroupName>" failed to send challenge command to Security Server, RC(<system error>)! Workgroup Server failed to initiate user authentication with AutoPilot M6 Security Server.	Look up the SystemCode for your platform, or call technical support.
IGE0052	Workgroup Server "<GroupName>" failed to receive challenge response from Security Server, RC(<system error>)! Workgroup Server failed to read user authentication response from AutoPilot M6 Security Server.	Look up the SystemCode for your platform, or call technical support.
IGE0053	Workgroup Server "<GroupName>" failed to send login command to Security Server, RC(<system error>)! Workgroup Server failed to send user authentication login request to AutoPilot M6 Security Server.	Look up the SystemCode for your platform, or call technical support.
IGE0054	Workgroup Server "<GroupName>" failed to receive challenge response from Security Server, RC(<system error>)! Workgroup Server failed to read user authentication login response from AutoPilot M6 Security Server.	Look up the SystemCode for your platform, or call technical support.
IGE0055	Workgroup Server "<GroupName>" failed to establish connection with Security Server, RC(<system error>)! Workgroup Server failed to establish a connection to AutoPilot M6 Security Server.	Look up the SystemCode for your platform, or call technical support.
IGE0056	Deleting link to Security Server <IPAddress-Port>. Workgroup Server is closing the connection to AutoPilot M6 Security Server.	None
IGE0061	SQL DB(DBType) - <DataSource>(USER:<DBUserName>) - connected. Workgroup Server successfully connected to the specified type of database using the indicated Data Source Name and Database User.	None
IGE0062	SQL DB(DB Type) - <Data Source>(USER:<DB User Name>) - connection failed. <DB-specific diagnostic message> Workgroup Server failed to connect to the specified type of database using the indicated Data Source Name and Database User.	Verify state of database server. See Database specific diagnostic message and consult documentation from database vendor for further information.
IGE0063	Command '<SQL statement>' - failed. <DB-specific diagnostic message> Workgroup Server encountered an error executing the specified SQL statement.	Verify state of database server. See Database specific diagnostic message and consult documentation from database vendor for further information.
IGE0064	SQL DB <DataSource> disconnected. Workgroup Server disconnected from the specified Data Source Name.	None

Appendix J: Deleted

This Page Intentionally Left Blank

Appendix K: WMQ Agent z/OS Error Codes and Messages

Table K-1. WMQ Agent z/OS Error Codes and Messages (Sheet 1 of 14)

Msgid	Message Text	Explanation	User Action
NSQ0100W	CreateMQManager: Unable to create event queue qmgr:queue RC(reason)	qmgr: Queue Manager queue: Event Queue reason: WebSphere MQ reason code.	Review WebSphere MQ reason code to determine the cause of the error and correct the problem. If necessary, contact Nastel Support.
NSQ0101W	CreateMQManager: Unable to create reply queue qmgr:queue RC(reason)	qmgr: Queue Manager queue: Reply Queue reason: WebSphere MQ reason code	Review WebSphere MQ reason code to determine the cause of the error and correct the problem. If necessary, contact Nastel Support.
NSQ0102W.	qmOpen: Open Reply Queue failed, ReplyQ qmgr:queue REASON (reason) UserID(user) AltID(altuser)	qmgr: Queue Manager queue: Reply Queue reason: WebSphere MQ reason code user: Original user altuser: Alternate user	Review WebSphere MQ reason code to determine the cause of the error and correct the problem. If necessary, contact Nastel Support.
NSQ0103E	qmOpen: Command Queue and/or Reply Queue not specified in qmgr	qmgr: Queue Manager	Use MQSC to specify the command and/or reply queues for the specified queue manager. If necessary, contact Nastel Support.
NSQ1104E	Unable to create reply queue on QMGR (qmgr)	The PCF translator was unable to create a reply queue for a command that requires a response. <i>qmgr</i> : Queue Manager	Check job log and sysout for associated error messages and correct any indicated problems. Review region size and increase, if necessary. If problem persists, contact Nastel Support.
NSQ0104W	Could not open \"ini\" RC(errno)	ini: ddname for ini file. errno: Error code after fopen() call.	Review errno reason code to determine the cause of the error and correct the problem. Also, inspect joblog for IECxxx error messages. Most likely cause will be missing ddname. If necessary, contact Nastel Support.

Table K-1. WMQ Agent z/OS Error Codes and Messages (Sheet 2 of 14)

Msgid	Message Text	Explanation	User Action
NSQ0105E	Assertion failed in file at line no: message	An internal error has been detected. file: Source file name line: line number message: Further information about error	If necessary, contact Nastel Support.
NSQ0106W	Timeout received, possible unreachable host	The attempt to contact the host timed out; the request will be retried.	If message is seen frequently or contact is not successfully made check TCP/IP routing, DNS server and firewall issues. If necessary, contact Nastel Support.
NSQ0107W	qmOpen: Clear Reply Queue failed, ReplyQ qmgr:queue REASON (reason) UserID (user) AltID (altuer)	qmgr: Queue Manager queue: Reply Queue reason: WebSphere MQ reason code user: Original user altuser: Alternate user	Review WebSphere MQ reason code for possible problems, for example security issues. If necessary, contact Nastel Support.
NSQ0108A	Unable to read filename file	The +u option was specified but filename could not be read. (All authorization requests will be failed, successful execution is unlikely.)	Insure that the specified file is available and can be read, then retry the function. If necessary, contact Nastel Support.
NSQ1001I	WMQ Agent started Vversion	version: Version of WMQ Agent in format vv.rr.mmmm, for example 04.05.0001.	None.
NSQ1003I	M6-WMQ PCF Command Server started Vversion	version: Version of PCF Command Server in format vv.rr.mmm, for example 04.05.0001.	None.
NSQ1101E	Unable to read ReplyQ qmgr:queue REASON (reason)	qmgr: Queue Manager queue: Reply Queue	Retry function; if problem persists, contact Nastel Support.
NSQ1102W	MQSC/PCF conversion failed for MQSC (mqsc) PCFID (pcfid) COUNT (count) COMPCODE (compcode) REASON (reason) QUAL (qualifier)	mqsc: MQSC text pcfid: PCF code count: Position of failing code. compcode: Completion code. qualifier: Further information.	If you are using the Nastel M6-WMQ SDK to send PCF messages, review for possible problems and correct your error. Otherwise, this is an internal error. In the latter case, contact Nastel Support.

Table K-1. WMQ Agent z/OS Error Codes and Messages (Sheet 3 of 14)

Msgid	Message Text	Explanation	User Action
NSQ1103E	qmgr failed to reply to QUEUE (queue) REASON (reason)	qmgr: Queue Manager queue: Queue name reason: WebSphere MQ reason code	Retry function; if problem persists, contact Nastel Support.
NSQ1104E	Unable to create reply queue on QMGR (qmgr)	The PCF translator was unable to create a reply queue for a command that requires a response. qmgr: Queue Manager	Check job log and sysout for associated error messages and correct any indicated problems. Review region size and increase if necessary. If problem persists, contact Nastel Support.
NSQ1108E	DoExcmd OpenEvent: qqOpen failed, EventQ qmgr:queue REASON (reason)	qmgr: Queue Manager queue: Event Queue reason: WebSphere MQ reason code	Retry function; if problem persists, contact Nastel Support.
NSQ1109W	Discovery cannot start, EventQ qmgr:queue not open for input	Because the specified queue could not be opened, discovery of WebSphere MQ objects will not be done. qmgr: Queue Manager queue: Event Queue	Review for possible problems and correct. If necessary, contact Nastel Support.
NSQ1110W	GetPcfEvent: MQINQ (value) failed RC(reason)	value: Object in error reason: WebSphere MQ reason code	Retry function; if problem persists, contact Nastel Support.
NSQ1111W	Failed to open input file file	file: ddname of file that failed to open.	Review for possible problems, for example missing ddname. If necessary, contact Nastel Support.
NSQ1111W	Failed to open input file file	file: ddname of file that failed to open.	Review for possible problems, for example missing ddname. If necessary, contact Nastel Support.
NSQ1112W	errno (errno) message	errno: C library error code message: Informative message text	Review for possible problems. Check for further messages in stdout. If necessary, contact Nastel Support.

Table K-1. WMQ Agent z/OS Error Codes and Messages (Sheet 4 of 14)

Msgid	Message Text	Explanation	User Action
NSQ1113W	Failed to write to input file file	file: ddname of failing file	Review for possible problems and any information in message NSQ1112W. If necessary, contact Nastel Support.
NSQ1114W	Failed to create output file file	file: ddname of failing file	Review for possible problems and any information in message NSQ1112W. If necessary, contact Nastel Support.
NSQ1115W	Failed to open output file file to read	file: ddname of failing file	Review for possible problems and any information in message NSQ1112W. If necessary, contact Nastel Support.
NSQ1116W	Response from client not legal, ignored	Malformed client response has been dropped.	Retry function; if problem persists, contact Nastel Support.
NSQ1117W	SRVTASK: RC(reason) from get_conn()	reason: MTF failure code	Review MTF reason code for possible problems. If necessary, contact Nastel Support.
NSQ1118W	QueueManager (qmgr): Command Server is stopped or not responding.	qmgr: Queue Manager	Attempt to restart the command server; inspect job log of queue manager for problems starting the command server. If necessary, contact Nastel Support.
NSQ1119W	QueueManager (qmgr): Command Server is not running.	qmgr: Queue Manager	Attempt to restart the command server; inspect job log of queue manager for problems starting the command server. If necessary, contact Nastel Support.
NSQ1120W	QueueManager (qmgr): is not running.	qmgr: Queue Manager	Attempt to restart the queue manager. Review z/OS console log for possible problems. Remove unused queue managers from MQSINI file. If necessary, contact Nastel Support.
NSQ1121W	QueueManager (qmgr): defined but does not exist.	qmgr: Queue Manager	Attempt to restart the queue manager. Review z/OS console log for possible problems. Remove unused queue managers from MQSINI file. If necessary, contact Nastel Support.
NSQ1122W	No queue managers have been found on your system.	The WMQ Agent did not find any valid Queue Managers to monitor.	Review for possible problems. If queue managers exist on this system, review previous error messages and stdout for further diagnostic information. If necessary, contact Nastel Support.

Table K-1. WMQ Agent z/OS Error Codes and Messages (Sheet 5 of 14)

Msgid	Message Text	Explanation	User Action
NSQ1123A	If queue manager(s) exist, refer to the Installation Guide to configure	A previous message (NSQ1122W) has been issued which indicates no queue managers exist; for M6-WMQ to function it must identify at least one valid queue manager.	See message NSQ1122W. The <i>M6 for WMQ Installation Guide</i> in the OS/390 chapter describes how to add queue manager entries to MQSINI.
NSQ1124T	Fatal error while initializing MTF, tinit RC (reason)	reason: MTF error code	If reason code is -7, try adding the proper EDCMTF dd statement. Contact Nastel Support.
NSQ1134E	Unable to connect to Queue Manager qmgr REASON (reason), will retry every interval seconds	qmgr: Queue Manager reason: WebSphere MQ reason code interval: Retry interval	Ensure queue manager is running. Review WebSphere MQ reason code. If problem persists, contact Nastel Support.
NSQ1136I	Successfully connected to Queue Manager qmgr	qmgr: Queue Manager	None.
NSQ1137W	QueueManager (qmgr): PASSED verification test.	qmgr: Queue Manager	None.
NSQ1138W	QueueManager (qmgr): FAILED verification test with RC (reason).	qmgr: Queue Manager reason: WebSphere MQ reason code	Review WebSphere MQ reason code for possible problems. If necessary, contact Nastel Support.
NSQ1139E	Queue manager qmgr not available, will retry every interval seconds	qmgr: Queue Manager interval: Retry interval	Ensure queue manager is running. Review WebSphere MQ reason code. If problem persists, contact Nastel Support.
NSQ1141E	Unable to open Queue qmgr:queue REASON (reason); will retry every interval seconds	qmgr: Queue Manager queue: Queue Name reason: WebSphere MQ Reason code interval: retry interval	Review WebSphere MQ reason code and correct any problems. Retry function; if problem persists, contact Nastel Support.
NSQ1142E	Queue manager qmgr not available, will retry every interval seconds	qmgr: Queue Manager interval: Retry interval	Ensure queue manager is running. Review WebSphere MQ reason code. If problem persists, contact Nastel Support.

Table K-1. WMQ Agent z/OS Error Codes and Messages (Sheet 6 of 14)			
Msgid	Message Text	Explanation	User Action
NSQ1143I	Successfully opened input PCF CmdQ qmgr:queue	qmgr: Queue Manager queue: PCF Command Queue Name	None.
NSQ1144E	Unable to read from queue qmgr:queue REASON (reason)	qmgr: Queue Manager queue: Queue Name reason: WebSphere MQ Reason code	Review WebSphere MQ reason code and correct any problems. Retry function; if problem persists, contact Nastel Support.
NSQ1145E	Queue manager qmgr not available; will retry every interval seconds	qmgr: Queue Manager interval: Retry interval	Ensure queue manager is running. Review WebSphere MQ reason code. If problem persists, contact Nastel Support.
NSQ1146E	Unable to convert PCFID(pcfid) USER(user) to MQSC, COMPCODE (compcode) REASON (reason) QUAL (qualifier)	pcfid: PCF code count: Position of failing code.comp code: Completion code. qualifier: Further information.	If you are using the Nastel M6-WMQ SDK to send PCF messages, review for possible problems and correct your error. Otherwise, this is an internal error. In the later case, contact Nastel Support.
NSQ1147E	Unable to write to queue qmgr:queue USER(user) REASON (reason)	qmgr: Queue Manager queue: Queue Name user: Original User reason: WebSphere MQ reason code	Review WebSphere MQ reason code and correct any problems. Retry function; if problem persists, contact Nastel Support.
NSQ1148E	Queue manager qmgr not available; will retry every interval seconds	qmgr : Queue Manager interval: Retry interval	Ensure queue manager is running. Review WebSphere MQ reason code. If problem persists, contact Nastel Support.
NSQ1149E	Unable to send reply to qmgr: queue USER(user) REASON (reason)	qmgr : Queue Manager queue: Queue Name user: Original User reason: WebSphere MQ reason code	Review WebSphere MQ reason code and correct any problems. Retry function; if problem persists, contact Nastel Support.
NSQ1150E	Queue manager qmgr not available; will retry every interval seconds	qmgr : Queue Manager interval: Retry interval	Ensure queue manager is running. Review WebSphere MQ reason code. If problem persists, contact Nastel Support.
NSQ1152E	Problem in PCF Command Server	A PCF command server function failed.	Review job log for previous error messages and correct any problems. Retry function; if problem persists, contact Nastel Support.

Table K-1. WMQ Agent z/OS Error Codes and Messages (Sheet 7 of 14)

Msgid	Message Text	Explanation	User Action
NSQ1153E	No resources available for Queue Manager qmgr	qmgr : Queue Manager	Use REGION JCL parameter to increase storage available and restart WMQ Agent. If problem persists, contact Nastel Support.
NSQ1154A	Queue Manager (qmgr): should be deleted or re-created.	qmgr : Queue Manager	Perform indicated action; if necessary, contact Nastel Support.
NSQ1155I	Successfully opened MQSC CmdQ qmgr:queue	qmgr : Queue Manager queue: MQSC Command Queue Name	None.
NSQ1156I	Successfully opened MQSC ReplyQ qmgr:queue	qmgr: Queue Manager queue: MQSC Reply Queue Name	None.
NSQ1157I	Program waiting for TCP/IP to become active.	The indicated program was unable to initiate TCP/IP communications. It will retry at intervals until TCP/IP becomes available. program: Program name. This problem can be caused by a missing or invalid SYSTCPD dd statement.	Ensure TCP/IP is running and retry function; consult with your TCP/IP administrator for SYSTCPD requirements. If problem persists, contact Nastel Support.
NSQ1158I	TCP/IP now active, initialization will proceed for program.	After retrying a failing TCP/IP communications connection, a successful connection was made. program: Program name	None.
NSQ1160E	Unable to open MQSC CmdQ and/or ReplyQ for Queue Manager qmgr REASON (reason); will retry every interval seconds	qmgr: Queue Manager reason: WebSphere MQ reason code interval: Retry interval	Review WebSphere MQ reason code and correct any problems. Retry function; if problem persists, contact Nastel Support.
NSQ1161E	CmdQ qmgr:queue	qmgr: Queue Manager queue: Command Queue Name	Refer to previous error message in job log.
NSQ1162E	ReplyQ qmgr:queue	qmgr: Queue Manager queue: Command Queue Name	Refer to previous error message in job log.
NSQ1200I	NSQTACON <vers.rel.patch> started	The NSQTACON task is running.	None.

Table K-1. WMQ Agent z/OS Error Codes and Messages (Sheet 8 of 14)

Msgid	Message Text	Explanation	User Action
NSQ1299I	stop issued, NSQTACON terminating	The NSQTACON task is stopped.	None.
NSQ1990T	Internal segmentation violation.	An internal error has occurred.	Contact Nastel Support.
NSQ1991I	Shutdown request of all WMQ Agent processes rcvd from ipaddr (service)	ipaddr: IP Address where request originated service: Service number of request	None.
NSQ1992T	Enqueue failed for WMQ Agent Queue Manager (qmgr)	Another WMQ Agent is already active for the indicated queue manager qmgr.	Ensure the two WMQ Agents are using different MQSINI files. Remove the qmgr entry from one of the MQSINI parameter files. Contact Nastel Support if this message is produced in error.
NSQ1997I	M6-WMQ PCF Command Server Vversion terminated with RC (reason)	version: PCF Command server version reason: Shutdown reason code, usually zero.	None.
NSQ1998I	Shutdown indicator was received from WMQ Agent	A request to shutdown the WMQ Agent has been received	None.
NSQ1999I	z/OS STOP issued, NSQMQ terminating	Use of the ""P/STOP NSQMQ"" OS/390 command has been detected.	None.
NSQ2001I	M6-WMQ Publish/Subscribe Server has started version	version: Version of Publish/Subscribe Server in format vv.rr.mmm, for example 04.05.0001	None.
NSQ2100I	An event was detected which was found in the SEVERITY file. event:: Event number	severity: Severity value from SEVERITY file	None.
NSQ2101W	Cannot write into event queue RC (reason)	reason: WebSphere MQ reason code.	Review WebSphere MQ reason code for possible problems and correct. If necessary, contact Nastel Support.
NSQ2102I	QMGR (qmgr) no longer exists, thread is terminating	qmgr: Queue Manager	None.
NSQ2104W	Unable to connect to QMGR (qmgr) RC (reason)	qmgr: Queue Manager reason: WebSphere MQ reason code	Review WebSphere MQ reason code for possible problems and correct. If necessary, contact Nastel Support.

Table K-1. WMQ Agent z/OS Error Codes and Messages (Sheet 9 of 14)

Msgid	Message Text	Explanation	User Action
NSQ2105I	QMGR(qmgr)is deleted, thread is terminating	qmgr: Queue Manager	None.
NSQ2106W	Unable to get DLQ name QMGR(qmgr)	qmgr: Queue Manager	Either specify a DLQ for the queue manager or restart the publisher without DLQ monitoring. If necessary, contact Nastel Support.
NSQ2107A	DLQ name not configured in qmgr, please edit QMGR(qmgr) properties	qmgr: Queue Manager	Perform indicated action (but see NSQ2106W); if necessary, contact Nastel Support.
NSQ2108W	Unable to open qmgr DLQ(queue) RC(reason)	qmgr: Queue Manager queue: Dead Letter Queue Name reason: WebSphere MQ reason code	Review for possible problems and correct. If necessary, contact Nastel Support.
NSQ2109W	MQINQ DLQ failed QMGR(qmgr) RC(reason)	qmgr: Queue Manager reason: WebSphere MQ reason code	Review for possible problems and correct. If necessary, contact Nastel Support.
NSQ2110I	Cannot read from input queue DLQ (queue) QMGR (qmgr) RC (reason)	queue: Dead Letter Queue qmgr: Queue Manager reason: WebSphere MQ reason code	Review for possible problems and correct. If necessary, contact Nastel Support.
NSQ2120E	Insufficient memory available to monitor QMGR (qmgr)	qmgr: Queue Manager	Increase REGION parameter and retry function; if problem persists, contact Nastel Support.
NSQ2121I	QMGR (qmgr) is deleted, thread is terminating	qmgr: Queue Manager	None.
NSQ2122I	DLQ thread for QMGR (qmgr) started.	qmgr: Queue Manager	None.
NSQ2123I	QMGR (qmgr) is deleted, thread is terminating	qmgr: Queue Manager	None.
NSQ2124W	Unable to get DLQ name QMGR (qmgr)	qmgr: Queue Manager Review for possible problems.	If necessary, contact Nastel Support.

Table K-1. WMQ Agent z/OS Error Codes and Messages (Sheet 10 of 14)

Msgid	Message Text	Explanation	User Action
NSQ2125W	Cannot read from input queue DLQ (queue) QMGR (qmgr) RC (reason)	queue: Dead Letter Queue qmgr: Queue Manager reason: WebSphere MQ reason code	Review WebSphere MQ reason code for possible problems and correct. If necessary, contact Nastel Support.
NSQ2126W	Cannot get SHUTDOWN message under browse cursor DLQ(queue) QMGR(qmgr) RC(reason)	queue: Dead Letter Queue qmgr: Queue Manager reason: WebSphere MQ reason code	Review WebSphere MQ reason code for possible problems and correct. If necessary, contact Nastel Support.
NSQ2127I	DLQ thread for QMGR (qmgr) has terminated	qmgr: Queue Manager	None.
NSQ2130T	Cannot get authentication signature	The M6-WMQ Event Publisher has been started with the -a parameter but this function has failed.	Contact Nastel Support.
NSQ2132T	Cannot create MQ node	An internal failure has occurred.	Contact Nastel Support.
NSQ2133T	Cannot open connections with MQNote	An internal failure has occurred.	Contact Nastel Support.
NSQ2134T	Cannot write SHUTDOWN message	An internal failure has occurred during shutdown.	Contact Nastel Support.
NSQ2135W	RawToPcf failed	An internal failure has occurred.	Contact Nastel Support.
NSQ2137E	Error for tsched () RC (reason)	reason: MTF error code	Retry function; if problem persists, contact Nastel Support.
NSQ2139E	Error for tsched () RC (reason)	reason: MTF error code	Retry function; if problem persists, contact Nastel Support.
NSQ2141E	Unable to connect to QMGR (qmgr) RC (reason)	qmgr: Queue Manager reason: WebSphere MQ reason code	Review WebSphere MQ reason code for possible problems and correct. Retry function; if problem persists, contact Nastel Support.
NSQ2143E	Unable to write to qmgr EventQ(queue) RC(reason)	qmgr: Queue Manager queue: Event Queue reason: WebSphere MQ reason code	Review WebSphere MQ reason code for possible problems and correct. Retry function; if problem persists, contact Nastel Support.
NSQ2145E	Unable to connect to QMGR (qmgr) RC (reason)	qmgr: Queue Manager reason: WebSphere MQ reason code	Review WebSphere MQ reason code for possible problems and correct. Retry function; if problem persists, contact Nastel Support.

Table K-1. WMQ Agent z/OS Error Codes and Messages (Sheet 11 of 14)

Msgid	Message Text	Explanation	User Action
NSQ2147E	Unable to write to qmgr DLQ (queue) RC (reason)	qmgr: Queue Manager queue: Event Queue reason: WebSphere MQ reason code	Review WebSphere MQ reason code for possible problems and correct. Retry function; if problem persists, contact Nastel Support.
NSQ2149T	Max number of subtasks (ntasks) is greater than MAXTASK (maxtasks)	ntasks: Number of MTF tasks requested maxtasks: Number of tasks MTF will allow	Reduce number of tasks requested and restart. If problem persists, contact Nastel Support.
NSQ2150E	Error on tinit() RC(reason)	reason: MTF error code	Retry function; if problem persists, contact Nastel Support.
NSQ2151W	Cannot create UDP MQ node, port (port) REASON (reason) message, will try next port	port: TCP/IP UDP port number reason: TCP/IP error code message: Further information	Review TCP/IP error code and message text for possible problems and correct. If necessary, contact Nastel Support.
NSQ2153T	Cannot create UDP MQ node	All attempts to create a UDP node failed, unable to start the M6-WMQ MQ Event Publisher.	Check that ports are not otherwise in use; if so, use a different port. If problem persists, contact Nastel Support.
NSQ2154E	Error for tterm() RC(reason)	reason: MTF error code	Retry function; if problem persists, contact Nastel Support.
NSQ2155T	Cannot open connections with the UDP MQNode	A possible TCP/IP error is preventing connection. Examine stdout for other error messages.	Contact Nastel Support.
NSQ2156E	Error for tterm() RC(reason)	reason: MTF error code	Retry function; if problem persists, contact Nastel Support.
NSQ2157W	Number of QMGRs is greater than max QMGRs parameter (-bmaxqmgrs)	maxqmgrs: Maximum number of queue managers that can be handled by the Publish/Subscribe Server.	Increase the max queue managers value (-b) and retry. If problem persists, contact Nastel Support.
NSQ2158W	Error on ThreadLstnr EvQ()RC (reason)	reason: Error code.	Retry function; if problem persists, contact Nastel Support.
NSQ2159W	Error on ThreadLstnr DLQ()RC (reason)	reason: Error code	Retry function; if problem persists, contact NastelSupport.

Msgid	Message Text	Explanation	User Action
NSQ2160E	Error from GetPcfStrListAttrValue() RC(reason)	reason: Error code.	Retry function; if problem persists, contact Nastel Support.
NSQ2162I	SHUTDOWN message received		None.
NSQ2163W	Error on mnRead() of UDP/PCF message RC(reason)	reason: WebSphere MQ reason code	Review WebSphere MQ reason code for possible problems and correct. If necessary, contact Nastel Support.
NSQ2164T	Loop detected on UDP read error	An unrecoverable state was reached handling UDP errors.	Contact Nastel Support.
NSQ2165W	Error for tterm() RC (reason)	reason: MTF error code.	Review for possible problems and retry. If necessary, contact Nastel Support.
NSQ2167W	MQINQ (function) failed with RC (reason)	function: Function attempted reason: WebSphere MQ reason code	Review WebSphere MQ reason code for possible problems and correct. If necessary, contact Nastel Support.
NSQ2168I	Unable to open temporary subscriber file QMGR (qmgr) FILE(file) RC(reason) message	qmgr: Queue Manager file: ddname of temp subscriber file reason: C library error code from open failure message: Informative error message	If using this function, review error and correct. Otherwise, none.
NSQ2170W	Unable to open permanent subscriber file FILE(file) RC(reason) message	file: ddname of temp subscriber file reason: C library error code from open failure message: Informative error message	If using this function, review error and correct. Otherwise, none.
NSQ2171I	Unable to open temporary subscriber file, QMGR(qmgr) QUEUE (queue)	qmgr: Queue Manager queue: Queue Name	None.
NSQ2172I	Unable to open subscriber file, QMGR (qmgr) QUEUE (queue)	qmgr: Queue Manager queue: Queue Name	None.
NSQ2173I	QMGR (qmgr) no longer exists, thread is terminating	qmgr: Queue Manager	None.
NSQ2174I	QMGR (qmgr) no longer exists, thread is terminating	qmgr: Queue Manager	None.
NSQ2175W	Unable to connect to QMGR (qmgr) RC(reason)	qmgr: Queue Manager	Review for possible problems. If necessary, contact Nastel Support.

Table K-1. WMQ Agent z/OS Error Codes and Messages (Sheet 13 of 14)

Msgid	Message Text	Explanation	User Action
NSQ2176I	QMGR (qmgr) is deleted, thread is terminating	qmgr: Queue Manager	None.
NSQ2177W	Unable to open qmgr EventQ (queue) RC(reason)	qmgr: Queue Manager queue: Queue Name reason: WebSphere MQ reason code	Review WebSphere MQ reason code for possible problems. If necessary, contact Nastel Support.
NSQ2179W	QMGR (qmgr) is deleted, thread is terminating	qmgr: Queue Manager	None.
NSQ2182W	Can not read from input QUEUE (queue) QMGR (qmgr) RC (reason)	qmgr: Queue Manager queue: Queue Name reason: WebSphere MQ reason code	Review WebSphere MQ reason code for possible problems and correct. If necessary, contact Nastel Support.
NSQ2183W	Unable to write to subscriber QMGR (qmgr) Queue (queue) RC (reason)	qmgr: Queue Manager queue: Queue Name reason: WebSphere MQ reason code	Review WebSphere MQ reason code for possible problems and correct. If necessary, contact Nastel Support.
NSQ2184W	Subscriber EventQ (queue) was not created QMGR (qmgr)	qmgr: Queue Manager	Review WebSphere MQ reason code for possible problems and correct. If necessary, contact Nastel Support.
NSQ2185I	Publisher thread for QMGR (qmgr) had terminated	qmgr: Queue Manager	None.
NSQ2187I	Publisher thread for QMGR (qmgr) started	qmgr: Queue Manager	None.
NSQ2990T	Signal sig received, terminating	An internal error has occurred sig: Signal code	Contact Nastel Support.
NSQ2992T	Enqueue failed for PUBSUB QueueManager (qmgr).	Another Publisher is already active for the indicated queue manager.	Ensure the two Publishers are using different MQSINI files. Remove the qmgr entry from one of the MQSINI parameter files. Contact Nastel Support if this message is produced in error.
NSQ2999I	z/OS STOP issued, NSQPUB terminating	Use of the ""P/STOP NSQMQ"" OS/390 command has been detected.	None.
NSQ3001I	M6-WMQ Message Server NSQMSG has started Version.	version: Version of WMQ Agent in format vv.rr.mmm, for example 04.05.0001.	None.

Msgid	Message Text	Explanation	User Action
NSQ3101E	Error for tsched (reason) in NSQMSG	reason: MTF error code	Retry function; if problem persists, contact Nastel Support.
NSQ3990T	Internal segmentation violation.	An internal error has occurred.	Contact Nastel Support.
NSQ3992T	Enqueue failed for MMF Queue Manager \\\"qmgr\\\".	Only one MMF Server per queue manager can be started on an z/OS system image. qmgr: Queue Manager	Do not try to start a second MMF Server for a queue manager. If this message is issued in error, contact Nastel Support.
NSQ3999I	z/OS STOP issued, NSQMSG terminating	Use of the \"P/STOP NSQMQ\" OS/390 command has been detected.	None.

Appendix L: Exit Codes When M6-WMQ Runs as Service on Windows

When running M6-WMQ applications as services on Windows, the following exit codes are returned if the service fails to run. In the case of the Workgroup Server, the workgroup log file is located in `[APWMQ_HOME]/groups/<workgroup>/<workgroup>.log`, where `<workgroup>` is the name of the workgroup).

Table L-1. Exit Codes when M6-WMQ Applications Run as Services on Windows

Application	Exit Code	Description
Workgroup Server	43	WGS uses "43" only in a case of license check failure. It writes to the log file the following messages: "IGE0043: Unable to obtain digital signature!" "IGE0014: Initialization failed, unable to proceed." If the workgroup server is a windows service, it sets the status code to 43 and quits.
	21010	Service could not locate workgroup database record.
	21101	Service encountered an error initializing internal variables.
	21027	Service could not load node-based licensing information.
	21028	Host is not licensed to run Workgroup Server.
	21031	Node-based licensing information is invalid or corrupt.
	21032	Workgroup license has expired.
	21046	Specified workgroup is not licensed.
	21047	Service could not load CPU-based licensing information.
	21048	Service is not licensed to use Kerberos authentication.
M6-WMQ Agent and Connection Manager	21100	Service was started with invalid argument(s).
	21101	Service encountered an error initializing internal variables.
	21110	Service encountered a memory access violation.
	21111	Service received a signal indicating that it should terminate.
	21113	Service failed to initialize.
Event Adapter	21100	Service was started with invalid argument(s).
	21101	Service encountered an error initializing internal variables.
	21102	Service could not resolve workgroup name to a valid host and/or port no.
	21110	Service encountered a memory access violation.
	21111	Service received a signal indicating that it should terminate.
Log Adapter	21100	Service was started with invalid argument(s).
	21101	Service encountered an error initializing internal variables.
	21102	Service could not open log file.
	21120	Service could not determine appropriate log file or specific platform.

Table L-1. Exit Codes when M6-WMQ Applications Run as Services on Windows

Application	Exit Code	Description
Message Server and Event Publisher	21110	Service encountered a memory access violation.
	21111	Service received a signal indicating that it should terminate.

Glossary

This appendix contains a list of reference material and documents relevant to M6-WMQ and other related Nastel products.

API: *See* Application Programming Interface

Application Programming Interface (API): A set of declarations of the functions (or procedures) that an operating system, library or service provides to support requests made by computer programs.

AP-WMQ: Nastel Technologies' WebSphere MQ management solution. Re-designated as AutoPilot M6 for WebSphere MQ with release 6.0. Abbreviated as AP/WMQ and AP-WMQ.

AP/WS: *See* AutoPilot/WebSphere

AutoPilot M6: Nastel Technologies' Application Management Platform. AutoPilot M6 monitors and automates the management of eBusiness integration components such as middleware application, application servers and user applications.

AutoPilot/WebSphere (AP/WS): AutoPilot/WebSphere Server enables AutoPilot to monitor and manage eBusiness applications for continuous operations in addition to its standard features.

BSV: *See* Business View.

Business View (BSV): A collection of rules that define a desired state of an eBusiness environment. Business Views can be tailored to presents information in the form most suited to a given user, as defined by the user.

Client: Any programming component that uses the AutoPilot M6 infrastructure; for example, the AutoPilot M6 Console.

Common Object Request Broker Architecture (CORBA): A Common Object Request Broker Architecture (CORBA) object can be invoked from a Web browser using CGI scripts or applets.

Console: The console acts as the graphical interface for AutoPilot M6.

Contacts: A subordinate to a given Manager or Expert.

CORBA: *See* Common Object Request Broker Architecture.

Data Source Name (DSN): A Data Source Name (DSN) is the logical name that is used by Open Database Connectivity (ODBC) to refer to the drive and other information that is required to access data. The name is use by Internet Information Services (IIS) for a connection to an ODBC data source, (For Example: Microsoft SQL Server database). The ODBC tool in the Control Panel is used to set the DSN. When the ODBC DSN entries are used to store the connection string values externally, you simplify the information that is needed in the connection string. This makes changes to the data source completely transparent to the code itself.

Deploy: To put to use, to position for use or action.

Domain Server: A specialized managed node that maintains the directory of managed nodes, experts etc. The domain server is also capable of hosting experts, managers etc.

DSN: *See* Data Source Name.

EVT: Event Log file extension (e.g.: *sample.evt*)

Event: Something that happens to an object. Events are logged by AutoPilot M6 and are available for use by AutoPilot M6 Policies or the user.

Expert: Services that monitor specific applications such as an applications server, web-server or specific components within the applications (Example: channels in WebSphere MQ.) Experts generate facts.

Fact: Facts are single pieces of data that have a unique name and value. One or more facts are used to determine the health of the object, application or server.

Graphical User Interface (GUI): A type of environment that represents programs, files, and options by means of icons, menus, and dialog boxes on the screen. The user can select and activate these options by pointing and clicking with a mouse or, often, with the keyboard.

GUI: *See* Graphical User Interface.

HMAC: *See* Hash-Based Message Authentication Code.

Independent WebSphere MQ Node: A WebSphere MQ node that runs a WMQ Agent and which is managed directly by an MQ Workgroup server.

IIS: *See* Internet Information Services.

Internet Information Services (IIS): Microsoft's brand of Web server software, utilizing HTTP to deliver World Wide Web documents. It incorporates various functions for security, allows CGI programs, and also provides for Gopher and FTP services.

Java: A platform-independent, object-oriented programming language developed and made available by Sun Microsystems.

Java Developer's Kit (JDK): A set of software tools developed by Sun Microsystems, Inc., for writing Java applets or applications. The kit, which is distributed free, includes a Java compiler, interpreter, debugger, viewer for applets, and documentation.

JDBC: *See* Java Database Connectivity.

Java Database Connectivity (JDBC): The JDBC API provides universal data access from the Java programming language. Using the JDBC 2.0 API, you can access virtually any data source, from relational databases to spreadsheets and flat files. JDBC technology also provides a common base on which tools and alternate interfaces can be built. The *JDBC Test Tool* that was developed by Merant and Sun Microsystems may be used to test drivers, to demonstrate executing queries and getting results, and to teach programmers about the JDBC API.

Java Server Pages (JSP): JSP technology enables rapid development of web-based applications that are platform independent. Java Server Pages technology separates the user interface from content generation enabling designers to change the overall page layout without altering the underlying dynamic content. Java Server Pages technology is an extension of the Java Servlet technology.

Java Virtual Machine (JVM): The "virtual" operating system that JAVA-written programs run. The JVM is a hardware- and operating system-independent abstract computing machine and execution environment. Java programs execute in the JVM where they are protected from malicious programs and have a small compiled footprint.

JDK: *See* Java Developer's Kit.

JRE: JAVA Run-time Environment. The minimum core JAVA required to run JAVA Programs.

JSP: *See* Java Server Pages.

JVM: *See* Java Virtual Machine.

KDC: *See* Key Distribution Center.

M6 for WMQ: Nistel Technologies' WebSphere MQ management solution. Re-designated as M6 for WMQ with release 6.0, prior releases retain the AP-WMQ or MQControl trademark.

M6 Web: A browser-based interface that provides monitoring and operational control over managed resources and applications.

Management Information Base (MIB): A specification that describes the properties and behavior of a network device. Network managers use MIBs to interact with SNMP-compatible devices. Each MIB is part of a directory structure that specifies where objects are found on the network.

Manager: Managers are the home or container for policies. All business views must reside on managers, and manager must be deployed prior to deploying a business view or policy.

Message Queue Interface: The Message Queue Interface (MQI) is part of IBM's Networking Blueprint. It is a method of program-to-program communication suitable for connecting independent and potentially non-concurrent distributed applications.

MIB: *See* Management Information Base.

MQControl: Nistel Technologies' MQSeries management product. Re-designated as AP-WMQ with release 4.0 and M6 for WMQ with release 6.0. Prior releases retain the MQControl trademark.

MQI: *See* Message Queue Interface.

MQSC: *See* WebSphere MQ Commands.

MQSeries: IBM's message queuing product. Renamed by IBM as WebSphere MQ.

Naming Service: A common server records "names" of objects and associates them with references, locations and properties.

Managed Node: A container that can host any number of AutoPilot M6 services such as experts, managers, policies, etc. Unlike managed nodes, it is a physical process.

ORB: Object Request Broker.

Orbix: CORBA product distributed by IONA Technologies.

Package Manager (PKGMAN): The command line utility that allows users to list, install, uninstall, verify and update AutoPilot M6 installation on any Managed Node.

PCF: *See* Programmable Command Format.

PKGMAN: *See* Package Manager.

Policy/Business Views: Business views are a collection of one or more sensors. Business views are used to visually present the health and status of the different systems as well as automatically issue remedial actions.

Programmable Command Format (PCF): A set of programmable commands that M6-WMQ uses to manage WebSphere MQ. PCF includes data definitions for items such as integers, strings, and lists. The commands can be submitted directly to a queue manager. PCF is comparable to MQSC, except for the fact that MQSC cannot be programmed.

SG: *See* Queue Sharing Group.

Queue Sharing Group (QSG): In z/OS, a group of queue managers in the same sysplex that can access a single set of object definitions stored in the shared repository, and a single set of shared queues stored in the coupling facility. The shared queue is a type of local queue. The messages on the queue are stored in the coupling facility and can be accessed by one or more queue managers in a queue-sharing group. The definition of the queue is stored in the shared repository.

Sensor: A rule that is used to determine the health of an object or application based on one or more facts. Actions can then be issued, based on health. Sensors are definable in AutoPilot business views by use of the sensor wizard.

Simple Mail Transfer Protocol (SMTP): A TCP/IP protocol for sending messages from one computer to another on a network. This protocol is used on the Internet to route e-mail. *See also* communications protocol, TCP/IP. *Compare* CCITT X series, Post Office Protocol.

Simple Network Management Protocol (SNMP): A de facto standard for managing hardware and software devices on a network. Each device is associated with a Management Information Base (MIB) that describes its properties and behavior.

SMTP: *See* Simple Mail Transfer Protocol.

SNMP: *See* Simple Network Management Protocol.

SNMP Master Agent: An implementation of the SNMP protocol. It includes a definition of the standard MIB. The master agent routes SNMP requests from subagent to subagent.

SNMP Subagent: The implementation of an MIB for a particular device. The MIB describes the device's desired behavior; the SNMP subagent carries it out.

TCP/IP: *See* Transmission Control Protocol/Internet Protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP): A protocol developed by the Department of Defense for communications between computers. It is built into the UNIX system and has become the de facto standard for data transmission over networks, including the Internet.

Virtual Machine: Software that mimics the performance of a hardware device, such as a program that allows applications written for an Intel processor to be run on a Motorola chip. *See* Java Virtual Machine.

WAP: *See* Wireless Application Protocol.

WebSphere MQ: IBM's message queuing product. Formally known as MQSeries.

WebSphere MQ Commands: A command-line language used to configure WebSphere MQ.

Websphere_MQ_Manager: A specialized manager capable of hosting one or more MQSeries specific policies, apart from the regular policies.

Workgroup Server (WS): Monitors WebSphere MQ nodes. A workgroup server consists of two agents, M6 managed node and workgroup.

Wireless Application Protocol (WAP): An open global specification that is used by most mobile telephone manufacturers. WAP determines how wireless devices utilize Internet content and other services. WAP enables devices to link diverse systems contents and controls.

WS: *See* Workgroup Server.

z/OS: *See* Z Series Operating System.

Z Series Operating System (z/OS): IBM architecture for mainframe computers and peripherals. The zSeries family of servers uses the z/Architecture. It is the successor to the S/390 and 9672 family of servers.